The Semi-automatic Expansion of Existing Terminological Ontologies using Knowledge Patterns Discovered on the WWW

An Implementation and Evaluation

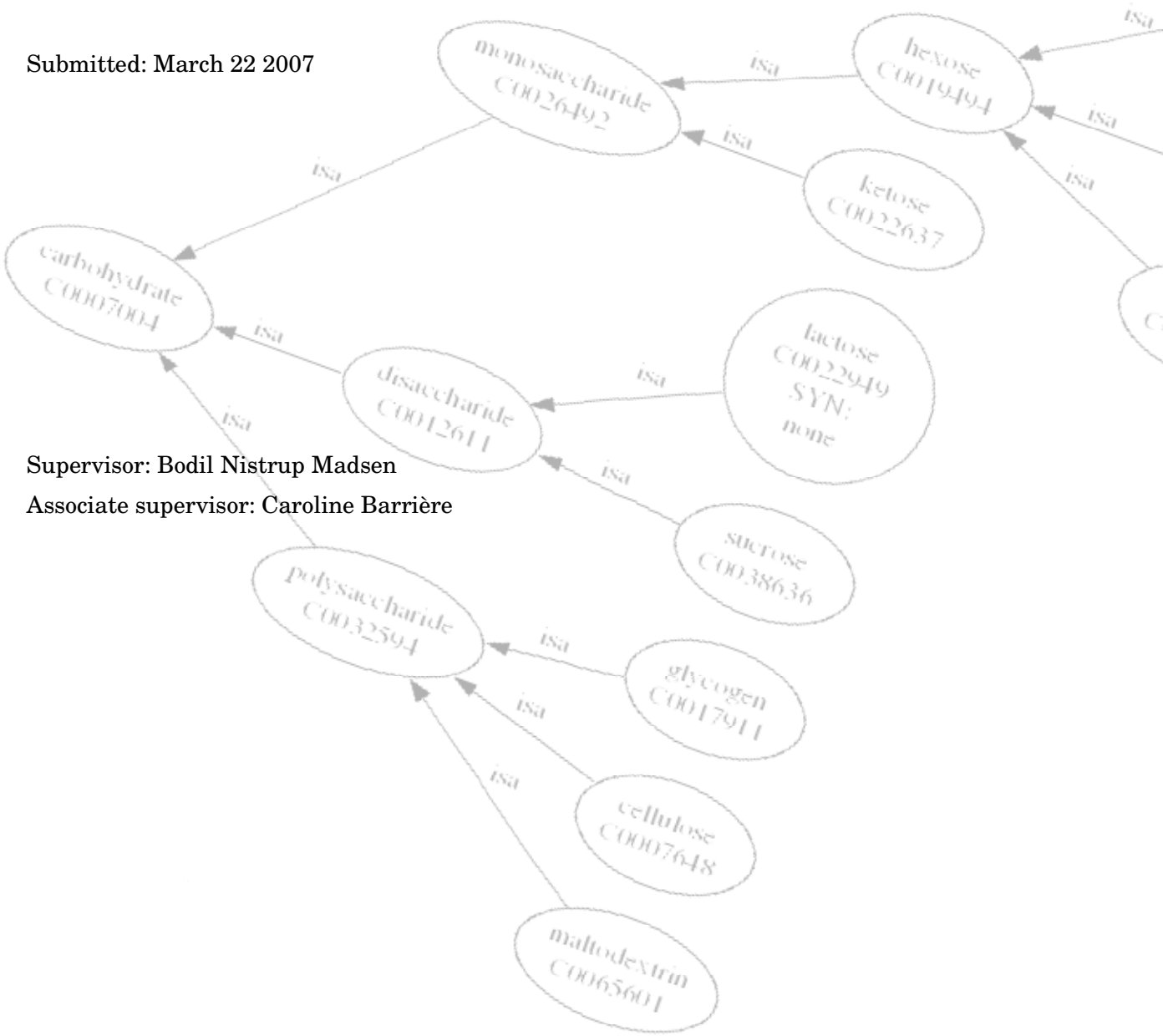
Halskov, Jakob

The semi-automatic expansion of existing terminological ontologies using knowledge patterns discovered on the WWW - an implementation and evaluation

by Jakob Halskov

Copenhagen Business School
Department of International Language Studies
and Computational Linguistics

Submitted: March 22 2007



Supervisor: Bodil Nistrup Madsen
Associate supervisor: Caroline Barrière

Abstract

The research object of this thesis is the so-called knowledge patterns and their usefulness in automatically extracting specific semantic relations from unannotated and uncategorized text on the WWW so as to facilitate semi-automatic updating and extension of existing ontological and terminological resources.

The main contribution of the thesis is the implementation of a complete ontology extension framework called WWW2REL which is 100% based on a knowledge-poor, domain-independent processing of WWW text snippets and includes the three stages of pattern discovery, pattern filtering and relation instance ranking. Unlike most comparable systems WWW2REL is special in that it is both highly portable, can be applied to any semantic relation type and operates directly on uncategorized WWW text snippets.

The system is tested on the biomedical UMLS Metathesaurus for four different relation types and manually evaluated by four domain experts. It is demonstrated that high precision in the task of knowledge discovery from a noisy text source can be achieved using a very simple instance relevance measure and two ranking heuristics. In contrast, many comparable systems operate on richly annotated academic text and tend to apply heuristics which are custom-tailored to a specific domain and/or relation type. When selecting the overall best ranking scheme, average system performance across all four relation types ranges between 70% to 65% of the maximum possible F-score by top 10 and top 50 relation instances, respectively.

Finally, the thesis experiments also examine the portability of individual knowledge patterns and of the ranking heuristics. It is concluded that synonymy KPs are the most domain independent closely followed by ISA KPs, whereas patterns for "may_prevent" and especially "induces" are more dependent on the domain. Empirical experiments also suggest that a ranking heuristic which penalizes relation instances whose arguments occur frequently in a general language corpus can be highly effective, but may need to be adapted to the domain in question.

Abstract

Forskningsgenstanden for dette projekt er de såkaldte "vidensmønstre" og deres anvendelighed i forhold til den automatiske fremfinding af semantiske relationer fra uopmærket og ukategoriseret tekstmateriale på WWW med henblik på en halvautomatisk opdatering af eksisterende terminologiske ressourcer.

Afhandlingens hovedbidrag består i en implementering og evaluering af et komplet ontologiudvidelsesværktøj, kaldet WWW2REL, som er 100% baseret på en vidensfattig og domæneuafhængig behandling af tekstfragmenter på WWW og omfatter både mønsteridentifikation og mønsterfiltrering så vel som en automatisk relevansvurdering af de ekstraherede relationer. I modsætning til de fleste sammenlignelige systemer skiller WWW2REL sig ud ved at være både domæne- og relationsuafhængig og samtidig netbaseret.

Systemet afprøves på den biomedicinske UMLS Metathesaurus for fire forskellige relationstyper og evalueres manuelt af fire fageksperter. Det påvises, at domæneuafhængig vidensfremfinding fra en ukategoriseret tekstkilde kan ske med høj præcision ved hjælp af et meget enkelt relevansmål og to heuristiske sorteringsmetoder. Mange sammenlignelige systemer anvender udelukkende faglitterære og semantisk opmærkede tekster og er ofte skræddersyet til et enkelt domæne og/eller bestemt relationstype. Ved valg af den samlet set bedste rangordningsalgoritme opnår systemet i gennemsnit mellem 70% og 65% af den højest mulige F-score ved henholdsvis top 10 og top 50 relationskandidater.

Afhandlingen undersøger endvidere anvendeligheden af de enkelte vidensmønstre og sorteringsmetoder på tværs af domæner. Det konkluderes, at vidensmønstre for synonymi er de mest tværfaglige, tæt fulgt af mønstre for den generiske relation, hvorimod vidensmønstre for de to kausale relationer er mindre tværfaglige. Slutteligt indikerer empiriske eksperimenter, at en sorteringsmetode, som straffer relationer, hvis argumenter er hyppige i et almensprogligt korpus kan være meget effektiv, men sandsynligvis bør tilpasses til det enkelte domæne.

List of Tables

List of Figures

Contents

Preface Thanks are extended to Hanne Dihn, Birgitte Graumann, Lena Skov Andersen and Nguyen Lee without whom the manual system evaluation would not have been carried out. Also, I am deeply indebted to Caroline Barrière whose untiring enthusiasm and feedback has provided an invaluable source of inspiration and encouragement. Finally, I am very grateful for the supervision provided by Bodil Nistrup Madsen and Sabine Kirchmeier-Andersen, for ideas and comments from colleagues at the Dept. of Computational Linguistics and for the fact that my wife put up with me through the years of thesis work.

1 Introduction

With the digital revolution and the genesis of a vast and freely accessible repository of text and knowledge known as the Internet, researchers from many fields, including text mining, computational linguistics and terminology, are struggling to overcome a major challenge of the Internet Age, namely information overload. How does one find the gold nuggets of relevant knowledge washing down the information river?

In the context of computational terminology, especially for the task of generating or updating ontologies and terminological knowledge bases, an important type of gold

nuggets are semantic and conceptual relations[1] expressed explicitly in natural language strings. Such strings have been called knowledge-rich contexts (KRCs) and a popular way of identifying KRCs has been by looking for so-called "knowledge patterns" or KPs [Meyer, 2001, p290], which are instantiations of semantic relations in text. A KRC has been defined as

> a context indicating at least one item of domain knowledge that could be useful for conceptual analysis. In other words, the context should indicate at least one conceptual characteristic, whether it be an attribute or a relation. [Meyer, 2001, p281]

The two following sentences are examples of a KRC containing a KP delimited by angle brackets.

1. Revici was also an early advocate of using selenium <to treat> cancer.
2. The use of antipsychotic medications<, including> haloperidol, can be associated with ...

In these cases the two KPs, "to treat" and "including", can be used as entry points to the terminological gold nugget or KRC. While the pattern "to treat" can identify the causal relation between "selenium" and "cancer", the pattern ", including" may identify the generic, or ISA, relation between "antipsychotic medications" and "haloperidol". The main strategy of pattern-based approaches to relation extraction from free text is to compile lists of reliable patterns instantiating specific semantic relation types and use these lists to find new instances and gradually improve the coverage of (existing) ontologies. Section 2.1 provides more details on pattern-based approaches to automatic knowledge acquisition (AKA).

Automatically extracting semantic relation instances, the building blocks of ontologies, from free text is a way of minimizing the labor-intensive phase of manual knowledge engineering and thus overcoming the long-standing knowledge acquisition bottleneck. While ontologies are the end-product of the terminological tasks of conceptual clarification and knowledge structuring, they play an ancillary but vital role in the wider field of Natural Language Processing (NLP), for example

1. allowing automatic inference in Question Answering (QA) systems
2. recognizing textual entailment[2]
3. allowing automatic query expansion in Information Retrieval systems

Especially the latter point is the object of intense interest because it may in time realize the vision of the Semantic Web on which users may search for *content* rather than textual *strings*. The benefits of conceptual indexing were described already in [Woods, 1997], and there are now multiple examples of concept-based (as opposed to

[1] in this thesis semantic relations are understood as a hypernym of conceptual relations. Synonymy is an example of a semantic relation type which is not also a conceptual relation.

[2] see e.g. the PASCAL conferences (www.pascal-network.org)

keyword-based) IR projects, for example Ontoseek [Guarino et al., 1999], Ontobroker [Decker et al., 1999] and Ontoquery[3].

Ontologies play an important part in making web search engines behave more intelligently by providing them with knowledge about the world so that a user query for the string "antipsychotics", for example, may also return documents containing not this string itself, but perhaps relevant hyponyms like "haloperidol". As a result of this ontological preoccupation in the research communities the focus in *computational* terminology is also shifting from the investigation of terms and termhood to investigating the conceptual and semantic relations between terms[4]. On the application side there has been a change from the automatic term recognition (ATR) task to the automatic relation extraction task, which, in a terminological context, includes ATR as a subtask.

Whether represented as lexical networks (e.g. [Byrd and Ravin, 1999]) or, in the case of conceptual relations, as ontologies (e.g. [Cimiano and Staab, 2005]), the approaches to identifying semantic or conceptual relations in free text basically fall into two categories.

1. pattern-based (pioneered by [Hearst, 1992, Ahmad and Fulford, 1992])
2. clustering-based (pioneered by [Michalski and Stepp, 1983, Fisher, 1987])

Clustering methods induce classes of co-hyponyms from text using the distributional hypothesis that lexical items which occur in similar contexts are semantically similar. While conceptual clustering can be a completely unsupervised approach, the pattern-based methods require a few training examples (seed relation instances) to learn recurrent patterns for a target relation type. These patterns can then be used to find more relation instances by which more patterns can be discovered and so forth.

While the advantage of conceptual clustering is that it is unsupervised and makes full use of the contextual information in the training data, its weakness is that the concept clusters which are induced are unlabelled. Also, it works only for the generic, or ISA, relation. Perhaps the main weakness of pattern-based approaches, on the other hand, is that the individual patterns must be learned prior to the extraction process. To be learned the patterns obviously must be present in the data source, and this entails a potential data sparseness problem, which can hopefully be overcome by using the WWW as data source, however (see sections 3.1 and 4.1 for more on this). It is also a potential weakness that some patterns are more reliable than others. However, this problem also affects conceptual clustering in that some contexts, and thus the features they provide, will be more reliable than others. Finally, although largely unexplored in the literature it is a potential weakness that patterns can be domain dependent (see section 6.5 for examples). The main advantage, on the other hand, is that patterns can be learned for any conceivable relation type and be used to build any kind of ontology rather than just taxonomies.

There is usually also a difference in terms of the purpose for which the two techniques of conceptual clustering and pattern-based relation extraction are used. In conceptual clustering the goal is to compile large lists of examples of specific classes or

[3] www.ontoquery.dk

[4] the latter has always been the main research object of manual terminology work (see subsection 2.2.4 for a discussion, however)

concepts so as to be able to find them in documents, for example for disambiguation purposes. In other words, it is not the *intension* of the concept, but its *extension* which is in focus. Although pattern-based relation extraction may also be used in this way (see the application survey in section 3.4), it can also be used to find non-taxonomical relations which may provide the essential and delimiting characteristics of specific concepts. For example, that pain killers [may prevent: pain] is a characteristic which delimits this concept from other drugs. When used in this way, it is the intension of the concepts which is in focus, and intension rather than extension is the classical research object of terminology (see subsection 2.2.3 for more).

In terminology (and not just in the domain of Biomedicine, which constitutes the case study of this thesis) many types of semantic relations are important. One need only glance at research like [Nuopponen, 1994, Nuopponen, 2005] to realize the wealth of different semantic relation types which may be important in terminology work. Some relation types are particularly prominent in certain domains. For example the causal relations, "induces" and "may_prevent", are important both in Biomedicine and Information Technology (IT) as illustrated in the experiments of this thesis (chapters 4 and 6). Other relation types are widely used across almost any conceivable domain, for example ISA, meronymy (or PART_OF) and FUNCTION are three basic relation types which are often used as delimiting characteristics when writing terminological definitions. Finally, the semantic (but not conceptual) relation of synonymy is important to clarify the concepts of any domain.

In a terminological framework the pattern-based approach is thus more appropriate than conceptual clustering, and this is why the main topic of the thesis is a thorough investigation of the discovery, filtering and application of knowledge patterns (KPs) to extract relation instances from the WWW in order to assist terminologists working to maintain and extend terminological resources.

1.1 Challenges

The main challenge in the pattern-based approach to automatic relation instance extraction is that KPs are not failproof access points to instances of the target semantic relations but can be noisy. In a landmark article [Meyer, 2001] lists the following challenges to using KPs in automatic extraction tasks.

1. unpredictability
2. polysemy
3. anaphoric reference
4. domain-dependency

That KPs are unpredictable simply reflects the fact that they are part of natural rather than controlled or artificial language. There is virtually no limit to the creativity with which human beings express themselves, also when conveying knowledge to each other. The polysemy, or ambiguity, of KPs is another fascinating feature of natural language (or annoying depending on the perspective). Anaphoric reference is a third

feature of natural language, notorious for its complexity and a hard nut to crack for NLP applications. Finally, the possible domain-dependence of KPs is a challenge which concerns recall rather than precision (see more below).

As for the precision-related challenges at least four kinds of noise can occur when relying on KPs to extract relation instances automatically.

1. The KP does not realize a semantic relation at all
2. The KP expresses a different semantic relation than the target one
3. The KP realizes the target semantic relation, but its arguments do not represent *domain-specific* concepts, or they are at least sematically too vague to be terminologically interesting
4. The KP realizes the target semantic relation, its arguments are domain-specific, but the relation is *incorrect*

An example of the first type of noise is the following sentence from the British National Corpus (BNC)

> What <is a> Caesarian Birth?

The pattern "is a", which might in other cases establish a hyponym-hypernym link, only establishes a link to the interrogative pronoun "what" and thus does not provide a hypernym of the concept represented by "Caesarian Birth". The second type of noise can be illustrated by the pattern "arise from" in the following sentence from a glossary of medical terms.

> Schwannomas and neurofibromas, tumors that <arise from> the sheaths that cover nerves and improve the conduction of nerve impulses.

In general language "arise from" will almost always (except for poetic language, perhaps) be used to establish cause-effect relations, so it is not unlikely that this string would be used as a KP for the retrieval of causal relation instances. In this case, however, it instantiates a locative relation between the concepts represented by "tumors" and "sheaths". As for the third type of noise, the BNC provides another example.

> The universe <is a> cold, dark place!

This time "is a" does establish a link between a hyponym (universe) and a hypernym (place), but this link involves concepts of such a general nature that the relation will presumably not be useful for terminologists who work bottom-up modelling the knowledge of special domains, or even subdomains.

This third type of noise is perhaps the hardest to identify and evaluate, because the line between fuzzy categories and domain-specific concepts can be difficult to draw (see also the discussion in section 2.2). Also, in non-terminological contexts this example would be perfectly valid and thus not be considered as noise. In this thesis, however, the purpose is to assist terminologists and domain experts compiling and

structuring *specialized* knowledge in terminological databases, and hence semantic relations between non-specialized concepts are regarded as noise.

Finally, the fourth type of noise is particularly relevant when using the WWW as a knowledge source. In neat collections of academic papers one would not expect to encounter many incorrect semantic relations, but when authorship, text type and many other important quality parameters are unknown, it is not totally inconceivable that some relation instances will simply be false. The following is an example from a synonymy experiment in subsection 5.5.5.

> 1000mg of vitamin c, <aka> Ester C, if you feel a cold or flu coming on.

Since ester c is a modified (chemically enhanced) form of vitamin c, the synonymy relation established by the KP, "aka", is incorrect. The informal acronym for "also known as", of course, signals that the communicative setting may not be an academic one, and this is perhaps the explanation why incorrect semantic relations are established. Incorrect relations may also arise from incomplete processing of the natural language strings (see the discussion in subsection 5.1.2) or in cases where the strings themselves are incomplete due to the nature of the empirical data (i.e. fragmentary WWW text snippets).

Besides tackling the noise, or precision, problem, KP-based approaches to the automatic extraction of semantic relations must address another issue which complicates matters. Although KPs form a smaller set of items than the set of terms of a domain, and although they are generally used across different domains, the "discovery power" of individual KPs is likely to differ greatly. Some patterns may be much more commonly used in domain X than domain Y, and some will be very reliable but occur only rarely. The quality conditions of a KP thus include at least the following three parameters.

1. High precision
2. High recall
3. High portability

Thus the main content of the three experimental chapters of this thesis (chapters 4, 5 and 6) is a comprehensive investigation of these three parameters based on a case study of KPs discovered in and evaluated on WWW text snippets. Striking the perfect balance between high recall, high precision and high portability can be hard, and the right balance may depend on the purpose of the application. However, to some extent the parameters are interdependent in that high precision KPs may tend to be domain-dependent and thus have a low portability and a low recall (at least in other domains than the one for which they were learned). Conversely, highly portable KPs will tend to have a high recall and, presumably, a somewhat lower precision.

As for the *system* precision and recall, WWW2REL extracts relation instances directly from the entire WWW and presents these to the user as ranked by their assessed reliability, so from a pragmatic viewpoint precision should be favoured over recall.

System recall is a somewhat artificial concept in the context of knowledge discovery using the entire WWW as a data source which nobody knows the exact bounds of (see subsection 3.1.2 for more on the Web as Corpus challenges).

1.2 Outline

The overall structure of the thesis is as follows. Chapter 2 contains its theoretical foundations and chapter 3 discusses methodological issues and provides a survey of comparable relation extraction systems. The KP discovery and filtering step of WWW2REL is described in chapter 4, while chapter 5 provides a comprehensive system test and evaluation. The topic of chapter 6 is also the system evaluation, but this time from a perspective of recall and portability rather than precision. Finally, chapter 7 summarizes the key results and contributions of the thesis and also outlines future work. Source code developed for the experiments and the system implementation is replicated in the appendices (chapter 8).

In more detail, the main research problems and hypotheses of the thesis are outlined in section 1.4, chapter 2 discusses theoretical aspects of the foundations of terminology (section 2.2), including the interdependent processes of knowledge discovery (section 2.1) and knowledge representation (section 2.3). In chapter 3 the methodological framework of the thesis experiments is outlined, and this chapter includes a brief introduction to corpus linguistics (section 3.1) with special emphasis on the nascent Web as Corpus field (subsection 3.1.1), information retrieval (section 3.2) and text mining (section 3.3). It also provides an overview of prominent, existing pattern-based applications for the automatic extraction of relation instances (section 3.4) and compares these systems with WWW2REL.

Chapter 4, then, describes the process of initializing WWW2REL. This initialization includes the establishment of a framework for pattern discovery (section 4.1) and pattern filtering (section 4.2) using selected relation types from a biomedical ontology as a case study. Chapter 5 discusses issues related to the implementation of WWW2REL (section 5.1) and the manual evaluation of its performance (section 5.2). It also presents a number of instance reliability ranking schemes and heuristics (section 5.3), and finally in sections 5.4 and 5.5 eleven system tests are carried out and evaluated by using the four sets of filtered patterns from chapter 4 to automatically retrieve and rank relation instances from the WWW.

Chapter 6 investigates the performance impact of system parameters like text snippet sample size (section 6.1) and the choice of reference corpus (section 6.2). Section 6.3 presents a comprehensive evaluation of the usefulness of WWW2REL's *individual* KPs in terms of precision and recall. Finally, the chapter also examines two additional parameters which are important when assessing the overall usefulness of the WWW2REL system, namely its ability to detect "new" knowledge not recorded in the starting ontology (section 6.4) and the portability of WWW2REL to another domain (section 6.5). Chapter 7 summarizes the results of the experiments and evaluation and also outlines directions for interesting future work.

1.3 Research delimitation

The research presented in this thesis is delimited along five different dimensions, namely as regards the purpose and perspective of the work, the empirical data used in the work, the techniques employed and finally the domain and language of analysis.

1. Purpose and perspective
   (a) The purpose is to develop a relation extraction system which may assist practical terminology work in any domain-specific setting.
   (b) Given (a) the perspective becomes the intension rather than extension of concepts.
2. Empirical data
   (a) The only source of empirical data are thousands of WWW text snippets each containing at most one or two sentences and possibly only sentence fragments. This ensures system portability, but is also motivated by a number of other advantages outlined in subsection 3.1.1.
3. Techniques employed
   (a) Given the delimitation in 1) the pattern-based approaches to relation extraction are more attractive than conceptual clustering. In other words, conceptual clustering techniques are ignored because the purpose is *not* to find long lists of possible instantiations of classes (i.e. the extension of concepts).
   (b) Partly given the fragmentary nature of the empirical data sophisticated NLP is not attempted.
      i. tagging and chunking is performed, but not full parsing
      ii. no attempt is made at resolving anaphora
      iii. semantic relations *within* NPs[5] (e.g. modifier-head relations) are ignored
4. Language of analysis
   (a) All experiments are restricted to English. The main motivation for this restriction is to ensure that the results may be useful to a wider research community, but the restriction is also dictated by the choice of case study and case ontology (see below).
5. Domain of analysis

[5] called "lexical term similarity" in [Nenadic and Ananiadou, 2006] as opposed to "syntactic term similarity" which they use to refer to KPs

(a) Given the delimitation in 1) the domain of analysis is not language for general purposes, but language for special purposes. More specifically, the domain of Biomedicine is selected as a case study for reasons outlined below.

While the semantics expressed by modifier-head relations remains an intriguing research area and has also been used to learn taxonomies from free text (see for example [Gillam, 2004, Gillam et al., 2005]), no attempt will be made at analyzing the modifier-head relations of biomedical terms. The reason is that these NP internal relations are largely expressed by *implicit* means, and the research object of this thesis is *explicit* knowledge patterns instantiating semantic relations *between* domain-specific concepts.

Finally, there are three compelling reasons for zeroing in on the biomedical domain. Firstly, Biomedicine is a huge domain which has an impact on the lives of practically all people on the planet. Secondly, because of increasingly swift drug development cycles, the biomedical domain is in dire need of tools which can assist in keeping ontological resources updated. The two relation types "induces" and "may_prevent" are particularly interesting because all drugs have to be tested for potential side effects, and copycat products are constantly introducing new side effects which have to be monitored. Thirdly, the Unified Medical Language System (UMLS) knowledge sources are not only among the most comprehensive ontological resources, they are also freely available[6], making them ideal as both a source of relation instances for KP discovery but also as a baseline for an automatic evaluation of system performance.

1.4 Contributions and hypotheses

The main contribution of this thesis is to implement, test and manually evaluate a complete ontology extension framework which is 100% based on a knowledge-poor, domain-independent, pattern-based processing of WWW text snippets and includes the three stages of pattern discovery, pattern filtering and relation instance ranking. A distinctive feature of this ontology extension framework is that it is optimized both for precision and portability (domain-independence) since these are two top priorities for the intended users, namely terminologists.

Secondary contributions include

- An in-depth empirical investigation of the performance of individual knowledge patterns, since such investigations are few and far between[7].
- A discussion of the pitfalls and challenges related to the evaluation of the recall of a knowledge discovery system versus an existing ontology.
- An investigation of system portability to another domain.

A key hypothesis is that high precision in automatic relation instance extraction can be achieved in spite of the following circumstances which complicate the task but boost system portability.

[6] http://umlsks.nlm.nih.gov

[7] [Barrière, 2001], [Girju and Moldovan, 2002] and [Marshman and L'Homme, 2006] are three examples.

1. The system operates exclusively on a noisy text source (the WWW).

2. The system makes no use of heuristics which are custom-tailored to any specific domain.

3. The system uses a very simple measure of relation instance reliability.

In the KP discovery phase it is hypothesized that enforcing certain restrictions on the form of candidate KPs, namely requiring that they contain a verb, will significantly reduce noise. If formal restrictions are not enforced, noisy KP candidates can be eliminated by measuring the range of different term pairs with which each candidate occurs during a ten-fold-validation process and deleting those with a low range.

As for the test phase it is hypothesized that high precision can be attained by grouping relation instances by their NP head, penalizing heads which are overly frequent in a general language corpus and ranking the instances by the range of different KPs with which they co-occur. Although it is difficult to find systems trained and tested in similar settings, the properties of WWW2REL and its performance can still be meaningfully compared to existing systems as is done in section 3.4.

A final hypothesis is that while assessing termhood by using frequency data from a general language corpus is an effective strategy for many specialized domains (including Biomedicine), this is not a good technique in domains where terms are predominantly formed by semantic extension of existing general language lexical units. The hypothesis is tested by applying the technique to a domain in which term formation is characterized by semantic extension (namely Information Technology).

2 Theory

This chapter provides the theoretical foundations for the methodological and empirical chapters which follow. As WWW2REL is essentially an application for automatic knowledge acquisition (AKA), the chapter starts off by a brief account of the AKA field, including a popular AKA technique called Dual Iterative Pattern Relation Expansion [Brin, 1998] which has provided inspiration for the implementation outlined in chapter 4.

WWW2REL is devised as an aid in practical terminology work and this focus necessitates a longer discussion of what is really meant by specialized knowledge (and specialized text) and how this might differ from knowledge as such (section 2.2). Also, since the relation instances extracted by WWW2REL must establish a link to a term in order to be judged relevant, this section includes a discussion on termhood (subsection 2.2.3) which leads to a brief account of the heated theoretical row over the research object of terminology as a science (subsection 2.2.4).

Finally, as WWW2REL is tested on a comprehensive biomedical ontology known as the ULMS Metathesaurus, section 2.3 contains a discussion on the properties of ontologies in general (subsection 2.3.1), *terminological* ontologies (subsection 2.3.2) and in particular the properties of the UMLS Metathesaurus (subsection 2.3.4).

2.1 Automatic knowledge acquisition

This section briefly outlines important challenges in the field of automatic knowledge acquisition, or AKA for short. Automatically extracting relation instances from the WWW is an AKA task, and from the following 10-year-old quote it is apparent how fast the AKA field is developing.

> It is often assumed that Knowledge Acqusition (KA) for expert systems and other knowledge-based programs must involve comprehensive interactive sessions with human experts. However there is now a growing awareness that a vast amount of human knowledge has already been extracted and codified in the form of printed text in dictionaries, thesauri, user manuals, encyclopaedias, reference guides and expository texts. [...] Knowledge Extraction (KE) systems [...] do not of course eliminate human input, but attempt to relegate it to a post processing phase, reducing the intellectual load on humans as far as possible whilst making good use of existing hard-won knowledge resources. [Bowden et al., 1996, p147]

In contrast to the eighties and early nineties the first stages of knowledge acquisition nowadays rarely involve human experts, and [Bowden et al., 1996, p147] have been proven right about the promise of AKA.

AKA is related to the field of knowledge discovery in databases (KDD), also known as data mining. However, the two terms are not synonymous as revealed by the following quote which defines KDD as

> [...] the non trivial extraction of implicit, previously unknown and potentially useful information in data. [Frawley et al., 1992]

Unlike data mining, AKA normally does not procure truly *new* knowledge, but rather *rediscovers* and, in the case of ontology learning, also pieces together *existing* fragments of knowledge typically found in natural language text rather than structured data. A more detailed discussion of the AKA-related fields of data mining, Information Retrieval and Information Extraction can be found in chapter 3 along with a number of prominent systems which automatically acquire knowledge.

First of all, AKA need not be used to learn *complete* ontologies from text. Although the field of ontology learning is gaining popularity day by day, most terminologists would probably argue that even the best ontologies produced automatically need heavy manual postediting depending on the complexity of the domain. Seeing as ontologies, whether generated manually or automatically, age rapidly, it can be equally useful to develop tools which update existing ontologies. Thus the system presented in this thesis makes no attempt at producing complete ontologies, but attempts only to discover, filter and apply knowledge patterns by means of the WWW in order to extract relation instances which can be used to *augment* ontologies.

Nevertheless, the following point about the direction of ontology learning research also applies to the work reported in this thesis.

> Ontology learning, in the Semantic Web context, is primarily concerned with knowledge acquisition from and for Web content and is thus

> moving away from small and homogenous data collections to tackle the massive data heterogeneity of the World Wide Web instead. [Buitelaar et al., 2005, p4]

To the extent that the challenges encountered in the empirical sections of this thesis are caused by the heterogeneity of the WWW, solutions and findings may thus also be useful for research on ontology learning.

2.1.1 Term and relation extraction

Term extraction is typically the first step in most AKA systems. Methods of automatic term recognition (ATR) are usually either linguistic, statistical or, more commonly, a mix of the two. Since the most common linguistic expression of specialized concepts is noun phrases (NPs), linguistic approaches (e.g. [Jacquemin, 1994]) rely on part-of-speech tagging and some degree of parsing to identify term candidates. Statistical approaches, on the other hand, often rely on an analysis of the degree of association between lexical units in the analysis corpus versus a reference corpus (see for example [Drouin, 2003] or [Gillam, 2004]). These analyses are based on contingency tables like the one presented in subsection 4.1.2.

ATR will be discussed no further at this point since its use in WWW2REL is limited to a very simple statistical technique of penalizing candidates which are overly frequent in a general language reference corpus (see subsection 5.3.1 for details). Also, ATR needs only be performed on one of the arguments in the binary relation instances extracted from the WWW, because the input term is fixed from start (see subsection 5.1.2).

Relation instance extraction involves the subtask of ATR, or Named Entity Recognition (NER), when carried out for a specific domain. The goal of relation instance extraction is

> [...] to detect occurrences of a prescribed type of relationship between a pair of entities of given types. While the type of the entities is usually very specific (eg genes, proteins or drugs), the type of relationship may be very general (eg any biochemical association) or very specific (eg a regulatory relationship). [Cohen and Hersh, 2005, p63]

Before we proceed further, it should be mentioned that some researches prefer to use the term "role extraction" for this task, as indicated by the following quotation.

> While there is much work on role extraction, very little work has been done for relationship recognition. Moreover, many papers that claim to be doing relationship recognition in reality address the task of role extraction: (usually two) entities are extracted and the relationship is *implied* by the co-occurrence of these entities or by the presence of some linguistic expression. These linguistic patterns could in principle distinguish between different relations, but instead are usually used to identify examples of *one* relation. In the related work for statistical models there has been, to the best of our knowledge, no attempt to distinguish

> between *different* relations that can occur between the *same* semantic entities. [Rosario and Hearst, 2004]

[Rosario and Hearst, 2004] are right that relation instance extraction, whether based on pattern matching or conceptual clustering, is perhaps more accurately described as role extraction because it is not the relation types *themselves* which are being recognized but rather pairs of entities which form instances of a *fixed* relation type in which each entity plays a particular role, for example "hyponym" or "hypernym" in the case of the ISA relation. Nevertheless, role extraction or not, the end product is meant to be new relation instances not registered in the target ontology, and if multiple sets of patterns are available multiple relation types can be recognized.

In a comprehensive survey of methods and tools for building ontologies from text [Gómez-Pérez and Manzano-Macho, 2005] identify three groups of AKA tools differing by purpose.

1. Identifying relations
2. Identifying concepts
3. Building up taxonomies or ontologies

The primary purpose of the present study is, in fact, a mix of all three purposes or tasks. WWW2REL finds relation instances and in this process identifies specialized concepts, but it also provides taxonomical and ontological fragments which can be pieced together by terminologists or perhaps by the system itself in future, more advanced versions.

2.1.2 Pattern-based approaches to relation extraction

AKA based on pattern matching is a well researched area. The patterns used in the retrieval of knowledge have been given various names, including

- semantic formulae [Lyons, 1977]
- lexico-syntactic patterns [Hearst, 1992]
- knowledge probes [Ahmad and Fulford, 1992]
- explicit relation markers [Bowden et al., 1996]
- knowledge patterns [Meyer, 2001]
- operators [Penagos, 2004]

Throughout this thesis the term "knowledge patterns" (or the acronym KP) will be used. It should be noted, however, that this term has also been used in a different sense in formal ontology. In formal ontology, a knowledge pattern has been defined as "a first-order theory whose axioms are not part of the target knowledge-base, but can be incorporated via a renaming of their non-logical symbols" [Clark et al., 2004, p196].

Patterns in the non-formal, natural language sense can be used to extract bits of knowledge from unstructured, free text and are characterized by the notion of surface semantics. This means that they can be intuitively interpreted and easily acquired. [Hearst, 1992] reports on the identification of a set of lexico-syntactic patterns which satisfy the following desiderata.

(i) They occur frequently and in many text genres

(ii) They (almost) always indicate the relation of interest

(iii) They can be recognised with little or no pre-encoded knowledge

She does not, however, present any empirical study of the extent to which desiderata (i) and (ii) hold. Such a study is provided in sections 6.3 and 6.5 of this thesis.

Knowledge patterns can be divided into

1. linguistic patterns
2. non-linguistic patterns

Examples of non-linguistic, or paralinguistic [Meyer, 2001], patterns are all kinds of punctuation marks, for example parentheses, and other symbols like equation signs, arrows and so on. This study will completely ignore non-linguistic patterns as most of these are ignored by the web search engines.

Among the linguistic patterns verbs and verb phrases (VPs) appear especially attractive because of their precision and their ability to identify terms (cf. [Barrière, 2001, Christensen, 2002]). However, for terminologically fundamental relation types like ISA and synonymy, many high recall patterns contain no verbs, for example "hypernym <such as> hyponym", "synonym1 <or> synonym2" and so on.

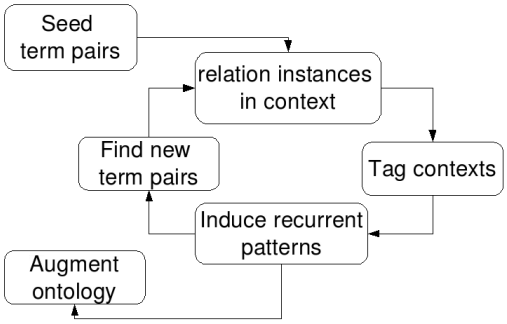
As discussed in section 1.1 using KPs to extract relation instances involves a number of natural language challenges, including polysemy, domain dependency and anaphoric reference. The issue of polysemy (i.e. KP ambiguity) is touched upon in section 6.3. The challenge of domain-dependency is an interesting problem which appears to be ignored by most research in this area. Section 6.5 presents an analysis of the domain dependence of KPs learned from the contexts of biomedical term pairs by examining their recall when applied to collections of IT text snippets. Finally, while anaphoric reference is an interesting challenge, covering it would involve the detection of inter-sentential semantic relationships, and this is simply not feasible when using short text snippets which are rarely more than one or two sentences long. Subsection 4.1.4 has details on the web corpora used in the thesis experiments.

Two additional challenges are

1. data sparseness
2. automatic evaluation

Although using the entire WWW minimizes the data sparseness problem, it may still be a problem when attempting to discover KPs using highly specialized term pairs from a domain like Biomedicine (see table 13 in subsection 4.1.3 for examples). As for the

Figure 1: The DIPRE technique



second challenge, automatic evaluation remains a largely unsolved problem both for ontology/taxonomy learning, but also for automatic relation instance extraction. The main problem is how to interpret new information which is not present in the gold standard ontology against which the automatic evaluation is performed. One option is to select for a specific relation type sets of target and non-target instances from the gold standard ontology and see to what extent the system can find the correct arguments given correct and incorrect inputs (see for example [Mukherjea and Sahay, 2006]). However, this is a suboptimal evaluation strategy for a system which is designed to *augment* existing ontologies with new relation instances, and for this reason the present study features a comprehensive manual evaluation performed by four domain experts. Subsection 2.1.3 has more on the evaluation challenge.

Dual Iterative Pattern Relation Expansion (DIPRE) Although the technique used in section 4.1 to discover KPs from text snippets on the WWW is not iterative, it is very much inspired by the Dual Iterative Pattern Relation Expansion, or DIPRE, technique introduced in [Brin, 1998], one of the founding fathers of Google.

[Brin, 1998] describes a unsupervised technique which extracts (author,title) pairs directly from unannotated web documents based only on a small set of five seed pairs. The basic idea of the DIPRE technique (visualized as figure 1) is that a tiny bit of existing semantic knowledge can be used to identify discoverrent patterns instantiating

similar bits of knowledge which in turn can be used to find more recurrent patterns and so on. Its iterative steps can be summarized as follows:

1. Start with a sample of seed term pairs (relation instances)
2. Find occurrences of these instances in context (e.g. on the WWW)
3. Induce recurrent patterns from the occurrences and their individual contexts
4. Use these patterns to find more term pairs (relation instances)
5. Augment database and repeat from step 2 with the extra instances.

Although the DIPRE approach to relation instance extraction is very promising, the main challenge is to control the expansion phase (step 4) so that the system does not drift too far from the starting point and starts extracting incorrect relation instances. The implementation of WWW2REL does not address this expansion issue, because it starts with a slightly larger set of seed instances (see section 4.1) and is able to extract plenty of patterns and subsequently new instances in a single pass due to the richness of data on the WWW.

2.1.3 The evaluation problem

The automatic evaluation of relation extraction systems is often carried out on a set of manually annotated documents. For systems designed to extract relations from the WWW manually downloading and annotating a set of web documents is impractical, because the very reason for searching on the entire WWW is to utilize the high precision of the pattern-based approach while compensating for its inherent data sparseness problems.

The Snowball application [Agichtein and Gravano, 2000], for example, is evaluated by extracting 13,000 <organization,location> pairs from an online resource and eliminating from this list those pairs which do not have a single co-occurrence in any test document and thus cannot possibly be extracted. Nevertheless, a problem with respect to an automatic evaluation of system precision remains.

> If the initial directory of organizations from Hoover's contained all possible organizations, then we could just measure what fraction of the tuples in Extracted are in Ideal (precision) and what fraction of the tuples in Ideal are in Extracted (recall). Unfortunately, a large collection will contain many more tuples than are contained in any single manually compiled directory. [...] If we just calculated precision as above, all the valid tuples extracted by Snowball, which are not contained in our Ideal set, will unfairly lower the reported value of precision for the system. [Agichtein and Gravano, 2000]

One solution is to evaluate system precision manually (as is done in this thesis by the four experts). The Snowball team suggests joining the set of ideal pairs with the set of extracted pairs on a unique key, in their case the organization name. In this

way <organization,location> tuples unseen in the gold standard are added to this standard before precision is computed. Additional problems complicate this procedure, namely variance and specificity. There may be multiple variants of an organization name (Microsoft, MS etc.) and locations can be more (Redmont) or less (California) specific. [Agichtein and Gravano, 2000] solve this problem by using the Whirl tool [Cohen, 1999] to conflate variant organization names and by accepting both locations at state/country and city level.

When extracting semantic relation instances for use in terminology work, however, additional evaluation problems arise. First of all, termhood is difficult to assess automatically, and there is no term tagger with an accuracy comparable to that of named entity taggers like Whirl. Thus one can expect more noisy output, in the sense that more relation instances will be considered invalid because their arguments are simply not specific enough and thus irrelevant to the terminologist. While it is correct that "haloperidol" is a drug, for example, "drug" is arguably too vague a hypernym to be useful to a terminologist building an ontology of central nervous system agents.

Another reason why evaluating performance may be more difficult when retrieving terminological knowledge than when retrieving facts in Information Extraction (IE) tasks is that the search space may be more open. While tuples like <country;capital> express one-to-one relationships, non-taxonomical conceptual relations like "induces" (from the UMLS) are many-to-many. For example, contaminated water may cause vomiting, but it may also cause other things, and vomiting can be induced by other substances than contaminated water (more on this example in subsections 5.5.3 and 5.5.4). The unboundedness of the search space makes it difficult to evaluate the precision and recall of systems which extract instances of many-to-many relationships. While the list of all countries and their capitals is fairly static and can easily be obtained, there is no exhaustive list of valid "induces" instances. Section 6.4 offers concrete examples of the issues raised here and attempts an automatic evaluation of system recall versus the UMLS.

2.2 Theories of specialized knowledge

The field of terminology is concerned with the acquisition, management and structuring of specialized knowledge. It is an open question, and indeed a source of the theoretical debate surveyed in this section, to what extent the properties of specialized knowledge differ so fundamentally from those of general knowledge that their description merits a completely distinct modus operandi and that the study of specialized knowledge as opposed to knowledge in general should constitute a scientific field in its own right. However, before this debate can be probed, a few basic concepts concerning the properties of knowledge in general need to be established in subsection 2.2.1.

2.2.1 What is knowledge?

Understanding the nature of knowledge is clearly not a trivial problem since philosphers have pondered this matter from the days of Plato and even before. In fact, an entire school of philosophical scholars known as "the skeptics" have proclaimed the impossibility of knowledge for centuries. As the following paragraphs will elucidate,

theories of knowledge are prone to be self-contradictory and ridden with circularities. However, these circularities and paradoxes mainly occur when studying existence from a top-down, philosophical perspective rather than the bottom-up, terminological perspective. Thus given the age-long successes of practical terminology and knowledge structuring, we will take the freedom of disregarding the rather extreme, albeit philosophically interesting, stance of the skeptics altogether.

Typically, knowledge is contrasted from opinion based on the strength of evidence. While one person might think that X is the case, another person can disagree, and the correctness of opinion must be determined by evidence in order to be classified as knowledge. Knowledge, then, could be defined as "justified true belief" [Orilia and Varzi, 1998]. Plato provided the first detailed theory of knowledge by introducing the notion of the Forms. Forms are mental represention of prototypical entities with idealized properties. Forms constitute the objects of our knowledge and exist independently of the objects, events or actions of reality, which they are supposed to represent. Moreover, unlike the dynamic sensory input constantly processed by our brains, forms are unchanging making it possible for knowledge to be stable.

Plato's pupil, Aristotle, made contributions to both metaphysics and epistemology which are important in the context of terminology. He argued that the human mind is capable of abstracting general concepts (not unlike Plato's Forms) from real world objects which share certain features or properties. This process of grouping instances into categories based on shared properties is a fundamental way of imposing order on an otherwise chaotic world. It is also the source of what he called "basic knowledge" which is, in turn, the prerequisite for all further knowledge.

Classical conceptual analysis Aristotle introduced the classical conceptual analysis which is "a proposition giving metaphysically necessary and jointly sufficient conditions for being in the extension across possible worlds for that concept"[8]. The notions of "extension" and "intension" were originally proposed by the German philosopher Gottfried Wilhelm Leibniz (1646-1716) and form part of the semantic triangle which is discussed in subsection 2.2.3. In short, the extension of a concept is the complete set of objects or entities to which the concept refers. The "necessary and jointly sufficient conditions", on the other hand, constitute the intension of the concept and determine the concept's extension. Analytical definitions are especially important in terminology, because their conceptual intensions pinpoint the exact and non-overlapping positions of the target concepts in a conceptual hierarchy.

An implication of the classical, or Aristotelian, conceptual theory is that every complex concept has a classical analysis. By complex concept we understand a concept which has an analysis in terms of other concepts, and the definition of a classical analysis was quoted above. A classical analysis of a complex concept has two components: the *analysandum* and the *analysans*. The former is the concept which is being analyzed, and the latter is the concept which acts as the vehicle of analysis. A "necessary and sufficient condition" for being a concept C is a condition which must hold for all members of C but at the same time a condition which necessarily implies membership of C. For example, to be a bachelor you must be an unmarried male and if you happen

[8]The Internet Encyclopedia of Philosophy, www.iep.utm.edu

to be an unmarried male, you are, in fact, a bachelor. Further conditions on classical conceptual analyses include the following.

1. classical analyses cannot be circular (*a bachelor is a bachelor)
2. the analysans of a classical analysis must be simpler than its analysandum
3. a classical analysis does not include any vague concepts in its analysandum or analysans[9]
4. the definition constraint of classical analyses implies a Substitutivity Principle whereby the analysandum and analysans are mutually substitutable [Orilia and Varzi, 1998]

Typical examples of analytical definitions in terminology thus have the format:

analysandum <ISA> analysans

analysans = genus proximum + differentia specifica

where *genus proximum* is the closest superordinate concept of the analysandum, for example "male" in the case of "bachelor", and *differentia specifica* are the characteristics which distinguish the analysandum from its cohyponyms, in this case the feature-value specification [marital status: unmarried].

From the viewpoint of classical conceptual analysis knowledge, then, can be defined as *justified belief in a definition*.

Pros and cons of classical conceptual analysis That classical conceptual analysis is computationally attractive is evidenced by a long-standing goal of the Artificial Intelligence (AI) field of analyzing complex concepts by means of universal primitives. The motivation for finding such universal conceptual primitives was that they might allow computer systems to decompose natural language strings into formal and language independent knowledge representations and either generate translations of the strings into other languages by means of this "interlingua" or apply logical inference to access information only implicitly present in the strings. However, rule-based machine translation systems and manually engineered expert systems have proven surprisingly fragile and only functional in highly specialized contexts. In short, they have met with the knowledge acquisition bottleneck mentioned in the introduction and have been replaced by, or at least augmented with, probabilistic and inductive knowledge acquisition systems. Nevertheless, classical conceptual analysis as such remains an extremely useful technique in practical terminology work and in mark-up languages like XML used for content representation.

While classical conceptual analysis is computationally attractive, objections to this approach are many. Generally speaking, most critics maintain that while some complex concepts can be analyzed by the classical approach other concepts cannot, for example many words represent vague or fuzzy concepts which are not characterized by binary membership conditions. In prototype theory (formulated in [Rosch, 1973] and popular

[9]The Internet Encyclopedia of Philosophy

in modern cognitive semantics) membership of a category is not a binary but a gradable property in the sense that while a three-legged cat does belong to the category of cats, it is a less prototypical example than a regular, four-legged instance of the species and so on. The implications these objections have had to terminological theory are summarized in subsection 2.2.4.

Paradoxes of inquiry and discovery While classical conceptual analysis has been hugely successful, it does involve what is known as Meno's paradox (from the dialogue by Plato), namely that the truth of a conceptual analysis entails its triviality.

> [...] our desire to know the analysis of [the concept] c cannot be satisfied unless we already know the analysis of c! But if we already know the analysis of c, we surely cannot learn it. So it appears that there is simply no way to learn the analysis of c. [Moffett, 2005]

In the context of linguistics, the paradox is known as "the paradox of language acquisition" and is illustrated by the following questions. How do we acquire the semantic primitives by which more complex concepts can be analyzed and understood, are these primitives truly universal and if so how many are there? One attempt at solving the paradox is the proposal of a "Universal Grammar" of which Noam Chomsky has been a modern advocate. A universal grammar involves the idea that a linguistic competence module containing a finite number of deepstructure rules is hardwired into the human brain. The existence of such a grammar would solve the problem afflicting decompositional approaches to semantics (e.g [Wierzbicka, 1992, Wierzbicka, 1995]), namely that the semantic primitives are themselves unanalyzable.

Critics of Universal Grammar and generative linguistics would argue that the only universal linguistic feature of the human brain is that it is equipped with a sophisticated pattern recognition module which allows a gradual and inductive generalization from large amounts of linguistic performance to a type of probabilistic grammar. Since the early 1990s inductive and probabilistic approaches to linguistics have been gradually replacing the generative approaches with the tangible successes of statistical NLP and data-driven Machine Learning (see section 3.1 for more on corpus linguistics).

In the context of computational terminology, the language acquisition paradox need not concern us. The paradox of discovery, however, still needs to be addressed. Relation extraction systems like the one implemented in this thesis do not miraculously extract new knowledge (i.e. new analyses) from collections of text. But as Plato asserts by his Theory of Recollection, while learning analyses of concepts does not in any sense bring forth new knowledge, it is not a meaningless activity because it does activate prior knowledge tacitly known by the individual or at any account by human society in general.

> [...] we do in fact know the analyses of most of our concepts; what we lack is *explicit, conscious access* [my emphasis] to those analyses. [Moffett, 2005]

In essence, what relation extraction systems try to do, then, is to help a terminologist (re)collect knowledge fragments in the form of linguistically instantiated semantic

relations which, when pieced together, may form an explicit and directly accessible conceptual analysis.

Types of knowledge In an article which argues that the alliance between terminology and knowledge engineering is perhaps not theoretically fruitful [Toft, 2000, p237] discusses the following knowledge dichotomy, originally introduced in [Oeser and Picht, 1999].

1. sphere
   (a) common sense knowledge (concrete)
   (b) specialized knowledge (abstract)
2. content
   (a) descriptive/declarative knowledge
   (b) procedural knowledge

Specialized knowledge, as opposed to common sense knowledge, requires a special language to be effectively communicated. While descriptive or declarative knowledge is knowing *that* something is the case, procedural knowledge is knowing *how* to accomplish a specific task.

The type of knowledge typically processed by terminologists is specialized rather than common, but it can be both abstract and concrete depending on the target domain. An example of abstract, specialized knowledge is the terminology of a domain like Computer Science, while the terminology of upholstery, for example, is specialized, but highly concrete. The knowledge of the Biomedicine domain which is the test case of this thesis is arguably more abstract than concrete.

As for the distinction between descriptive versus procedural knowledge, the classical research object of terminology is descriptive knowledge because it is easier to analyze this kind of knowledge conceptually by means of feature-value matrices and represent it in formal concept systems. Thus the knowledge sought by WWW2REL is descriptive rather than procedural.

Implicit and explicit knowledge The terms "implicit" (or "tacit") and "explicit" knowledge are mainly used in the field of knowledge management about organizational activities such as making sure that your employees codify important practices and work routines so these bits of implicit knowledge (typically of the procedural variety) can be rendered explicit, be disseminated and boost the productivity of the organization.

Nevertheless, the distinction can also be applied to characterize the nature of the descriptive knowledge sought by relation extraction systems like WWW2REL. Most of the knowledge expressed by means of natural language is, in fact, only tacitly or implicitly present, and implicit knowledge exists in virtually any natural language text. World knowledge and inference mechanisms allow human beings to derive much more information from the linguistic signs in a text than is explicitly stated. For example, given the short sentence

His wife and her students left the room.

a human being will be able to infer a long list of knowledge fragments (conceptual relations), none of which are actually *explicitly* present in the sentence. The following are just a few examples.

- ISA(wife,woman)
- ISA(wife,adult)
- PART_OF(room,building)
- HAS(wife,husband)

Decoding the complete semantic content of the sentences which follow may require that these conceptual relations, or characteristics, are retrieved from the set of ontologies stored in the fuzzy database known as the human brain. The example illustrates the need to explicate, or recollect, implicit knowledge, for example by harvesting conceptual relations from natural language text and representing them in ontologies (the topic of section 2.3). For Language for General Purposes (LGP) monumental ontologies, for example WordNet[10], have already been manually produced, but for Language for Special Purposes (LSP) the availability of ontological resources depends on the subject field. However, specialized ontologies tend to change at a much faster pace than general ontologies (due to the exponential technological progress of modern science), and thus the need to automate their construction is much more pronounced.

In conclusion, the focus of this thesis is exclusively on the automatic acquisition of *specialized*, *descriptive* and *explicit* knowledge in the form of semantic relation instances.

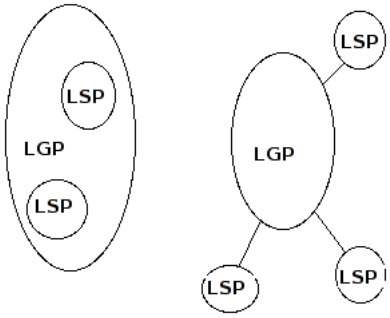
2.2.2 LSP and LGP

Specialized knowledge is typically communicated by a Language for Special Purposes, or LSP. The acronyms LSP and LGP (Language for General Purposes) often demarcate not only two distinct functions of natural language but also two distinct fields of linguistic research. Since the research object of terminology is specialized knowledge, the distinction between LSP and LGP mirrors the scholarly divide between the fields of terminology and lexicology.

However, as is usually the case with dichotomies, one can always ask the unorthodox question: How distinct is LSP really from LGP, or more specifically, what are the main features which distinguish the two? The key question is whether LSP and LGP are part of the same language system or not. Figure 2 visualizes the "unicentric" versus the "pluricentric" view of LSP [Kragh, 1995]. If one adopts the unicentric view of LSP, then the distinction is a purely quantitative one. Certain structures occur more frequently in certain varieties of LSP than in the language as a whole, but there are no qualitative differences between the two functions of language, no linguistic features

[10] http://wordnet.princeton.edu

Figure 2: Unicentricity vs. pluricentricity



which are unique to a certain LSP, for example. In fact, LSP and LGP should be considered "not as different forms of language, but as different ways of using the same language" [Widdowson, 1974].

The corpus linguistics point of view (see section 3.1 for more details) is clearly unicentric. The borderline between LSP and LGP is not a clear-cut one, but rather a continuum. Any text can be analyzed linguistically and statistically for a number of linguistic features (for example pronoun-noun ratios) which are known to be characteristic of a range of communicative settings. Depending on how a text is positioned along a number of these dimensions, for example "involved versus informational" or "narrative versus non-narrative" [Biber et al., 1998, p148], it can be categorized as being more or less LSP'ish. This kind of multidimensional analysis of linguistic features has become the bread and butter of automatic text categorization.

The ratio of explicit versus implicit knowledge in a given text can be another feature with which to characterize LSP and LGP. This ratio varies depending on the communicative purpose and setting of the discourse. LGP will tend to have a lower density of explicit knowledge, because fully decoding LGP messages typically only requires knowledge of very general ontologies which can be presupposed of any reasonably educated adult human being familiar with the given language in which the message is encoded. On the other hand, LSP aimed at intermediates or novices, tends to have a higher density of explicit knowledge, because a proper decoding is only possible given knowledge of much more specific (sub)-ontologies which the author (or sender) does not expect of his readership (or receivers). However, LSP in an expert-expert setting will have a lower density of explicit knowledge.

Academic language Empirical analyses of a range of LSP versus LGP corpora have, in fact, suggested the existence of a third, intermediate use of language, namely a kind of basic scientific vocabulary realized by families of Academic Words [Coxhead, 2000]. In his corpus-based analyses, [Coxhead, 2000] compiled a balanced Academic Corpus of approximately 3.5 million words representing a wide range of scientific domains and genres. He defined the following three criteria which must be met by an academic word.

Table 1: Academic word families

head word	word forms
modify	modification(s)
	modified
	modifies
	modifying
	unmodified
category	categories
	categorize
...	...

1. it must not occur among the 2,000 most frequent words of English
2. it must occur at least 10 times in each of four corpus sections[11] and in 15 or more of 28 subject areas
3. it must occur at least 100 times in the Academic Corpus

Table 1 lists two examples out of the 570 academic word families identified by [Coxhead, 2000]. As will be evidenced by the manual evaluation of WWW2REL in chapter 5, this kind of specialized, but domain independent, vocabulary also represents vague or fuzzy concepts which are not helpful when extending an existing terminological ontology.

What we are looking for when extracting domain-specific concepts are typically noun phrases (NPs). Based on extensive empirical investigations the Longman Grammar of Spoken and Written English (LGSWE) observe that almost 60% of all NPs in a corpus of academic prose have a modifier, either premodifier, postmodifier or both. Whereas this is only the case for 15% of the NPs in a corpus of conversation and 30% in fiction [Biber et al., 1999, p578]. Moreover, on the discourse distribution of NP types in academic prose, the LGSWE study reveals that

> Although it is by no means an absolute rule, repeated references to an entity tend to follow the same progression of noun phrase types across texts: N + postmodifer -> premodifier + N -> simple noun -> pronoun [Biber et al., 1999, p586]

An example of this phenomenon taken from the domain of Biomedicine is given in table 2.

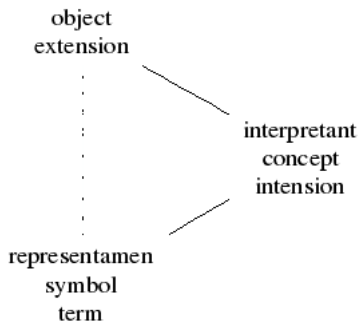
The tendency of using increasingly compact expressions to refer to the same concepts has an impact on the implementation of WWW2REL outlined in subsection 5.1.2. Since the system makes no attempts at resolving anaphora, the final stage of identifying the reference of pronouns is completely ignored. However, an attempt is made at conflating the two NP types representing the first and second references to a given entity in a discourse. The system also handles the third reference by grouping NPs by their head.

[11] Arts, Commerce, Law and Science

Table 2: NP compression in academic writing

	NP
1st mention	bleeding in the stomach
2nd mention	stomach bleeding
3rd mention	bleeding
subseq. mentions	it

Figure 3: The semiotic triangle



2.2.3 Termhood and the semiotic triangle

When attempting to extract fragments of specialized knowledge from the entire WWW, the distinction between what is specialized and what is common (or general) becomes important since the WWW is a repository of both types of knowledge (see subsection 3.1.1). Essentially, the problem is all about determining the degree of termhood of the arguments of the relation instances which are retrieved. As described in the introduction, when looking for bits of specialized knowledge, knowledge patterns are considered "noisy" if they lead to the identification of a general language, fuzzy concept. But what is meant by fuzzy? What is the difference between terms and words and thus the difference between concepts referred to in specialized versus general language? This question of termhood is addressed in the following.

The semantics of lexical items, whether functioning as words or terms, can be explained by semiotic models. The most famous of these are Ferdinand de Saussure's (1857-1913) dualistic model of the linguistic sign and Charles Sanders Peirce's (1839-1914) triadic model of signs in general. Saussure defined linguistic signs as a link between two mutually dependent parts, namely a signifier (a sound pattern) and a signified (a concept), and asserted that this link is (ontologically) arbitrary, essentially echoing the conventionalist stance taken by Hermogenes in Plato's Cratylus (language works by pure convention). It should be noted that the relation between signifier and signified is often only *relatively* arbitrary as can be observed in the semantics of noun compounds which are usually a non-arbitrary juxtaposition of individual signs.

Peirce's model of signs (see figure 3), on the other hand, has three components: an

object, the representamen (or symbol) and an interpretant (or concept). Unlike Saussure's signified, Peirce's interpretant is itself a sign in the mind of the interpreter, and the interpretation of a sign can thus involve a theoretically infinite chain of intermediary signs. Cascades of semantic triangles are an inherent feature (and paradox) of human language and cognition because the semantic content of linguistic signs can only be made explicit by relating these signs to other signs and so forth.

In terminology the expressions extension, intension and term are typically used instead of object, interpretant and representamen, respectively. Terms need not have an extension (for example astrophysical "worm holes" may not exist) and intensions need not have a term (for example newly created notions in the mind of a scientist), but the extension and intension of a concept will always be inversely related to each other. The more features expressed in an intension, the less objects or referents will be included in its extension and vice versa. As terminological ontologies will typically contain concepts having relatively rich intensions augmenting these ontologies may involve data sparseness problems as described in chapter 4.

For terminology [Suonuuti, 1997] proposes the definition as a fourth component turning the triangle into a pyramid. The definition has no direct bearing on the term or its extension, but it pinpoints the exact position of the concept in a conceptual hierarchy by describing the necessary and sufficient characteristics of the concept (see also subsection 2.2.1) which will include essential relations to other concepts in the hierarchy. The specificity of reference thus seems to be the property which grants termhood status to a lexical unit. In other words, terms are lexical units which can be defined analytically and be dissociated from context without losing referential specificity. Nevertheless, termhood criteria have been debated by many scholars, and the following are but a few of numerous definitions of the term.

1. a verbal designation of a general concept in a specific subject field (ISO 1087-1/ISO 12620)

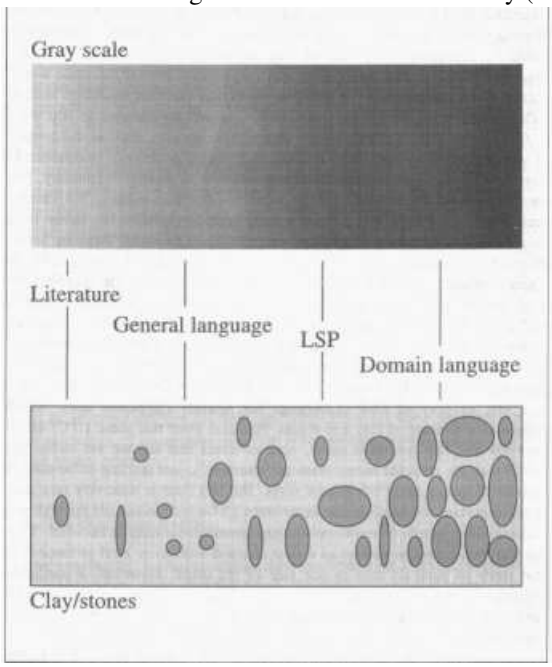
2. the terminological unit represents a concept, uniquely and completely, taken out of any textual context. The existence of this one-to-one relationship between a linguistic expression and an extra-linguistic object is a situation which particularly concerns the terminological units. [Bourigault, 1992]

3. It can be argued that a term is a place-holder for its definition, which in turn is a place-holder for a concept: in standardized terminology the definition should follow the principle of substitutability for exactly this reason. [Gillam, 2004, 74]

4. Terms signify concepts belonging to a specific subject field [Madsen, 1991, p84]

What distinguishes a term from a word is thus supposedly the monosemy and contextual independence of the former. The reference of words, as opposed to terms, is typically so general and imprecise that they become ambiguous if they are not accompanied by considerable amounts of context. However, the modal "should" in the third definition indicates a possible gap between linguistic reality and terminological desiderata.

Figure 4: Terms as stones in clay (from [Melby, 1995])



I would argue that contextual independence is not a characteristic which delimits terms from words, at least not in all domains. In some domains, for example Information Technology (IT), terms are often formed by semantic, in fact metaphorical (cf. [Meyer, 1997]), extension of existing, general language lexical units. They thus require contextual specification when used in communicative situations where the domain context is not fixed. IT terms like "window", "desktop", "menu", "icon" and so on do not represent unambiguous concepts when taken out of any textual context, but must be accompanied by a domain label to secure their termhood (cf. [Halskov, 2005d, Halskov, 2005a]). Thus the first and the fourth definitions appear to be the most viable ones.

Approaching the nature of termhood from a Machine Translation angle, Alan Melby realized that using "a list of universal, language-independent sememes" was problematic because "the concepts of a narrow domain and the concepts of general language are of a fundamentally different nature" [Melby, 1995, p50]. "Lexical units and terminological units are both derived from sequences of characters from the same writing system, but neither is a subset of the other". "A word is thus a chunk of pliable clay and a term is a hard stone" [Melby, 1995, p52]. Melby elaborates on his stone-clay analogy (figure 4) by admitting that terms can be *superficially* ambiguous, a phenomenon known as term variants, but stressing that only words are *fundamentally* ambiguous.

While Melby's analogy is very apt, it does perhaps downplay the extent to which the terminology of some domains is affected by "determinologization". This phenomenon has been defined as

> [...] the ways in which terminological usage and meaning can 'loosen' when a term captures the interest of the general public. [Meyer, 2000, p12]

When large numbers of non-experts start referring to their desktop computer cabinet as their "hard disk", for example, one might indeed say that the former stone is starting to act like clay, at least in non-specialized communicative contexts.

But does this mean that terms can be polysemous? Bodil Nistrup Madsen argues as follows.

> One term corresponds to one concept (intension, sense), i.e. terms are never polysemous, only expressions are polysemous [...] In the same way it may be argued that only expressions, not terms, can be homonyms. [Madsen, 1991, p84]

The above erroneous usage of the expression "hard disk" and the coincidence that the string "window" may be used to refer to both physical and digital windows, does not cause the two terms to be polysemous. In the domain-specific context when IT experts convey information about their domain, "window" will always refer to the digital representations occurring on computer monitors and "hard disk" will never refer to desktop cabinets. Even if two different concepts happen to have the same label, they are still treated as two unrelated entities in a termbase. However, when searching on the WWW for semantic relations (see subsection 3.1.1), the communicative context is not fixed and determinologization can be a very real problem (see also section 6.5).

2.2.4 Schools of terminology

The topic of this section is the research object of terminology as viewed by the following three distinct terminological schools.

1. General Theory of Terminology (GTT)
2. Communicative Theory of Terminology (CTT)
3. Socio-cognitive Theory of Terminolgy (SCTT)

The expression "terminology" has multiple meanings, usually one of the following.

1. a set of terms belonging to a special language
2. a body of knowledge about any set of terms, the concepts they represent and the relations between these
3. the practical task of identifying and processing 1) so as to arrive at 2)
4. the structuring of 2) in conceptual hierarchies (ontologies)

However, the big issue, theoretically speaking, is whether one might not replace "terms" with "words" and reach the conclusion that there is no difference between the fields of lexicology and terminology, or rather that terminology must be regarded as a subfield

of lexicology and not a scientific field in its own right. The debate about the theoretical foundations of terminology can be traced back to Saussure's dualistic model which gives rise to two perspectives from which linguistic signs can be viewed, namely the semasiological and onomasiological perspective. An argument supporting the claim that terminology is indeed a scientific field in its own right is that its perspective is onomasiological, i.e. the focal point is the conceptual side of linguistic signs rather than the usage of these signs themselves.

General Theory of Terminology (GTT) Traditional terminological theory, best known as *The General Theory of Terminology* (GTT), is summarized in Wüster's *Einführung in die allgemeine Terminologielehre und in die terminologische Lexikographie* [Wüster, 1991] published posthumously in 1979. According to GTT, terminology is distinct from lexicology in three respects.

First of all, the modus operandi of terminologists should be purely onomasiological (as opposed to semasiological). GTT stresses the primacy and the autonomy of the concept as part of universal conceptual structures (ontologies). Concepts are the starting point, which is followed by a naming (labelling) process that must ensure unambiguous reference. In lexicology the point of departure is the lexical unit and the mapping of its semantic structure. A consequence of the onomasiological perspective is that phenomena like synonymy have been largely neglected because this is a relation between lexical items rather than between concepts.

Secondly, terminologists focus on vocabulary and consider morphology, syntax and other aspects of linguistic performance to be largely irrelevant.

Finally, it is argued that the semantic content of terms should be protected (invariant) through standardized usage so that terms are characterized by a 1:1 mapping between term and concept, and this mapping is specified by means of the classical analytical definition (see subsection 2.2.1). If these principles are adhered to, terms are clearly different from words which by nature are polysemous and exposed to semantic shifts over time.

It seems intuitively true that the chronology of term formation has the following order.

1. conceptual innovation
2. term creation
3. definition (mapping from term to concept)
4. standardization (elimination of ambiguity in the mapping of 3).

However, the final step of standardization is very hard, if not impossible, to achieve in reality, as speakers are free to use terminology as they please. The main discussion in newer theories of terminology, in fact, is whether all terms should be treated *as if they were standardized* and whether various linguistic, cognitive and social contexts of term usage should not also be investigated.

Communicative Theory of Terminology (CTT) Communicative Theory of Terminology (CTT) is an example of a terminological school which argues that the linguistic and communicative aspects of terminology should be investigated more carefully.

In [Cabré, 2000, Cabré, 2003] it is claimed that the research object of terminology is not concepts, nor terms but rather Terminological Units (TU).

> At the core of the knowledge field of terminology we, therefore, find the terminological unit seen as a polyhedron with three viewpoints: the cognitive (the concept), the linguistic (the term) and the communicative (the situation) [...] each one of the three dimensions, while being inseparable in the terminological unit, permits a direct access to the object. [Cabré, 2003, p187].

Although GTT accounts for one dimension of the terminological polyhedron, namely the conceptual one, it fails to fully consider the other dimensions. This does not mean that GTT is flawed, because TUs are such complex and multidimensional phenomena that they can hardly be accessed on all fronts at once. It does mean, however, that GTT can only be an ancillary component in a more comprehensive theory which, according to Cabré, is just beginning to emerge.

While J. C. Sager did not use the name “Communicative Theory of Terminology”, he discussed the communicative aspects of specialized discourse a decade before Cabré. He argues that a message can be defined as "the totality of intention, assumed expectation, knowledge content and language selected by the sender" [Sager, 1990, p100]. The extent to which the recipient is able to decode the knowledge content and the intention of the message, that is the text and its purpose, will be decided by three important factors, namely

1. precision
2. appropriateness
3. economy of expression

The highest degree of precision of a term could be achieved by using the complete definition of the concept it represents. However, this practice would generate a large number of lengthy and complex terms which would violate the principle of economy of expression. According to this principle, compactness of realisation (see also the example in table 2), for instance by acronymy, abbreviation or ellipsis, is to be desired in efficient specialist communication, but these techniques may reduce precision if the interlocutors fail to remember the full form. Arriving at expressions which are both economical but also precise is quite a balancing act, and the key to success is the third constraint, namely appropriateness.

Appropriateness is essentially a pragmatic criterion, a communicative norm or set of conventions which has been gradually established over time through frequently repeated special speech acts. Appropriateness decides which definitions can be presupposed in a given communicative context and thus "also decides the degree of general and special reference required in the individual speech act" [Sager, 1990, p112]. It is,

in other words, a question of convention and pragmatic context whether a long definition (partly phrased in words with general reference) or a compact term, which can be considered a substitute label for the same definition, happens to be used. In cases where "the recipient has neither the lexical nor the conceptual resources" [Sager, 1990, p114] precision and appropriateness of expression may take precedence over economy. This will presumably often be the case in textbooks or popular science magazines where domain-specific concepts have to be explained to non-experts. Such domain-specific, but "non-economical" texts are rich in explicit knowledge and thus ideal targets for pattern-based relation extraction systems like WWW2REL.

Socio-cognitive Theory of Terminology (SCTT) Unlike CTT SCTT does not criticize GTT for ignoring the linguistic or communicative aspects of terms, but it claims that the treatment of the conceptual side of terms in GTT is idealized. SCTT claims that there is often no clear separation between general and specialized knowledge in human cognition. In other words, it questions the premise that the concepts represented by terms are really fundamentally different from the meanings represented by words.

While the GTT approach must be counted among the positivist or objectivist theories of science, SCTT, as advocated in [Temmerman, 2000], is a hermeneutic or experientialist theory. The premise in experientialism is that reality does not exist independently of the perceiving subject. All knowledge comes from experience, and meaning cannot be completely objectified because it always involves a subject and is perceived and expressed through an inescapable filter (natural language). [Temmerman, 2000] thus claims that terms, more often than not, represent categories (or "notions" in the terminology of [Sager, 1990]) which are as fuzzy and dynamic as those represented by words. She argues that clear-cut concepts, which are not prototypical to some extent, are extremely rare outside of exact sciences like Mathematics and Chemistry [Temmerman, 2000, p223]. The analytical (intensional) definitions used in GTT are thus often inadequate because prototypical categories with gradable membership cannot be understood in a logical or ontological structure.

In SCTT one speaks of Units of Understanding (UU) rather than of concepts. These UUs typically have prototype structure and are in constant evolution. UUs can rarely be intensionally defined but should be interpreted by means of "templates of understanding" which are composed of different modules of information depending on the receiver and the context.

Criticism of GTT A proponent of CTT argues that

> Wüster developed a theory about what terminology should be in order to ensure unambiguous plurilingual communication and not about what terminology actually is in its great variety and plurality [Cabré, 2003, p167]

Another critic of GTT has claimed that

> in the traditional theory of terminology, there is not a single explanation of the formal relationship between 'concept' and 'terms' which makes it essentially different from the relationship between meaning and words in general linguistic semantics. [Kageura, 2002, pp21-22]

That terms are formally indistinguishable from words makes Kyo Kageura speculate that termhood is really an "aspectual category" [Kageura, 2002, p26]. If this is the case, it is an argument against the pluricentric view of LSP (see figure 2).

> Most of what currently passes for a theoretical foundation of terminology amounts to little more than a simplified, a priori theory of conceptual structures supported by largely prescriptive principles of what "should be" rather than what is actual usage of terms. [Kageura, 2002, p1]

By using conceptual structure as the basis for a theory of terms one runs the risk of building a theory of something which can be used to describe terms rather than a theory of terms themselves.

Like Kyo Kageura, Jennifer Pearson also points to the infelicities of traditional terminological theory. She criticizes GTT for conjuring up "pure terminologies" which are idealized and out of touch with reality. She argues vehemently against this notion by stating that "it is futile to propose differences between words and terms without reference to the circumstances in which they are used" and that we need to consider "what happens when terms are actually used in text rather than simply as labels for concepts in knowledge structures" [Pearson, 1998, pp7-8]. Language users, including specialists, often violate usage standards either because they are unfamiliar with them or because they need to use new terminology which has not yet been standardized.

In defence of GTT Although the GTT approach to terminology and knowledge representation has been accused of being minimalistic because concepts are regarded as clear-cut and non-overlapping nodes whose ontological position can always be unambiguously identified by means of analytical definitions, it has a number of obvious advantages. First of all, it allows the generation of unambiguous, formal ontologies (see subsection 2.3.2) which can be used for making logical inferences in advanced AI systems, for example. Secondly, it has proven its worth for several years now in practical terminology work as an ideal tool for conceptual clarification in virtually any domain.

A counter-argument against the GTT criticism voiced by [Temmerman, 2000] is that even if concepts are fuzzy, it is still possible to use analytical definitions and build ontologies which are not fuzzy. The only complication is that several definitions (and thus several ontologies) may arise when dealing with fuzzy concepts. But it is, in fact, typical of terminology work that there is more than one interpretation or perspective on the knowledge of a subject field. The terminologist will then simply construct one ontology for each interpretation and select the most appropriate one or try to standardize the fuzzy concepts as described in the following quote.

> Both in the case of ambiguous descriptions and in the case of different understandings it may be useful - as a basis for concept clarification - to set up several versions of an ontology representing the different views. In the case of descriptive terminology work, it would perhaps be relevant to set up different ontologies in which the superordinate concept of *blotting* [emphasis as in original] and the characteristics vary. In the case of

> normative terminology work, however, the terminologist - in co-operation with subject specialists - will have to decide upon one superordinate concept and one delimiting characteristic. *This does not mean that the different views or understandings could not be presented in the final result of the terminology work.* [...] *One very important thing to stress is that, no matter which solution is chosen, none of the different versions of the ontologies will contain indeterminacy, but it will be possible to describe indeterminacy of language by comparing different solutions* [my emphasis]. [Madsen, 2007]

2.2.5 Theoretical stance of the thesis

In conclusion, the source of the heated debate on the scientific foundations of terminology appears to be a basic difference of purpose. Is our purpose to ensure unambiguous and efficient communication between experts (i.e. normative), or is it to document the myriad of ways in which specialized knowledge can be expressed in natural language (i.e. descriptive)? Or is it even to understand what goes on inside the heads of people when they process, store and communicate specialized knowledge expressed by means of terms?

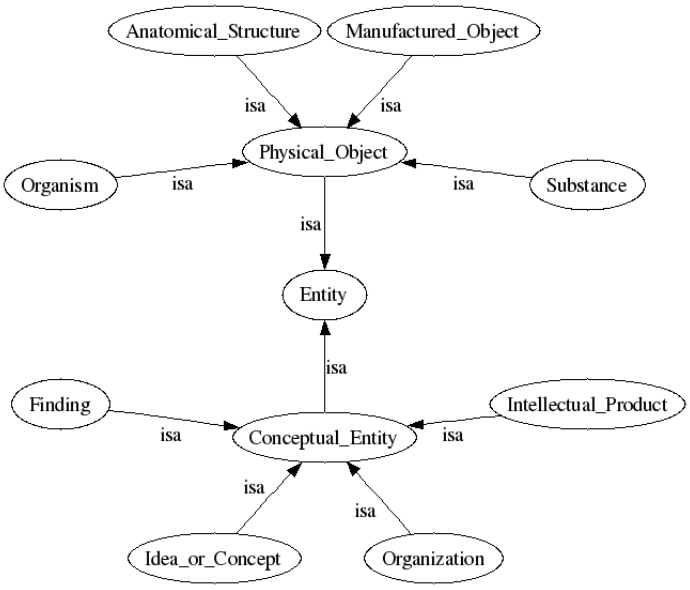
Although they disagree on the way concepts should be represented, GTT and SCTT are both purely conceptual approaches to terminology and knowledge structuring. The approach adopted in this thesis is purely inductive and descriptive and manipulates the linguistic rather than the conceptual side of terms. It thus differs from GTT which is a conceptual and *primarily* prescriptive framework and from SCTT which is a conceptual and, perhaps, more descriptive theory. While it shares the descriptive approach of CTT, it differs from CTT in that WWW2REL is a text mining system which makes no attempt at analyzing the larger units of discourse but operates at the more minute level of the individual sentence or even sentence fragment. Nevertheless, WWW2REL is designed to be a useful tool for terminologists which may organize its output in ontologies using GTT principles, for example.

2.3 Knowledge representation

Although (pattern-based) AKA is the main research focus of the thesis, this section gives a brief account of the subsequent step of knowledge structuring and knowledge representation. The reason such an account is deemed pertinent is that the starting point of the relation extraction system is a set of seed relation instances from an existing ontology (in this case the UMLS Metathesaurus). Consequently, basic knowledge about the properties and the structure of this ontology, but also of ontologies in general, will be useful. The cursory overview of the field of ontology and knowledge representation presented in this section is also needed in order to be able to discuss the choice of UMLS relation types used in the empirical experiments and system evaluation in chapters 4 and 5. Finally, the section aims to define how *terminological* ontologies differ from linguistic, philosophical and other types of ontologies.

Section 2.2 included a survey of the theoretical schools of terminology, and it was concluded that the main aim of classical terminology (General Theory of Terminology

Figure 5: A fragment of the "Entity" subontology in the UMLS Semantic Network



or GTT) is normative in that its raison d'être is to ensure unambiguous communication between domain experts through conceptual clarification, standardization and representation. In practical terms, the GTT methodology is to identify the essential characteristics (which may be semantic relations) of the key concepts of the target domain, pinpoint their closest superordinate concepts and by an iterative process define the ontological positions of these key concepts (relative to each other). The resulting structure is a knowledge representation. These structures are also known as ontologies or concept systems, and they can be visualized as a network of inter-connected nodes with concept labels[12]. Figure 5 is a visualization of a small fragment of the *Entity* subontology which is part of the UMLS top ontology (the Semantic Network).

Generally speaking, the way knowledge is stored and represented depends on the particular purpose for which the knowledge has been compiled. Visualizations like the one in figure 5 can serve as intermediate representations which may be helpful as part of a conceptual clarification and structuring process. To be really useful in real world applications like AI or IR systems, a more formalized structure including, for example, a translation of concept definitions into a logical language or a complete mapping between individual concepts and their term variants would be necessary. Typically, the structure would also be stored in a database so as to facilitate quick retrieval of (sub)sets of information.

[12]in terminological ontologies the labels need not be terms which are actually used in natural language

2.3.1 Studying existence

The science of ontology has a single, but ambitious goal, namely the study of existence. Not surprisingly, its history is both long and intertwined with fundamental scientific disciplines like Epistemology and Philosophy (cf. subsection 2.2.1). This subsection will not ponder the philosophical aspects of ontology, but merely highlight a few aspects of knowledge representation which are important in the context of modern ontologies like the UMLS Metathesaurus, for example.

The following two definitions of "ontology" are taken from [Buitelaar et al., 2005] and [Sowa, 2000], respectively.

> An ontology is an explicit, formal specification of a shared conceptualization of a domain of interest, where formal implies that the ontology should be machine-readable and shared that it is accepted by a group or a community. [Buitelaar et al., 2005, p3]
>
> The subject of *ontology* is the study of the *categories* of things that exist or may exist in some domain. The product of such a study, called *an ontology*, is a catalog of the types of things that are assumed to exist in a domain of interest D from the perspective of a person who uses language L for the purpose of talking about D. [Sowa, 2000, p492]

While *philosophical* ontologies categorize things which exist, *terminological* ontologies also model a whole range of non-generic relationships, primarily between domain-specific concepts. In this sense, the second definition of ontology quoted above is perhaps a little restrictive, because it seems to limit the field of ontology to building taxonomies of categories. On the other hand, the definition is interesting because it underscores the significance of the variable of language. Ontological structures are not necessarily language independent, since the speakers of language A may have conceptualized some parts of reality differently from the speakers of language B. Some ontological categories in a domain D and the language A may simply not exist in language B conceptualizations of the same domain or they may have a wider range of subtypes in A than B, for example.

Irrespective of the language variable, however, all human beings have the ability to abstract from the infinite details of sensory input simplified mental representations by grouping together instances observed in reality into prototypical categories on the basis of certain shared essential characteristics. Or perhaps rather on the basis of certain essential, but *different*, characteristics.

> All perception begins with contrasts: light-dark, up-down, hard-soft, loud-quiet, sweet-sour. Such contrasts [...] are the source of distinctions for generating the categories of existence [...] The contrasts, which relate the categories and determine whether a particular entity belongs to one or another, are more fundamental than the categories themselves. [Sowa, 2000, pp68-69]

The contrasts mentioned by [Sowa, 2000] can be represented as feature specifications, i.e. feature-value pairs in a feature-value matrix. The contrast between "hard" and

"soft", for example, could be possible values for a feature called "flexibility", and as visualized in the two feature-value matrices below, "flexibility" could be the one feature which distinguishes the two co-hyponyms "floppy disc" and "hard disc" from each other, i.e. the "subdividing dimension" in the terminology of [Madsen et al., 2005b].

$$floppy\ disc \begin{bmatrix} ISA: & data\ storage\ device \\ medium\ type: & magnetic \\ medium\ flexibility: & soft \end{bmatrix}$$

$$hard\ disc \begin{bmatrix} ISA: & data\ storage\ device \\ medium\ type: & magnetic \\ medium\ flexibility: & hard \end{bmatrix}$$

Contrasts, distinctive features or subdividing dimensions are an essential tool when clarifying key concepts in a particular domain so as to arrive at an optimally specified, consistent and unambiguous knowledge representation structure for the domain. The concepts represented by term variants can be decomposed into feature-value matrices, superordinate concepts can be identified (in this example the hypernym "data storage device") and co-hyponyms like "floppy disc" and "hard disc" can be distinguished from one another by identifying a subdividing dimension for which the two concepts have differing values. A dimension like "flexibility" could likely be used to distinguish a number of other co-hyponyms from eachother (presumably in a range of other domains than IT), and this explains why [Sowa, 2000, pp68-69] emphasizes that contrasts are more fundamental than the categories they can be used to define.

In the context of Biomedicine, which is the case study for which WWW2REL is tested in chapters 4 and 5, the "may_prevent" relation type is important because it may, for example, provide the distinctive feature of a class of central nervous system agents called "analgesics" as illustrated by the following feature-value matrix.

$$analgesic \begin{bmatrix} ISA: & central\ nervous\ system\ agent \\ may\ prevent: & pain \\ ... & ... \end{bmatrix}$$

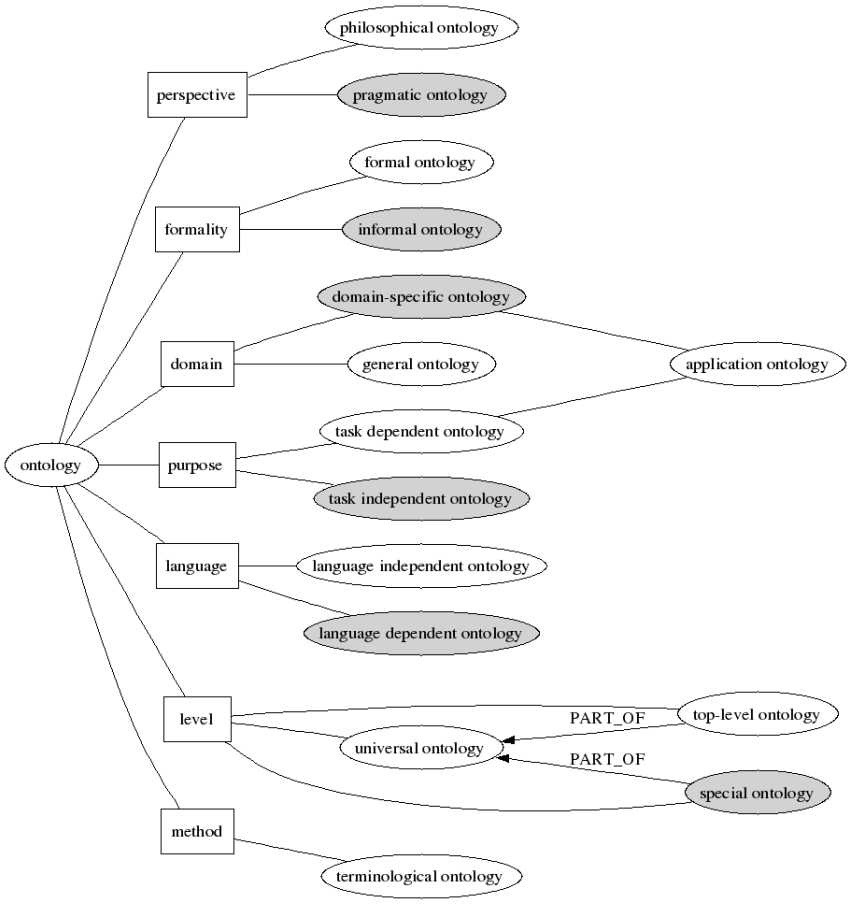
2.3.2 The terminological ontology

The purpose of this subsection is to discuss the properties of terminological versus other kinds of ontologies and to classify the UMLS Metathesaurus by means of a set of dichotomies which are introduced and discussed.

In her article "Alting på sin plads og plads til alting"[13] [Madsen, 2000] Bodil Nistrup Madsen presents a typology of ontologies based on a comprehensive literature survey. The typology is replicated in figure 6 in which the node labels have been translated from Danish into English and the filled circles indicate the feature values of the UMLS Metathesaurus.

[13]Own translation: "A place for everything and room for it all"

Figure 6: Typology of ontologies



While general (and philosophical) ontologies usually model reality from the top down using very superordinate and somewhat fuzzy concepts (or rather categories), the level of terminological ontologies is "special" in the sense that they typically model reality from the bottom up using highly specialized and clear-cut concepts. That the UMLS Metathesaurus is a "special" ontology explains why an ATR-like heuristic is introduced in WWW2REL in subsection 5.3.1.

As for the dichotomy based on the dimension of domain, even if the concepts represented in an ontology do belong to a specific domain, it may not be a terminological ontology in the strict sense of the word. According to [Madsen et al., 2004] neither WordNet, nor the UMLS Metathesaurus, would be classified as truly terminological ontologies, because neither ontology meets the following three principles.

1. Uniqueness of dimensions
2. Uniqueness of primary feature specification
3. Grouping by subdividing dimensions

The three principles have been implemented by means of automatic consistency checks in a strictly onomasiological ontology editor called Computer-Aided Ontology Structuring, or CAOS [Madsen et al., 2005a]. A screendump from CAOS2 can be seen in figure 7, and this example will be used to illustrate the principles behind *terminological* ontology building. The first principle means that a subdividing dimension[14] can only be associated with one concept in a particular ontology[15]. The second principle means that a particular primary feature specification[16], i.e. a feature specification which is not inherited from superordinate concepts, can only appear on *one* of the daughters of the concept containing the dimension in question. According to this principle, the primary feature specification, [striking technique: front], on concept 1.1.1 in figure 7 cannot also appear on concept 1.1.2. Finally, the third principle means that subdividing dimensions cannot overlap. In other words, concepts 1.1.1 and 1.1.2 can have "only one feature specification containing as an attribute the subdividing dimension of the mother concept" [Madsen et al., 2004], i.e. the attribute "striking technique" in the example. The benefits of adhering to the three principles are structural simplicity and logical consistence.

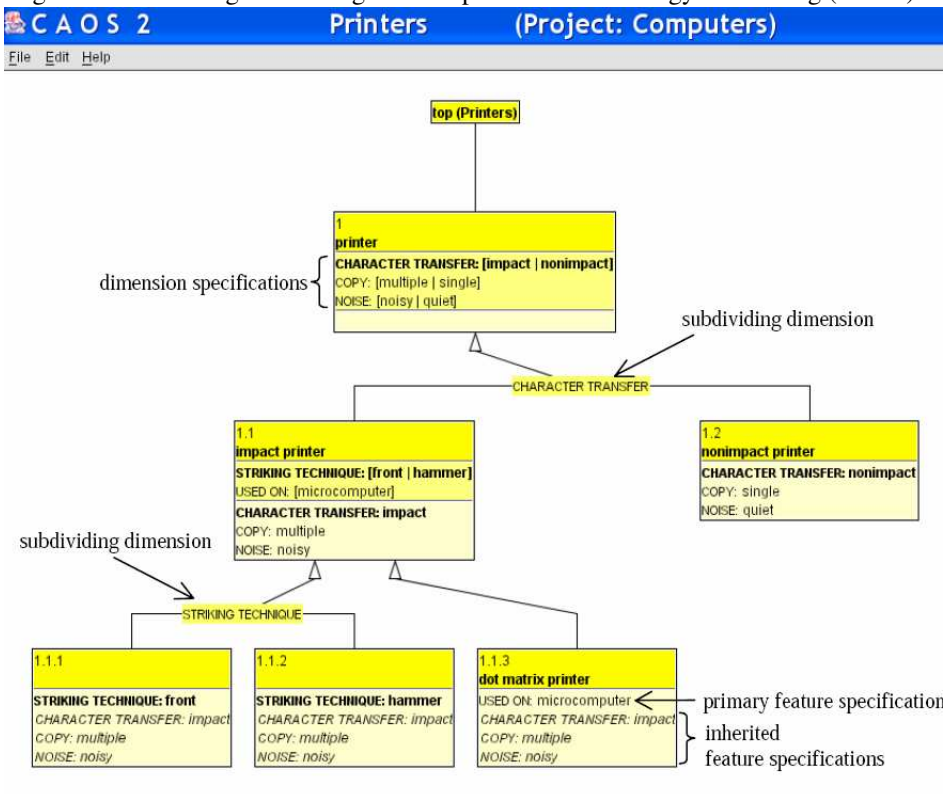
Returning to the classification of the UMLS Metathesaurus based on the typology represented in figure 6 and discussed in this subsection, the following can now be asserted.

- *The UMLS Metathesaurus is a pragmatic, informal, domain-specific, task independent, but language dependent special ontology*

It is informal because even if all concepts in the Metathesaurus are linked to a table of definitions[17], these definitions are phrased in statements of natural language rather

[14] for example "character transfer" in figure 7
[15] the concept "printer" in the example in figure 7
[16] for example [striking technique: front] in figure 7
[17] named "MRDEF"

Figure 7: Terminological ontologies: Computer Aided Ontology Structuring (CAOS)



than a formal language. It is special and domain-specific because it models only specialized concepts from Biomedicine, and it is language dependent because term variants and definitions are phrased in English. Even if the Metathesaurus does not meet the three principles outlined in [Madsen et al., 2004], its characteristics align very well with what is normally expected of terminological ontologies, in the more inclusive sense of the term.

2.3.3 Choice of relation types

The literature on the variety of semantic relation types used in terminology and lexicology is vast (see [Nuopponen, 2005] for a small sample). However, this section will only very briefly introduce some of the most well-known relation types used in terminology work and justify the choice of relations to be examined in the experimental part of the thesis.

Needless to say, concepts and conceptual relations are the bread and butter of ontologies. While conceptual relations are typically visualized as links, they can, as illustrated in subsection 2.3.2, also be represented as a feature-value pair in a feature specification, and thus the tasks of identifying conceptual characteristics and conceptual relations are in a sense quite similar.

> "Giving information about a concept relation and a related concept corresponds to the information on a characteristic of a concept" [Madsen et al., 2001, p7].

The number of ways in which concepts can be related to each other approaches infinity, but paralleling the distinction between logical and ontological concept systems, there are also two supertypes of conceptual relations, namely logical and ontological relations.

1. Semantic relations
   (a) synonymy
   (b) antonymy
   (c) ...
   (d) Conceptual relations
      i. logical relations
         A. ISA
      ii. ontological relations
         A. meronymy
         B. causality
         C. ...

As can be seen from the above diagram, the logical relation is also known as the ISA relation (or generic relation). It is fundamental in terminology, because it provides the genus proximum in classical definitions and because of the feature inheritance property

of logical concept systems. For example, if the concept "vehicle" has a particular feature, its hyponyms, for example the concept "bus", necessarily also has this feature. There are a wide range of ontological relation types, but the most universal one is meronymy, also known as the PART_OF or partitive relation (e.g. house - door).

An important difference between a relation like synonymy and the other relation types listed above is that synonymy does not relate two different concepts, but rather two different terms referring to the same concept. Although the links in terminological ontologies represent conceptual relations, synonymy still plays a vital role in terminology because identifying and grouping synonyms under a concept label is a key part of the conceptual clarification process.

Since ISA and synonymy are the two most fundamental relation types used in terminology (and lexicology for that matter) there is, of course, no avoiding these relations in the empirical experiments in this thesis. Additionally, the experiments will examine the two causal, ontological relations, "induces" and "may_prevent", which are both relatively common in the UMLS Metathesaurus (see table 3 for statistics) and very important in the domain of Biomedicine (this claim is empirically tested in subsection 4.1.2). Finally, the importance of causality is also suggested by a number of more recent terminological studies of this relation type, including [Marshman, 2002], [Barrière, 2001, Barrière, 2002] and [Girju and Moldovan, 2002].

Thus a total of four different relation types, three conceptual and one semantic, are empirically investigated in this thesis.

1. ISA (hyponymy and hypernymy)
2. "induces" relation
3. "may_prevent" relation
4. synonymy

Subcategories of causal relations like "induces" and "may_prevent" have been established from both an existence dependency perspective (see [Barrière, 2001, Barrière, 2002]) and from a role perspective (see [Madsen et al., 2001]), but it is beyond the scope of this thesis to discuss these subclassifications. After all, such an analysis is unlikely to have any impact on the methods which are employed or the results which are reported in chapters 4, 5 and 6.

2.3.4 UMLS knowledge sources

This subsection briefly introduces the UMLS knowledge sources and presents a few references to academic work relating to the usefulness of the UMLS knowledge sources in NLP and computational terminology.

The UMLS knowledge sources comprise

1. a Metathesaurus
2. a Semantic Network
3. a SPECIALIST lexicon

Table 3: Example semantic relations in the UMLS

relation type	#concept pairs
isa,inverse_isa	318,391
has_contraindicated_drug	79,335
tradename_of,has_tradename	75,767
part_of,has_part	42,637
may_prevent	18,805
induces,induced_by	3,854
may_diagnose,may_be_diag.	3,037

The Metathesaurus is a gigantic database containing information about 1.3 million biomedical and health related concepts and the relations between them. The Semantic Network is an upper-level ontology for this domain which ensures that concepts in the Metathesaurus are categorized in a consistent manner. It contains 134 semantic types and 54 types of semantic links and is described in [McCray, 2003]. Finally, the SPECIALIST lexicon contains some 300,000 biomedical terms and was compiled to facilitate the development of NLP software for the biomedical domain. Table 3 lists some examples of the 54 semantic relations defined in the UMLS Metathesaurus along with the number of registered concept pairs for each relation type (in the 2006AB edition of the UMLSKS).

As evidenced by the following quote the UMLS clearly is a monumental effort towards bringing together diverse biomedical terminologies and forming a coherent framework for knowledge structuring within this domain.

> The integration of standardized biomedical terminologies into a single, unified knowledge representation system has formed a key area of applied informatics research in recent years. The Unified Medical Language System (UMLS) is the most advanced and most prominent effort in this direction, bringing together within its Metathesaurus a large number of distinct source-terminologies. The UMLS Semantic Network which is designed to support the integration of these source-terminologies, has proved to be a highly successful combination of formal coherence and broad scope. [Smith et al., 2004]

Examples of research made possible by the UMLS knowledge sources include mapping between ontologies [Burgun and Bodenreider, 2001], extending terminological ontologies with hyponyms of existing concepts [Bodenreider et al., 2002b], evaluating the performance of information extraction systems [Klavans and Muresan, 2000], semantic annotation of medical abstracts [Vintar et al., 2002], evaluating context features for medical relation mining [Vintar et al., 2003] and many more.

Nevertheless, various papers (for example [Bodenreider, 2001], [Bodenreider et al., 2002a], [Kumar and Smith, 2003] and [Smith et al., 2004]) have identified a number of problems, for example coverage, redundancy or structural issues in both the UMLS Metathesaurus and in the UMLS Semantic Network (SN). That such problems exist in the

UMLS knowledge sources should come as no surprise given the diverse terminological sources from which the UMLS have been formed and given the complexity of the domain as observed in the following quote.

> One of the interesting aspects of the use of ontologies within bioinformatics is the complexity and difficulty of the modelling entailed. Compared to the modelling of man-made artefacts such as aeroplanes, some argue that natural systems are difficult to describe [19]. Biology is riddled with exceptions and it is often difficult to find the *necessary* conditions for class membership, let alone the *sufficiency* conditions. [...] There are several potential reasons for this, including:
>
> - Membership claims are in fact incorrect
> - Current biological knowledge is not rich enough to have found appropriate necessary and sufficiency conditions
> - In the natural world, the boundaries between classes may be blurred. Evolution is often gradual and the properties that distinguish one class from another may be only partially represented in some individuals. [Stevens et al., 2004, p640]

The three points raised in the above quotation reflect the conflicting viewpoints of classical terminology and modern theories like Sociocognitive terminology which argues that concepts in some domains should be viewed as prototype categories with graded membership conditions rather than clear-cut concepts (see subsection 2.2.4 for further details). That biomedical knowledge is complex and often incomplete, or even uncertain, is also reflected by at least one of the system tests, namely the one probing the beneficial effects of selenium in subsection 5.5.2.

Discussing the structural infelicities and possible redundancies of the UMLS is beyond the scope of this thesis. WWW2REL primarily makes use of the UMLS to obtain training term pairs, and the correctness of these pairs should not be affected by ontological redundencies or circularities. However, it is interesting to observe that the UMLS has been criticized for a low coverage in certain areas. For example, [Bodenreider et al., 2002a] have established that the coverage of concepts in the UMLS Metathesaurus range from 2% for gene product symbols to 44% for molecular functions. Applications like WWW2REL should thus be able to find unrecorded relation instances and augment even as comprehensive an ontology as the UMLS.

2.4 Conclusion

This chapter introduced the field of pattern-based AKA, discussed the basic properties of knowledge and specialized knowledge and of knowledge representation. Given the objective of the thesis, namely the automatic extraction of semantic relation instances for augmenting existing terminological ontologies, the exposition focused on the properties of specialized, descriptive knowledge. The properties of termhood were discussed (subsection 2.2.3), and this led on to a brief summary of the ongoing debate about the status and objective of terminology as a scientific field. The viewpoints

of various terminological schools were juxtaposed (subsection 2.2.4), and it was concluded that WWW2REL fits into neither of these frameworks because they are either prescriptive (GTT), purely conceptual (SCTT) or focused on units of discourse beyond the level usually processed by text mining systems (CTT).

The chapter also discussed basic techniques and principles of knowledge representation as well as a typology of ontologies (section 2.3). It was observed that finding semantic relation instances is essentially a way of identifying the distinctive features of concepts and thus an essential task in practical terminology work (subsection 2.3.1). Even though the UMLS Metathesaurus is not a terminological ontology in the strict sense of the term (subsection 2.3.2), it is still a special and domain-specific ontology and is ideal for testing the WWW2REL system because it is freely available and very comprehensive.

Before turning to the implementation and evaluation of WWW2REL in chapters 4, 5 and 6, chapter 3 will now introduce and discuss the methodological framework of text mining systems and also survey a range of similar systems.

3 Methodology and applications

While chapter 2 discussed theoretical aspects of terminology and knowledge engineering, the contents of this chapter will be of a more practical nature and primarily introduce the applied sciences which constitute the methodological foundations for the implementation of WWW2REL.

Reflecting the revived interest in corpus linguistics and statistical Natural Language Processing (NLP) over the past two decades, the methodological stance taken in this thesis is predominantly empirical and inductive. However, many branches of empirical linguistics have emerged, and the system implementation and evaluation is based on metrics and techniques originating from a number of these applied disciplines. Time and space constraints do not allow an exhaustive treatment of each of these disciplines, but the following sections will provide cursory overviews of each discipline and discuss the specific metrics and the empirical data which are subsequently used in chapter 4.

Section 3.1 introduces the discipline of corpus linguistics and discusses the nascent field of Web as Corpus (subsection 3.1.1). In section 3.2 key metrics from the field of Information Retrieval are presented. These are important because the entire evaluation performed throughout chapters 4, 5 and 6 makes use of similar metrics. Section 3.3 discusses the field of text mining with special attention given to text mining for the biomedical domain. Finally, section 3.4 describes and compares existing *pattern-based* relation extraction systems in terms of their performance and in terms of their similarity to WWW2REL.

3.1 Corpus linguistics

Although text corpora need not be in digtial form, corpus linguistics is very much a product of the digital revolution with easy access to vast quantities of digitized text and fast computers to help detect regularities in this text. A landmark in corpus linguistics was the establishment of the Brown corpus, a one million word, balanced corpus of

American English in 1967 by Henry Kucera and Nelson Francis [Kucera and Francis, 1969]. The corpus consisted of 500 text samples of 2,000 words each and representing fifteen different genres. One million words is not much by today's standards, but being highly balanced and carefully worked out, the Brown corpus continued to be used even through the following decades when the Chomskian focus on linguistic competence rather than performance made corpus linguistics a less attractive endeavour [McEnery and Wilson, 1996, pp4,18].

Towards the end of the eighties corpus linguistics reemerged from oblivion as faster computers and more digitized text revealed the promises of inducing usage patterns semi-automatically. Especially the field of British lexicography led on, and projects like the COBUILD project, instigated by Professor John Sinclair at the University of Birmingham, remain one of the great feats of that time. The most important revelation provided by, for example, corpus-based lexicography was that many native speaker intuitions about word frequencies and word collocations, for instance, were actually off the mark when compared to the results from extensive analyses of large quantities of actual usage. These insights paved the way for statistical NLP and the Machine Learning approaches to linguistics which are in vogue today, at the expense of introspective linguistics.

Some twenty-odd years after the Brown corpus was compiled, the British National Corpus (BNC) set a new standard by featuring 100 times as many running words and being equally balanced. Today even the BNC is considered small, but it is still useful and, in fact, will be used in this thesis to find verbs characteristic of Biomedicine (see subsection 4.1.2) and to eliminate semantic relation instances whose arguments have a low termhood (see subsection 5.3.1).

As a paradigm for the study of language corpus linguistics can be applied wholesale or with moderation, so to speak. The two degrees of empiricity in language study are often distinguished as

1. corpus-based
2. corpus-driven

> In a corpus-driven approach the commitment of the linguist is to the integrity of the data as a whole, and descriptions aim to be comprehensive with respect to corpus evidence. The corpus, therefore, is seen as more than a repository of examples to back pre-existing theories or a probabilistic extension to an already well defined system. The theoretical statements are fully consistent with, and reflect directly, the evidence provided by the corpus. Indeed, many of the statements are of a kind that are not usually accessible by any other menas than the inspection of corpus evidence [...] recurrent patterns and frequency distributions are expected to form the basic evidence for linguistic categories; the absence of a pattern is considered potentially meaningful. [Tognini-Bonelli, 2001, p84]

While the adjective "corpus-based" can be affixed to virtually any language study which to some (unknown) extent has made use of corpora to provide examples or test pre-existing theories, the adjective "corpus-driven" clearly involves a greater degree of

commitment to the corpus or corpora used in the study. [Tognini-Bonelli, 2001] has the following comments on what really distinguishes the corpus-*based* linguist from the corpus-*driven* linguist, so to speak.

> [...] given that the data is non-negotiable, does the linguist choose to revise the theory and derive it more directly from corpus evidence, or does (s)he opt to insulate the data from the theory? [...] Given what has been said in the paragraphs above, the answer is that the corpus-based linguist will go for the second option, feeling that a certain amount of variation that has not been accounted for is not important enough to topple a well-established theoretical position. [Tognini-Bonelli, 2001, p67]

Insulating corpus evidence, for example by relegating it to a separate linguistic realm called "performance", has been a popular strategy even in corpus-based linguistics. Another, less extreme, strategy employed when the data does not completely fit the theory is standardization. As observed by [Tognini-Bonelli, 2001] tagging raw text is, strictly speaking, against the spirit of the corpus-driven approach in that the set of tags is the product of a linguistic theory which is thus given higher priority than the actual data which is in a sense forced to comply with this theory.

Given the above quotations and discussion, the approach taken in the empirical investigations of knowledge pattern usage in this thesis must be classified as corpus-driven. At no point are pre-existing theories about the form of the patterns allowed to guide the discovery and filtering process. The only point at which the work violates the tenets of the corpus-driven research framework is when KP contexts are tagged and chunked so as to facilitate the extraction of relation instances[18]. Doing so allows a simplification and generalization of linguistic variation which is essentially the first step of moving from expression to content. It is thus justifiable from the perspective of practical terminology work which, at least as expounded in the GTT school (see subsection 2.2.4), necessarily involves standardization as part of the process of knowledge representation.

Even if the present work can be characterized as truly corpus-driven, one important question remains. Can the WWW be regarded as a corpus? Subsection 3.1.1 will attempt to answer this question.

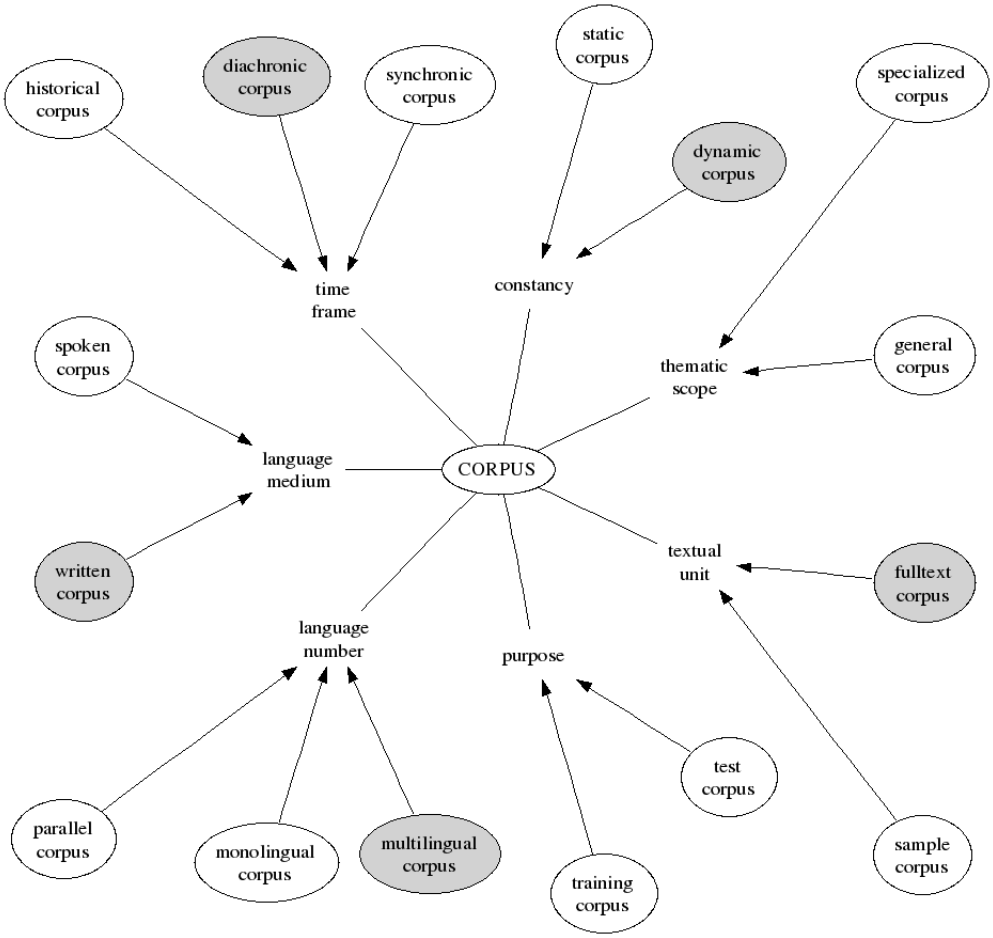
3.1.1 Web as corpus research

Before discussing the status of the WWW as a corpus, it will be worthwhile to revisit the corpus typologies and definitions of the "corpus" concept by some of the grand old men of corpus linguistics. The following definitions are taken from four text books in corpus linguistics by John Sinclair, Douglas Biber, Graeme Kennedy and Tony McEnery and Andrew Wilson, respectively.

1. A corpus is a collection of naturally-occurring language text, chosen to characterize a state or variety of a language. [Sinclair, 1991, p171]

[18]Forcing a verb for the "induces" and "may_prevent" relations (see subsection 4.1.5) does not violate the corpus-driven tenets, because the usefulness of doing so was established in another, truly corpus-driven study (cf. [Barrière, 2001]).

Figure 8: Corpus typology



2. [...] it utilizes *a large and principled collection of natural texts* [my emphasis], known as a 'corpus'. [Biber et al., 1998, p4]

3. Whereas a corpus designed for linguistic analysis is normally *a systematic, planned and structured compilation of text* [my emphasis], an archive is a text repository, often huge and opportunistically collected, and normally not structured. [Kennedy, 1998, p4]

4. So a corpus in modern linguistics, in contrast to being simply any body of text, might more accurately be described as *a finite-sized body of machine-readable text, sampled in order to be maximally representative of the language variety under consideration* [my emphasis]. [McEnery and Wilson, 1996, p24]

The key delimiting characteristic which sets a corpus aside from a text collection appears to be the way in which the texts have been compiled, namely in a "principled" versus an "opportunistic" manner. In other words, corpus status is granted to text collections which have been compiled so as to be as representative a sample as possible of the population they are meant to model.

Since the textual content freely accessible on the WWW has not been compiled in a principled manner, we may already by this point deny the WWW corpus status. However, the WWW, or rather subsets of the textual content accessible through the WWW, could be called representative in the sense that given their sheer volume they will, in their entirety, be more representative than any manually constructed corpus could ever hope to be.

Although textual content on the WWW, strictly speaking, does not qualify as a corpus because no planning went into its design, it does share the following properties with genuine corpora (marked by grey in figure 8). It is a

- dynamic
- multilingual
- diachronic
- written
- fulltext

collection of text. Probably the key feature which sets the WWW aside from most other text collections (and corpora) is its dynamic character. It is the dynamic nature of the WWW which makes it impossible to document how much text and what kind of text is actually accessible in the entire collection. URLs constantly appear, disappear and reappear, and although projects like the Internet Archive[19] attempt to take snapshots of parts of the web at regular intervals, these snapshots are only small samples of all accessible text on the WWW. In this sense the WWW can perhaps neither be characterized as synchronic or diachronic (although marked as such in figure 8). It is not truly synchronic because lots of ageing web documents remain accessible, and it is not completely diachronic because most web pages either get updated regularly or deleted. In fact, the WWW can be interpreted as the species of corpora called "monitor corpora" by John Sinclair in 1991 [Sinclair, 1991, pp24-25], although he had probably not envisaged as anarchistic a text source as the present-day WWW.

As for the dimension of "thematic scope" in figure 8 the WWW in its *entirety* represents a collection of text which could be called "general", but through focused querying one can indirectly access the infinite range of specialized text clusters making up this whole. When querying the WWW for very specific terms, like the antipsychotic drug "haloperidol" (see section 5.5), for example, it is presumably mainly the specialized text clusters which return hit results as laymen are unlikely to discuss the properties of haloperidol in their web logs. Also, the case of using the technical synonym for

[19] www.archive.org

"vomiting", namely "emesis", appears to focus the query towards the specialized text clusters (see subsections 5.5.3 and 5.5.4).

In short, querying the WWW has become the main source of knowledge and information for millions of users around the world. Even if the textual content on the WWW has not been placed there in a principled manner, it has become used as a *de facto* corpus in NLP tasks in recent years. The single most important benefit of using the web as a corpus is that it solves the long-standing problem of data sparseness. The use of the WWW in computational linguistics was ushered in by the introduction to the 2003 special issue of the journal *Computational Linguistics* [Kilgarriff and Grefenstette, 2003], and recently Web as Corpus (WAC) workshops have appeared at established conferences like the European Association for Computational Linguistics[20].

In short, the paradigm seems to be shifting from optimizing recall on minute data collections towards optimizing precision on vast collections on data.

> Language is so expressive that it is practically impossible for the patterns learned from a relatively small training set to cover all the different ways of describing events. Consequently, the IE patterns learned from manually annotated training sets typically represent only a subset of the IE patterns that could be useful for the task. [Patwardhan and Riloff, 2006]

In the task of augmenting terminological ontologies with new semantic relation instances, it is a great help to rely on the WWW for discovering KPs and subsequently for finding new instances with these KPs. First of all, as the WWW covers any conceivable domain of interest it should be possible to achieve system portability. Secondly, since the WWW features the largest text collection on the planet, using it should minimize data sparseness problems. Especially for highly specialized, perhaps even domain-specific, relation types, learning and applying knowledge patterns without using the WWW will often involve a costly and time-consuming manual compilation of a specialized corpus, and this endeavour might even still be thwarted by data sparseness. Not only is the WWW a vast data source, it also contains even the newest bits of knowledge and is freely available. As is discussed in subsection 5.2.2, however, it may also contain incorrect or old knowledge. The first is clearly undesirable, but the latter may be relevant in a terminological context. Finally, the WWW is a potential source of knowledge on any conceivable domain and in a wide range of languages. In fact, much recent research in automatic relation extraction illustrates that the web is already being used as a corpus to solve this task (see table 5 in section 3.4).

Of course, it is not unproblematic to make use of an unprincipled text collection as if it were a corpus, or indeed a *specialized* corpus. The main disadvantage of searching for knowledge on the entire WWW is that it is a noisy source of knowledge. This noise, of course, affects negatively the performance of the system implemented and evaluated in this thesis but, as argued above, the advantages of relying only on the WWW outweigh the disadvantages of doing so. All this to say that the performance of relation extraction systems operating on tidy collections of research papers will be better (in terms of precision) than systems operating exclusively on the WWW. For example, automatic term recognition is easier when based on domain specific text than on text

[20] http://eacl06.itc.it/workshops/workshop.htm

covering all conceivable domains, because there is less polysemy in domain-specific text than on the WWW as such. For example querying the WWW for the keyword "virus" will yield both biological viruses, computer viruses and a range of metaphorical viruses, but in a corpus of biomedical research papers virtually all occurrences of "virus" will refer to the biological concept.

A few additional challenges affecting WAC research are described in subsection 3.1.2.

3.1.2 Problems with Web as Corpus

This subsection introduces and discusses four factors which may constitute problems for language studies based (or driven) by the Web as Corpus, or WAC approach. The list is by no means exhaustive, but nevertheless treats some of the more conspicuous and typical problem areas.

Structure Possibly the main challenge facing applications which treat the web as a corpus in language study of any kind is its lack of structure. The text accessible through WWW represents virtually any conceivable text type, genre and language. Practically all stylistic levels of language use are represented, and it is often impossible to verify information about the author of a particular text. Indeed the success of projects like the web-based encyclopedia, Wikipedia, are largely attributable to the practice of having multiple authors write and edit the same text.

While the multilingual nature of the WWW was listed as an advantage, it may also be a curse in disguise. Even if information about the language and national origin of individual web pages is indexed by most web search engines, this information is not completely reliable as non-nationals may acquire national domain names, for example. The retrieval of textual content in non-target languages may thus be caused by homographs, loan words and so on.

Dynamicity Secondly, it can also be very difficult to assess the exact date when a particular text was produced. Although most web search engines allow the user to define certain date ranges as part of their queries, the dates are often unreliable because they refer not to a document's date of genesis but to the date at which the latest change in the document was observed and indexed by a web crawler. However, as the so-called RSS feeds[21] gain popularity on the WWW, web content is becoming richly annotated with more reliable meta-data such as the date of publishing, for example. Unfortunately, RSS feeds are mostly used for news services and web logs of a predominantly general language rather than terminological character. Extracting bits of knowledge from the WWW at large thus means that relations constituting both old and state-of-the-art knowledge are likely to be retrieved. However, both old and state-of-the-art knowledge can, in fact, be useful to terminologists building concept systems (more on this in subsection 5.2.2).

[21] Really Simple Syndication

Document formats Thirdly, web pages are typically adorned with a variety of metalanguage mark-up like HTML, XML, javascript and so on. Although such mark-up can fairly easily be stripped from the document, and, in fact, is stripped automatically when using search engine APIs like Google's or Yahoo's to download text snippets, other sources of noise remain. One such source of noise is that HTML entities should be translated back into regular letters (see appendix 8.8).

Content duplication Fourthly, the WWW is rife with duplication. Retrieving multiple identical text snippets may constitute a bias both when discovering KPs (see section 4.1), but also when testing the system and ranking instances by their reliability (see chapter 5). For these reasons the Google API "filter" flag has been employed whenever possible. This filter is supposed to eliminate near-duplicate content and also multiple results coming from the same Web host as it says in the API documentation[22]. In spite of these web search engine API settings, some measure of duplicate content could not be avoided in the collections of text snippets used in WWW2REL. However, there are two reasons why this is unlikely to have biased the results to any significant extent. Firstly, a filtering technique called "iteration range" is used when filtering KPs in section 4.2, and this looks beyond simple KP frequency and requires all candidate patterns to occur with a wide range of *different* term pairs to be considered valid. Secondly, an instance reliability measure referred to as "KP range" is used when ranking relation instances in chapter 5, and this also looks beyond the instance frequency itself and focuses on the range of *different* KPs with which a candidate instance co-occurs.

Distribution and copyright An additional challenge with the WAC approach to language study is "how to make the corpus available to other researchers" [Sharoff, 2006, p453]. This is mainly a problem when compiling full text corpora from the WWW, which cannot be distributed without the consent of the person or organization holding the copyrights. [Sharoff, 2006] discusses how storing and distributing only a long list of URLs pointing to the (freely available) textual content would allow other researchers to regenerate the web corpus on their own computer without violating copyright regulations. However, many URLs will be so-called "deep links" pointing to interior parts of the target websites, and doing so may be deemed illegal (cf. the Danish "Newsbooster" trial). Even if legality is not an issue, collections of URLs age rapidly due to the dynamicity of the WWW, so this distribution strategy may also prove suboptimal.

As for the textual data on which the experiments presented in this thesis are carried out, the author will take the liberty of providing the reader with a URL pointing to the complete text archives[23]. This should not violate any copyrights as the web corpora are not based on full texts, but on minute text snippets each containing at most one or two sentence fragments.

[22] see http://code.google.com/apis/soapsearch/reference.html

[23] they can be found at http://www.halskov.net/phd

3.2 Information Retrieval and Information Extraction

Information Retrieval (IR) is the science of searching for documents which are relevant in a particular context. WWW2REL searches not for individual documents, but for fragments of knowledge occurring in a wide range of documents (across the entire WWW). In this sense what the system performs is closer to Information Extraction (IE) than IR. IE is about finding events, but also "specific facts about prespecified types of entities and relationships of interest" [Spasic et al., 2005]. As mentioned in the thesis introduction, however, IE differs from AKA in that it primarily targets conceptual *extension* rather than *intension*, and it is the latter which is the research object of terminologists and thus the focal point of WWW2REL.

IR (and the wider field of Machine Learning) features methods for measuring the performance of the document retrieval process which can also be used to measure the performance of relation extraction systems like WWW2REL, and for this reason a cursory description of the IR evaluation metrics is given in this section.

The two basic performance measures are "precision" and "recall". They are defined as

$$precision = \frac{|\{documents_{relevant}\} \cap \{documents_{retrieved}\}|}{|\{documents_{retrieved}\}|}$$

$$recall = \frac{|\{documents_{relevant}\} \cap \{documents_{retrieved}\}|}{|\{documents_{relevant}\}|}$$

In other words, the precision of an IR system is the proportion of relevant documents in the set of retrieved documents, and its recall is given by the proportion of relevant documents retrieved out of the total number of relevant documents which could possibly be retrieved. Needless to say, it is easy to optimize either quality parameter on its own, but to be really useful IR systems need to find the right balance between high precision and high recall. This balance is given by the so-called F-measure.

$$F = 2 * \frac{precision * recall}{(precision + recall)}$$

In technical terms, F is the weighted harmonic mean of precision and recall.

The formulae given here will be applied multiple times in this thesis, for example in chapter 5 when measuring the performance of various relevance ranking algorithms versus a gold standard provided by four domain experts. The only formal difference when applying these IR evaluation metrics to the task of relation instance extraction is that "documents" should be exchanged for "semantic relation instances".

3.3 Text, data and web mining

The terms "text mining", and its synonym "text data mining", are often used to describe applications which operate on natural language text to solve a number of specific tasks. This section will discuss what exactly differentiates the field of text mining from corpus linguistics (or more generally computational linguistics), data mining, web mining, Information Retrieval (IR) and other related, empirically founded methodologies.

Table 4: Data mining, text mining, IR and computational linguistics

	textual data	non-textual data
Finding patterns	computational linguistics	stdandard data mining
Finding novel nuggets	real TDM	?
Finding non-novel nuggets	IR	database queries

> Text mining is differentiated from both Information Retrieval (IR) and text summarisation (TS) in that while IR and TS focus on the largest units of text such as documents, text mining operates at a *finer level of granularity* [my emphasis] and examines relationships between specific kinds of information contained both within and between documents. Text mining is also differentiated from full-blown natural language processing (NLP) in that NLP attempts to understand the meaning of the text as a whole, while text mining and knowledge extraction concentrate on *solving a specific problem* [my emphasis] in a specific domain identified *a priori* (possibly using some NLP techniques in the process). [Cohen and Hersh, 2005, p58]

While the difference between text mining and IR is a matter of the scale of analysis, the main difference between text mining and data mining is that in text mining pieces of information are extracted from *unstructured text*, whereas in data mining information is extracted from *structured data*. For this reason data mining is often referred to as Knowledge Discovery in Databases[24]. Also, the data which is analyzed in data mining is often non-textual. As for the distinction between Information Retrieval, computational linguistics and text/data mining table 4 which is taken from [Hearst, 1999] is instructive because it clarifies the concepts using just three delimiting characteristics, namely the target and purpose of the analysis as well as the novelty of what is found. The acronym "TDM" in the table stands for text data mining.

In the opinion of [Hearst, 1999] real text (data) mining, or TDM, should uncover previously unknown information from text and as such it is comparable to the field of "hypothesis generation". Computational linguistics typically "just" automatizes and improves language analysis and synthesis for a variety of purposes, for example transforming text from one language into another, but does not discover new "nuggets" of knowledge to use the terminology of [Hearst, 1999]. Nor does IR, but the difference between IR and computational linguistics is that IR processes (and returns) textual units in a qualitatively different manner than applications in computational linguistics. IR processes entire documents at a time to determine whether they contain relevant nuggets of knowledge for a specific user. Real TDM, on the other hand, tries to uncover entirely new nuggets from collections of documents which are often unrelated.

Responding to the text mining typology in table 4 [Kroeze et al., 2003] raise the following criticism.

[24] the ACM Special Interest group SIGKDD organizes annual conferences in Knowledge Discovery and Data mining

> [...] the 'nuggets' and searched-for 'needles' already exist and they are already known by someone, and the problem is to locate them. Finding them cannot be regarded as novel information, in other words there is no such thing as novel information *nuggets*; it is a contradiction in terms. [...] Therefore, these two columns should be merged and the process can be called *non-novel investigation*. [Kroeze et al., 2003, p96]

The authors procede to establish a new typology of data and text mining which basically distinguishes between

1. non-novel investigation = information retrieval
2. semi-novel investigation = knowledge discovery
3. novel investigation = knowledge creation

[Kroeze et al., 2003] are right that standard text mining, including what [Hearst, 1999] calls "real TDM", procures no new knowledge, in that the nuggets or patterns already exist in the text. However, there is a degree of novelty in finding these nuggets in various text sources, grouping them and displaying them in a structured manner to a user who may not have been aware of their existence. This is exactly what WWW2REL attempts in the following chapters.

Finding, or rather creating, truly new knowledge (called "intelligent text mining" by [Kroeze et al., 2003]) has traditionally been the domain of human beings, and to accomplish this task computer systems will need to make use of advanced techniques from Artificial Intelligence.

Finally, table 4 does not mention the field of "web mining", but this is simply a slightly more inclusive term which embraces both text mining and data mining, since patterns can be found in all sorts of data on the WWW only some of which is text.

In conclusion, what is attempted by WWW2REL, then, is a semi-novel investigation of text on the WWW to discover nuggets of specialized knowledge which are new only in the sense that they may not be recorded in the target ontology.

3.3.1 Text mining for the biomedical domain

Although Biomedicine is simply a convenient case study for the test and evaluation of WWW2REL, examining recent developments in the popular field of biomedical text mining will provide a reference point for comparisons. However, it must be stressed that the system presented in this thesis is not custom-tailored for the biomedical domain (due to portability considerations) as are many of the applications to be discussed in following.

[Cohen and Hersh, 2005] provide an excellent survey of recent research in biomedical text mining and identify the following prominent areas.

1. Named entity recognition (NER)
2. Text classification

3. Extracting synonyms and abbreviations
4. Relationship extraction
5. Hypothesis generation

Since relationship extraction is the topic of the thesis, the other branches of biomedical text mining will be ignored. Among the pattern-based approaches to biomedical relationship extraction, [Cohen and Hersh, 2005] distinguish between those which use patterns manually generated by domain experts (e.g. [Yu et al., 2002]), those which induce patterns automatically from term pair contexts (e.g. [Yu and Agichtein, 2003]), those relying mainly on collocational statistics and those making heavy use of NLP methods like parsing. In this classification, WWW2REL is a bit of a mix in that it induces patterns automatically, but also features basic NLP methods (tagging and chunking) and simple statistics (frequency statistics from a general language reference corpus).

It is evident from the survey in [Cohen and Hersh, 2005] that practically all applications have been trained and tested only on collections of medical papers or on MEDLINE which is the most comprehensive bibliographic database of biomedical journal citations and abstracts in the world[25]. All records in MEDLINE are indexed with the National Library of Medicine's controlled vocabulary known as MeSH (Medical Subject Headings) which contains some 23,000 descriptors related by parent/child relations and cross references. "The MeSH thesaurus is used by the NLM for indexing articles from 4,600 biomedical journals for the MEDLINE/PubMed database." [Ananiadou and McNaught, 2006, p54]. MeSH, in fact, is one of the source vocabularies providing concepts and relations for the UMLS Metathesaurus described in subsection 2.3.4).

While the methodology for extending terminological ontologies using the WWW (chapter 4) could be applied to any conceivable domain, one might ask if the approach is useful in the context of Biomedicine which already has these vast and widely used terminological resources.

> While manual curation and indexing can be an aid to researchers searching for approprate literature, a recent study of the information content of MEDLINE records by Kostoff et al. found a significant amount of conceptual information present only in the abstract field and missing from the MeSH terms. This is not surprising since the MEDLINE indexers and the MeSH vocabulary, while broadly based, cannot be expected to represent all of the concepts of interest for all potential users. Clearly, the full text of biomedical literature contains a wealth of information important to users that may not be completely captured by reviewers and curators. [Cohen and Hersh, 2005, p58]

The above quote suggests that going beyond MEDLINE and mining biomedical literature in a wider sense is indeed a useful endeavour. Automatically extracting semantic relation instances from free text on the WWW only takes this idea a bit further,

[25] see www.nlm.nih.gov/pubs/factsheets/medline.html

Table 5: Performance of pattern-based relation extraction systems

relation types	performance	publication
multiple, also non-ISA	P: 49%-85%	[Pantel and Pennacchiotti, 2006]
location;organization	P: 88%-96%	[Agichtein and Gravano, 2000]
ISA, "related classes"	P: 84%-100%	[Popescu et al., 2004, Etzioni et al., 2004]
any	P: 51%-56%	[Turney, 2006]
country;capital and 9 more	P: 14%-96%	[Alfonseca et al., 2006a, Alfonseca et al., 2006b]
UMLS: causes, diagnoses ..	P: 75%-100%	[Mukherjea and Sahay, 2006]
gene-protein	P: 65%	[Gaizauskas et al., 2003]
gene-protein synonyms	P: 71%-90%	[Yu et al., 2002]
"semantically related"	F: 37%-68%	[Nenadic and Ananiadou, 2006]
PART_OF	P: 55%	[Charniak and Berland, 1999]
causal	P: 66%	[Girju and Moldovan, 2002]

namely away from mining neat collections of meticulously compiled but rapidly ageing biomedical papers to mining the largest and most dynamic collection of text on the planet. In fact, recent research in biomedical text mining already seems to be heading in this direction, see for example [Mukherjea and Sahay, 2006] to be discussed in section 3.4.

3.4 Pattern-based relation extraction systems

This section provides a non-exhaustive survey of existing applications using the pattern-based approach to automatically acquire knowledge in the form of semantic relation instances from text. Table 5 lists eleven such systems along with the relation types they extract and the performance reported in the respective publications. The performance scores (P for precision and F for f-score) pertain to the correctness of the relation instances returned by the systems. As indicated in the table some systems focus on relation types which are specific to the domain of Biomedicine whereas other systems handle more domain-independent relations (e.g. ISA, PART_OF, causality etc.). As observed in the following quote one cannot directly compare the performance of systems operating on different data and solving different tasks.

> Unfortunately, at this time precise comparative evaluation of existing IE systems developed for the biomedical domain cannot be made, since the tasks and text collections addressed by researchers vary widely. [Gaizauskas et al., 2003]

Although this is true, it is still worthwhile to examine a wide range of relation extraction systems because this will elucidate how they differ and also pinpoint possible unique features of the WWW2REL system (see table 7). When comparing the reported performance scores in table 5 with eachother and with the precision scores reported in this thesis (chapter 5), it should be stressed that extracting relation instances from non-noisy domain-specific text in which terms have already been manually annotated with MeSH descriptors is an easier task than finding relation instances in unannotated and

uncategorized WWW text snippets. On the other hand, the relation types extracted by many systems custom-tailored for Biomedicine are often somewhat more specific[26] than those extracted by systems operating on general language text collections. However, this does not necessarily make the automatic extraction a harder task, because it allows the developers to make use of highly domain-specific techniques as described in subsections 3.4.1 and 3.4.6, for example.

The two systems presented in [Turney, 2006] and [Nenadic and Ananiadou, 2006] differ from the others in that the relation type to be extracted is not fixed in advance. For this reason these systems will be left out of the survey. Also, it should be noted that the precision scores listed for the system developed by [Turney, 2006] pertain to two rather complicated tasks, namely solving word analogy questions and classifying implicit semantic relations in noun-modifier pairs. In both cases explicit patterns are induced empirically from free text, but they are not used to extract *new* relation instances, but rather to classify relations between *existing* instances. In this way [Turney, 2006] performs relationship recognition rather than relation instance extraction, or "role extraction" as it is called in subsection 2.1.1.

Out of the eleven pattern-based relation extraction systems listed in table 5 the two systems developed by [Mukherjea and Sahay, 2006] and [Pantel and Pennacchiotti, 2006] are most similar to WWW2REL. The following subsections will briefly discuss how the systems work and how they differ from eachother and from WWW2REL. The survey begins with a brief review of two systems which are less similar to WWW2REL, but which illustrate the point raised in the above about custom-tailoring to a specific domain.

3.4.1 SGPE

The SGPE system developed by [Yu et al., 2002] extracts synonymous gene and protein names from MEDLINE abstracts and fulltext medical journal articles. In contrast to most of the extraction systems discussed in this section the synonymy KPs used in SGPE are identified manually by the system developers and then used to retrieve sets of candidate synonyms from the abstracts or articles. The authors define two types of biomedical synonyms, namely synonymy between short and long forms and synonymy between single word short forms. They argue that the first type of synonymy is easier to detect automatically than the second, because it essentially is a mapping between full forms and their acronyms. The work presented in [Yu et al., 2002] thus only considers the second type of synonymy.

SGPE is an interesting case because it uses a range of filters to eliminate terms which are not genes or proteins from the sets of candidate synonyms. Two of the filters are similar in nature to the BNC-based termhood filter implemented in this thesis and described in subsection 5.3, but the other filters are highly domain-specific and would reduce system portability if reproduced. The domain-independent filters are a dictionary of units (sec, min etc.) and an unnamed dictionary of common English words. Candidate synonym sets are deleted if two-thirds or more of the terms are common English words. The domain-specific filters feature the following rules as outlined in

[26] e.g. the roles of amino acid residues in protein molecules [Gaizauskas et al., 2003]

[Yu et al., 2002].

- delete set if any single term has more than six letters
- delete set if any term contains two or more dashes
- delete candidate if it is listed two or more times in the same abstract or article

The two first rules are clearly specific to Biomedicine, or rather to gene and protein names. Including such rules would lower system portability and be against the objectives outlined in the thesis introduction. The last rule is based on the assumption that "most of the authors introduce a synonym only once in their abstracts" [Yu et al., 2002], and will not be useful in the context of WWW2REL which operates on text snippets which are typically only one or two sentences long.

3.4.2 Snowball

Although it is an IE system focusing on conceptual extension rather than intension, the "Snowball" developed by [Agichtein and Gravano, 2000] makes use of techniques relevant also in the task of terminological relation extraction. Snowball is based on the DIPRE technique described in subsection 2.1.2, but the strength of the system is that in each iteration of the algorithm it assigns a confidence score to each of the induced patterns to better control the expansion phase. The confidence of a pattern, P, is defined as

$$Conf(P) = \frac{P.positive}{P.positive + P.negative}$$

where P.positive is the number of positive matches for P, and P.negative is the number of negative matches.

An <organization;location> pair is considered negative if there is, by the current iteration, a high confidence pair with the same organization but a different location. Analogously, an <organization;location> pair is considered positive if the exact same pair has previously been detected with high confidence. At each iteration Showball then recomputes the confidence of the extracted pairs and keeps only those for which it is most confident.

There are two reasons why the Snowball methodology is not well suited for augmenting terminological ontologies in a framework like WWW2REL which is not meant to be custom-tailored for a particular domain. As mentioned in subsection 2.1.3, Snowball relies on a named-entity tagger when detecting new <organization;location> pairs in documents, and for many domains NE taggers simply do not exist because they typically require much manual data to be trained. Secondly, while it is the case that most organizations have a single headquarters, the relations examined in this thesis behave quite differently in terms of cardinality. For example, drugs typically have multiple side effects and exhaustive lists of these are not easy to come by.

Table 6: System test (from [Mukherjea and Sahay, 2006])

relation	semantic type (example input term)	Precision
causes	Disease (Typhoid)	0.82
diagnoses	Anatomical Abnormality (Cyst)	1.00
consists of	Organic Chemical (Butane)	0.75
affects	Gene (Statin)	0.80
binds	Amino Acid, Peptide or Protein (Rhodopsin)	0.83

3.4.3 RelationAnnotator

The system developed by [Mukherjea and Sahay, 2006] resembles WWW2REL in that its knowledge source is also WWW text snippets, and it is also tested on the domain of Biomedicine by using UMLS terms. In fact, it appears to be one of the only relation extraction systems for Biomedicine operating directly on the WWW.

The authors first employ their system to classify a range of biomedical terms. They do this by looking for ISA relations between a set of 100 UMLS terms and a set of 10 common semantic types or classes in the UMLS, for example "vitamin", "protein" and so on. In the absence of domain experts they scramble the terms and classes to obtain an equal number of positive and negative examples. Using a set of handcrafted KPs they then retrieve the total Google hit count for each term-KP-class triplet when inserting all patterns in the KP position. Using a minimum threshold value of 25 they report a precision score of 87.5% in this classification task.

Secondly, the authors have implemented a RelationAnnotator which they test for the five relation types listed in table 6 combined with a range of UMLS terms representing five semantic types as input. Supporting the RelationAnnotator is the BioAnnotator, a named entity recognizer which conflates term variants in the text snippets which are known to represent the same UMLS concept. The relation instances are ranked by their total snippet co-occurrence with all KPs used for the target relation type, and average precision scores (evaluated by the authors themselves) are also given in table 6. Precision is computed as "the number of entities for which at least one relation that was identified by our system is correct [divided by] the number of entities for which at least one relation was identified by our system" [Mukherjea and Sahay, 2006].

Although there are many similarities between the Relation/Bio Annotator and WWW2REL, the two approaches differ in the following ways.

1. WWW2REL induces all KPs automatically using term pairs from the starting ontology (no handcrafting)
2. It makes no use of biomedical Named Entity Recognition (NER).
3. It features a range of different instance reliability measures (section 5.3)
4. It is evaluated by four domain experts rather than the developer himself
5. Its portability is tested by applying it to another domain (section 6.5)

6. The degree of "new" knowledge extracted by WWW2REL is assessed (section 6.4)

Although biomedical NER would presumably increase precision, this technique would only apply to the domain of Biomedicine and thus reduce the overall portability of WWW2REL. Also, NER may not be helpful in the task of identifying *new* relation instances not recorded in the available terminological resources.

3.4.4 Espresso

[Pantel and Pennacchiotti, 2006] report on a relation extraction system named "Espresso" which is in some ways more similar to WWW2REL than the RelationAnnotator. Unlike the system developed by [Mukherjea and Sahay, 2006], Espresso is meant to be a general purpose relation extraction system and its KP arsenal is induced from textual corpora. KPs are induced by extracting all substrings containing a seed term pair and filtered by assessing their reliability. The reliability of each KP, r(p), is computed as "its average strength of association across each input instance i in I', weighted by the reliability of each instance i" [Pantel and Pennacchiotti, 2006].

$$r(p) = \frac{\sum_{i \in I'} \frac{pmi(i,p)}{max_{pmi}} * r(i)}{|I'|}$$

where max_{pmi} is the maximum pmi of all instances with all patterns, $|I'|$ is the number of different relation instances co-occurring with the pattern, p, and pmi(i,p) is the pointwise mutual information of a particular instance and a particular pattern. Pmi(i,p) is given by the probabilities of the two events.

$$pmi(i,p) = log \frac{P(i,p)}{P(i) * P(p)}$$

In other words the ratio between the number of times i and p actually co-occur (the numerator) and the number of times they could be *expected* to occur (the denominator). The pointwise mutual information of a relation instance and a pattern can be estimated using Google hit counts as follows.

$$pmi(i,p) \approx \frac{C_{google}(t_1, p, t_2)}{C_{google}(t_1, *, t_2) * C_{google}(*, p, *)}$$

where t_1 and t_2 are the two terms forming the relation instance, for example "aspirin * bleeding", and "*" is a word wild card matching one or more complete words.

Espresso not only filters the induced KPs by their reliability, it also filters the extracted relation instances by their assessed reliability.

> Estimating the reliability of an instance is similar to estimating the reliability of a pattern. Intuitively, a reliable instance is one that is highly associated with as many reliable patterns as possible (i.e., we have more confidence in an instance when multiple reliable patterns instantiate it.) [Pantel and Pennacchiotti, 2006]

The formula for computing the reliability of an instance i, r(i), is thus completely analogous with that for computing the reliability of a pattern, p.

$$r(i) = \frac{\sum_{p \in P} \frac{pmi(i,p)}{max_{pmi}} * r(p)}{|P|}$$

Finally, Espresso features a unique technique for utilizing "generic" KPs, i.e. patterns with a high recall but low precision, to boost system recall without lowering system precision. The noise generated by generic KPs is simply filtered out by means of WWW hit counts and the assumption that instances extracted by generic KPs will also be extracted by *at least one* of the system's reliable KPs.

Again, WWW2REL is similar to Espresso in many ways, but differs in that

1. With WWW2REL KPs are induced from a noisy source (text snippets on the WWW)
2. WWW2REL is tested on the same noisy knowledge source not an offline corpus like (TREC-9)
3. It is not based on an iterative algorithm
4. It uses a binary rather than a continuous measure of KP reliability, r(p)
   (a) r(p) is set to 1 for all KPs accepted by a combination filter (see table 33 in subsection 4.2.5)
   (b) r(p) is set to 0 for all other KPs
5. Its instance ranking schemes operate on a web corpus sample and no further WWW querying is necessary
6. It does not try to harness the recall power of generic KPs

The main reason WWW2REL is not based on an iterative algorithm (like Snowball and Espresso) is that when using the WWW to induce KPs and extract relation instances data sparseness is less of an issue. If more term pair contexts are needed, the number of text snippets returned by the web search engine can simply be increased instead of going through one or more DIPRE expansions. However, if the number of seed term pairs is really low, or the pairs co-occur extremely rarely on the WWW, the WWW2REL KP discovery module could easily be made iterative.

Setting r(p) to 1 for all filtered KPs is motivated by the fact that effective KP filtering techniques (see section 4.2) eliminate low precision patterns, and also, implementing a BNC-based termhood filter (subsection 5.3.1) is expected to further increase precision. As for the final difference, when searching on the entire WWW, boosting recall is rarely necessary, but boosting precision is all-important.

3.4.5 KnowItAll

The KnowItAll system is a purely web-based IE system originally described in [Etzioni et al., 2004] and extended in [Popescu et al., 2004]. It consists of "an extensible ontology and a small number of generic rule templates from which it creates text extraction rules for each class and relation in its ontology" [Etzioni et al., 2004]. The generic rule templates are similar to the ones described in [Hearst, 1992], for example "NP1 <such as> NPList2" for the ISA relation. Filling out one of the argument slots (for example "cities" for NP1) the templates are converted into "discriminator phrases" and fed to an Extractor module which retrieves candidate relation instances from web documents using a range of search engines and a PoS tagger. The candidates are then processed by an Assessor module which determines the probability of each instance. The probabilities are computed by treating the hit counts of the discriminator phrases as conditionally independent features of the relation in question. These are combined by means of a naive Bayesian classifier, and KnowItAll estimates probabilities by bootstrapping positive and negative instances from the web.

In contrast to WWW2REL the objective of KnowItAll is to extract hundreds, or even thousands, of instances of specific input classes. In other words it focusses on conceptual extension rather than intension and on building taxonomies rather than compiling also non-hierarchical relation instances which may provide terminologists with the distinctive features they need for their definitions. Although the KnowItAll has been extended with a module for finding *related* classes [Popescu et al., 2004], the exact nature of the relation is not specified as is the case with most other systems described in this section, including WWW2REL. Finally, the system also differs from WWW2REL in that the KPs, or rule templates, are manually devised rather than induced.

3.4.6 PASTA

PASTA [Gaizauskas et al., 2003] is an acronym for Protein Active Site Template Acquisition. It is a biomedical IE system which extracts two relation types, called "template relations" by the authors, and three so-called "template elements" from biomedical abstracts. The two relation types are IN_PROTEIN and IN_SPECIES and the three elements are RESIDUE, PROTEIN and SPECIES. The system features the three following stages, or levels, of processing.

1. Text processing
    (a) section analysis
    (b) tokenization
    (c) sentence splitting
2. Terminological processing
    (a) morphological analysis (protein-specific affixes)
    (b) lexical lookup (biological lexicons)
    (c) terminology parsing

3. Syntactic and semantic processing

   (a) PoS tagging

   (b) phrase tagging

   (c) predicate-argument representation

4. Discourse processing

   (a) ontology-based inference

   (b) coreference resolution

   (c) extension of existing ontology

Like SGPE (see subsection 3.4.1) PASTA is custom-tailored for the biomedical domain, or rather the subdomain of protein structures, because it makes use of a protein lexicon in the terminology processing phase to identify protein elements in the abstracts. The system has been applied to a corpus of 1,513 MEDLINE abstracts on macromolecular structures and in the task of template (i.e. relation) extraction it achieves an overall precision of 65%.

3.4.7 [Alfonseca et al., 2006b]

[Alfonseca et al., 2006b, Alfonseca et al., 2006a] describe how so-called "rote extractors" can be applied to unannotated text to learn KPs instantiating any type of semantic relation. A rote extractor estimates the probability of a relation r(p,q) between two entities, p and q, on the basis of the surrounding context $C_1pC_2qC_3$. The probability of the relation given a specific context can be calculated as the number of times the two related elements r(x,y) appear in this context $C_1xC_2yC_3$, divided by the total number of times that x appears in the context with any other word.

$$P(r(p,q)|C_1xC_2yC_3) \doteq \frac{\sum_{x,y \in r} c(C_1xC_2yC_3)}{\sum_{x,x} c(C_1xC_2zC_3)}$$

where x is known as the *hook* and y is called the *target*. The basic extraction procedure is as follows.

1. Download a "target corpus" by querying the WWW for a number of seed term pairs
2. Extract seed contexts and identify recurrent patterns
3. Download a "hook corpus"
4. Apply patterns from 2) to the hook corpus and extract new pairs
5. Compute pattern precision using the above formula
6. Repeat

The authors address two weaknesses in this approach, namely that the patterns are inflexible and may be ambiguous (i.e. extract instantiations of different relation types). To overcome the first limitation the patterns are generalized by enriching the texts with PoS and NER tags, allowing wild cards and computing string edit distances. To resolve the ambiguity issue the authors build a table in which the rows are different hooks, the columns are different relation types and the cells are sets of relation instances, possibly the empty set. Pattern precision is then computed by measuring to what extent the individual patterns retrieve instances, or "targets", not recorded in the target cell. Precision rates are reported not for the individual patterns but for each of ten different relation types, and they range from 14% (birth-place) to 96% (death-year).

Again, introducing NER may boost precision, but reduces system portability, and for this reason WWW2REL does not rely on NER. Also, assessing KP precision in the fashion of [Alfonseca et al., 2006b, Alfonseca et al., 2006a] is not possible when extracting instances of complex many-to-many relations like "induces" and "may_prevent" (chapter 5) for which no exhaustive gold standard can be obtained. Hence the need for a manual evaluation of KP precision. Further, WWW2REL does not download hook corpora, but hook+{KP} corpora, where {KP} is the set of all filtered patterns learned for the target relation type. This is done to measure the precision of each KP based on a manual evaluation (see section 6.3). Finally, because WWW2REL operates on sentence fragments rather than complete documents it ignores the left and right contexts when discovering KPs.

3.4.8 [Charniak and Berland, 1999]

[Charniak and Berland, 1999] report on a system which can extract PART_OF relations from a 100 million word newspaper corpus (the North American News Corpus) with a precision of 55%. The approach is inductive and domain-independent. Five different patterns retrieve a number of candidate parts and these are then filtered on morphological and statistical grounds. Candidates containing suffixes suggesting qualities (e.g. -ing or -ness) are deleted and association strength is measured using log likelihood ratios between the wholes and the parts (see subsection 4.1.2 for the formula). Even using a corpus of 100 million words the authors list data sparseness as a very real problem.

3.4.9 [Girju and Moldovan, 2002]

The relation extraction system described in [Girju and Moldovan, 2002] is interesting because it targets the causal relation type which embraces the two UMLS relations investigated in chapter 4 of this thesis. The authors distinguish between simple causatives (e.g. *generate*), resultative causitives (e.g. *kill*) and instrumental causatives (e.g. *poison).* They focus on intra-sentential syntactic patterns of the form "NP1 verb NP2" in which the verb is a simple causative. The causal markers are induced by extracting verbs which connect word pairs related by the CAUSE_TO relation in Wordnet. While the patterns are induced automatically, their filtering is carried out by manual means. The filtering strategy involves semantic constraints on the NPs and the verbs, for example the most general WordNet subsumer of the effect argument must be either a *human action*, *phenomenon*, *state*, *psychological feature* or *event*. Another constraint is that

verbs with a high number of a Wordnet senses are penalized. Evaluating the system against the judgments of two humans an average precision of 66% is reported on a set of 300 relation instances of which 230 are causative.

3.4.10 [Nenadic and Ananiadou, 2006]

The system developed by [Nenadic and Ananiadou, 2006] is similar to KnowItAll (subsection 3.4.5) in that it extracts "semantically related terms" rather than relation instances of a predefined relation type. On the other hand, it is similar to SGPE and PASTA in that it extracts these unlabelled relationships from biomedical MEDLINE abstracts. More specifically, a richly annotated and term tagged subset of 2,000 MEDLINE abstracts known as the Genia corpus[27]. The system identifies the following three types of term similarities.

1. Lexical term similarities
2. Syntactic term similarities
3. Contextual term similarities

Lexical term similarities are NP internal relations between head and modifier(s) and will not be discussed given the topic of the thesis. The difference between syntactic and contextual term similarity is rather subtle. While syntactic similarity is identified by means of Hearst-like patterns (i.e. simple strings like "such as", "e.g."), contextual similarity is established through more generalized patterns not unlike those described in subsection 3.4.7. These patterns are called "context patterns" and include lemma and PoS tags. As a measure of term relatedness "the distance between two terms is calculated as the mean of the sum of distances (the number of edges) of their respective classes from the nearest common ancestor in the Genia ontology" [Nenadic and Ananiadou, 2006]. Combining the three similarity measures the authors achieve f scores of 68% for semantically related (distance ≤ 3) and 37% for highly related terms (distance ≤ 1). They conclude that syntactic patterns have a high precision but low recall, while the opposite is the case for context patterns. However, as discussed in subsection 3.1.1 using the web as a corpus can be a way of overcoming this inherent limitation of syntactic patterns, or KPs.

The main difference between WWW2REL and the system developed by [Nenadic and Ananiadou, 2006] is that the text WWW2REL operates on is neither term tagged nor annotated with ontological information. Another key difference is that WWW2REL extracts instantiations of specific relation types rather than unlabelled associative relations. Finally, only the "context patterns" in [Nenadic and Ananiadou, 2006] are induced, whereas the syntactic patterns are compiled from the literature.

3.4.11 System comparison

Table 7 summarizes the differences and similarities of the pattern-based relation extraction systems outlined in this section as compared with the WWW2REL system

[27] see www-tsujii.is.s.u-tokyo.ac.jp/~genia

Table 7: Comparison of pattern-based relation extraction systems

system/subsection	portability	KP ind.	text source	non-hier.	iterative
Espresso	high	yes	TREC-9	yes	yes
Snowball	low (NER)	yes	newspapers	yes	yes
SGPE	low (Bio. filters)	no	MEDLINE	no	no
PASTA	low (Bio. lexicons)	yes	MEDLINE	yes	yes
RelationAnnotator	low (Bio. NER)	no	WWW snip.	yes	no
KnowItAll	high	no	WWW docs	no	yes
3.4.10	low (Bio. ontology)	yes/no	MEDLINE	yes	no
3.4.7	low (NER)	no	WWW docs	yes	yes
3.4.8	high	yes	newspapers	no	no
3.4.9	low (Wordnet)	yes	TREC-9	yes	no
WWW2REL	high	yes	WWW snip.	yes	no

implemented in this thesis. The column "portability" indicates whether each system relies on domain-dependent techniques or resources and if so which ones. The "non-hier." column indicates whether each system extracts also non-hierarchical relation types, and "KP ind." indicates whether KPs are induced from text or produced introspectively. While most systems induce their patterns automatically, few systems are truly portable (only Espresso, KnowItAll and [Charniak and Berland, 1999]) and few systems operate on noisy WWW text fragments (only RelationAnnotator, KnowItAll and [Alfonseca et al., 2006b]). In comparison with the systems described in this survey WWW2REL is special in that it is both portable, inductive, works for any conceivable relation types and uses the web as a corpus. The three most similar systems either operate on less noisy text (Espresso), have low portability (RelationAnnotator) or focus on IE tasks (KnowItAll).

3.5 Conclusion

This chapter outlined the methodological considerations which have a bearing on the experiments which are to follow in chapters 4 and 5. It discussed the strengths and weaknesses of using the WWW as if it were a corpus. Secondly, it described the related fields of text mining, data mining, web mining, Information Retrieval (IR) and Information Extraction (IE) and concluded that the system implemented in this thesis is a text mining system, although the metrics by which it is evaluated originate from IR and Machine Learning. As for the novelty of the semantic relations retrieved by the system, it was concluded that these are only new in the sense that they are recovered from diverse sources and presented in a structured manner to the user who may not have been aware of their existence and may find them useful for updating his termbase. Finally, the chapter made references to a range of existing pattern-based relation extraction systems, and it was concluded that WWW2REL differs from most existing applications by combining the following three features.

- It operates exclusively on fragmented textual content on the WWW.

- It induces KPs automatically from unannotated text relying on filtering techniques described in section 4.2.
- It can be applied to any conceivable relation type.
- It is conceived as a domain-independent application to ensure portability (this is tested in section 6.5).

Chapter 4 will now describe the first step in the WWW2REL ontology extension framework, namely discovering and filtering KPs for the target relation(s).

4 Pattern discovery and filtering

This chapter and chapter 5 describe the implementation and evaluation of WWW2REL, a highly portable relation extraction system operating exclusively on text snippets found on the WWW. As the system is inspired by the DIPRE technique (subsection 2.1.2), it only makes use of information which already exists in the specific ontology it aims to augment. Apart from the BNC (which is used as an optional termhood filter and could be exchanged for the freely available Google ngram counts[28]) all information used by the system is freely accessible on the WWW. In fact, figure 9 illustrates how the WWW serves multiple purposes in the system implementation. Thus all arrows marked by "WWW" indicate that text snippets from the WWW are used at these points to either discover or filter KPs, retrieve relation instances or even rank these by assessed reliability. The dotted lines in the figure indicate phases which are only mandatory during system evaluation, but which could be omitted when subsequently running the evaluated system.

The figure also provides a roadmap to most of the remaining sections of the thesis. The starting point is marked by the filled circle named "ontology", representing subsection 4.1.1. Subsection 4.1.2 discusses how the selection of the target relation type(s) may be guided by corpus analysis, for example in the absence of domain experts. Subsection 4.1.3 describes issues related to the selection of term pairs instantiating the target relation type(s). The topic of subsection 4.1.4 is the retrieval from the WWW of a training corpus of text snippets containing these pairs in context. Finally, subsection 4.1.5 discusses how pattern candidates can be extracted from such a training corpus.

While section 4.2 introduces various techniques which can be used to filter out noisy patterns, section 5.1 outlines the actual implementation of a system using these filtered knowledge patterns to extract relation instances from the WWW. Section 5.2 introduces the manual system evaluation setup and discusses issues related to such an evaluation. In section 5.3 a number of ranking schemes and heuristics are devised, and sections 5.4 and 5.5 report in great detail on the performance of these schemes and heuristics in eleven actual system tests involving the four relation types for which the system is tested. Finally, section 6.4 explores the degree of new knowledge retrieved by WWW2REL and also assesses recall versus the starting ontology (the UMLS Metathesaurus).

[28] See www.ldc.upenn.edu/Catalog/

Figure 9: The role of WWW in the system implementation

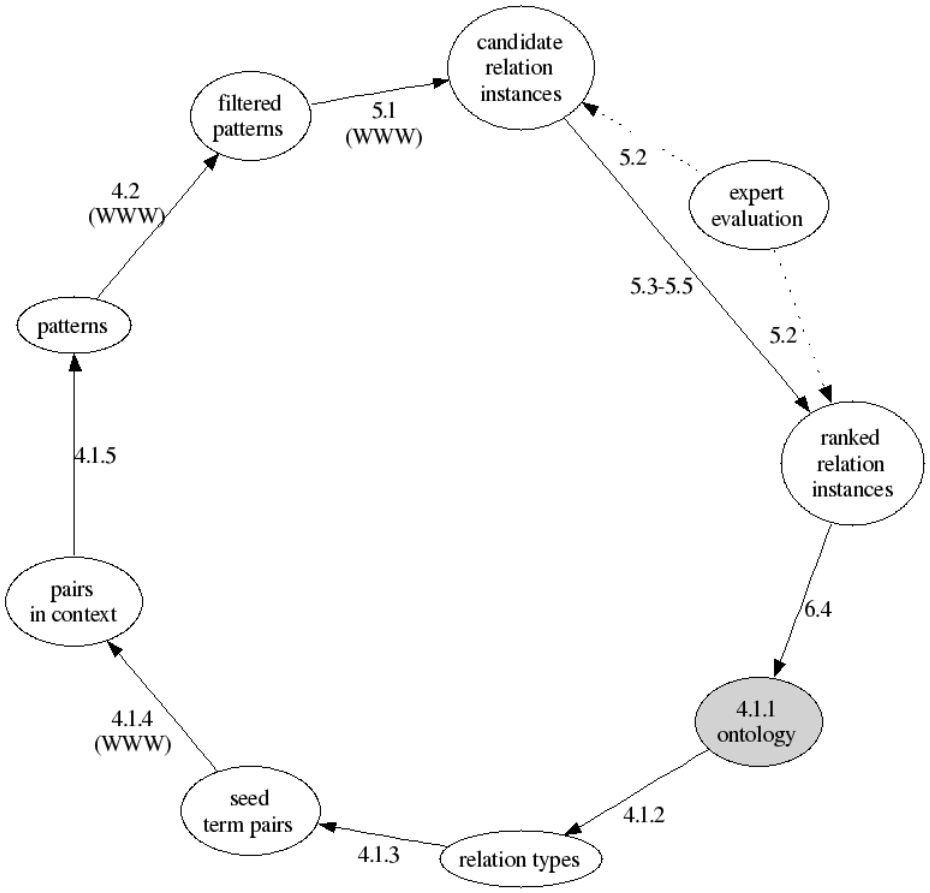
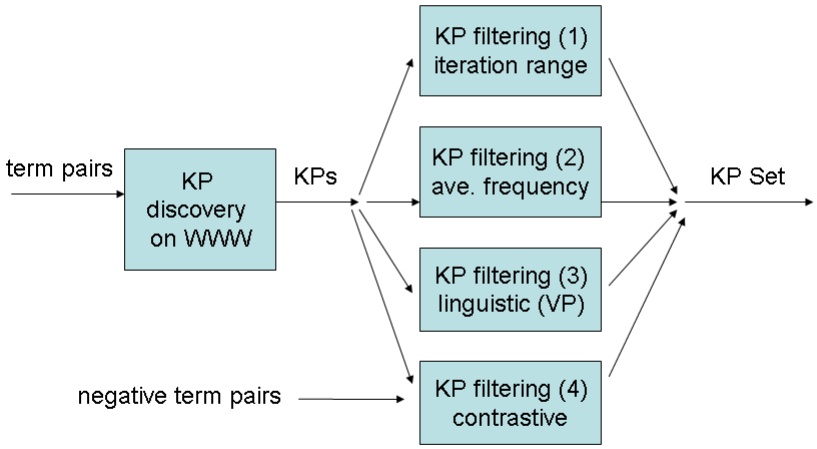


Figure 10: WWW2REL diagram: discovering KPs and extracting relation instances from the WWW

Step 1 - Discovering KPs
- no direct evaluation
- filter thresholds empirically established
- choice of filter(s) depends on the semantic relation



Step 2 – Finding relation instances
- searching for relation(term1,term2) knowing term1 only
- human judges establish a gold standard from system unfiltered output (i.e. for all ranking algorithms taking all candidates would give a recall of 100%)
- evaluation of recall and precision of each ranking method

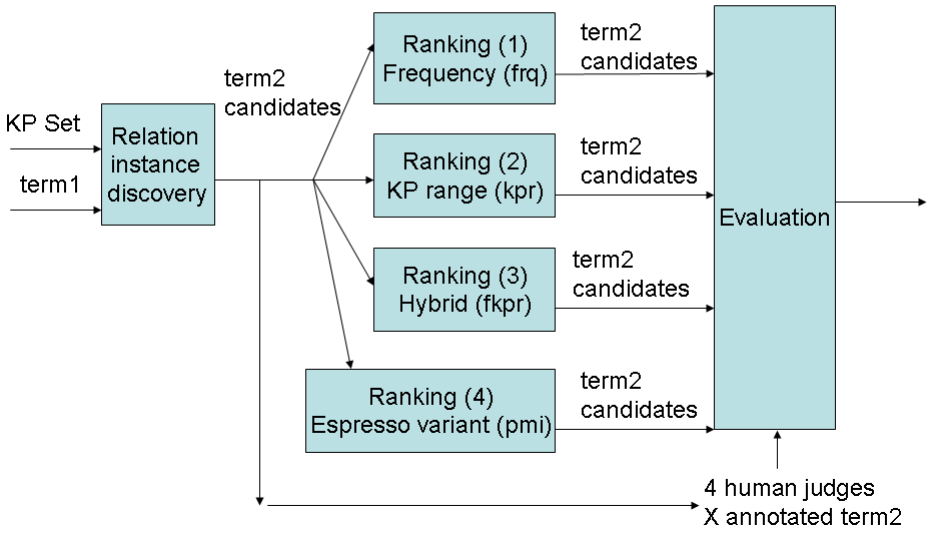


Figure 10 also summarizes the contents of chapters 4 and 5, but in terms of system input and output. It illustrates the two steps of KP discovery (top diagram) and relation instance extraction (bottom diagram). In the first step, system input is four sets of term pairs and output is four reduced sets of KPs. In the second step system input is four sets of KPs combined with a number of input terms (see table 38), and output is four sets of ranked relation instances.

4.1 KP discovery

This section describes the methodology used to discover knowledge patterns in WWW text snippets based on a number of seed term pairs extracted from the target ontology. The KPs instantiate four different semantic relation types, namely two classical relations (synonymy and ISA) and two arguably less universal relations ("induces" and "may_prevent"). The section starts off by analyzing to what extent "induces" and "may_prevent" relations indeed seem to be particularly characteristic of biomedical text (subsection 4.1.2) and procedes to examine a number of practical issues complicating the selection of seed term pairs from the UMLS Metathesaurus (subsection 4.1.3), the retrieval of term pair contexts (subsection 4.1.4) and identifying KP candidates from these contexts (subsection 4.1.5).

4.1.1 Start with a known ontology

The UMLS knowledge sources are an ideal starting point for the construction of our relation extraction system for two very simple reasons. First, they are provided free of charge by the US National Library of Medicine. Secondly, they form the most comprehensive knowledge representation system for the domain of Biomedicine, a domain characterized by such a fast-paced concept formation that semi-automatic tools are needed to keep track of its terminology. For more details on the UMLS the reader is referred to subsection 2.3.4.

4.1.2 Select target relation(s)

Having selected an ontology, the next step is the selection of a number of semantic relation types for which new instances should be identified by the system. A fundamental question at this point is: which are the most important semantic relation types for the target domain, i.e. Biomedicine in this case? Although asking domain experts may provide a quick answer, these may not be at hand or may not have given much thought to the ontological characteristics of their domain.

Over the decades, practical terminology work has revealed the two conceptual relations, ISA and PART_OF, along with the semantic relation of synonymy to be domain-independent. However, any domain will typically feature a number of additional conceptual relations which are, if not unique to the domain, then at least characteristic of it. Since VPs typically establish semantic relations between nominal arguments and can be used as knowledge probes in terminology (see e.g. [Christensen, 2002]), one way of empirically getting an indication as to which relation types are the most important ones for a particular domain is to compare the frequencies of all VPs in a comprehensive

Table 8: A contingency table of observed and expected frequencies

	observed	frequencies		expected	frequencies
	y=VP	y≠VP			
x=BioMed	O_a	O_b	R_1	$E_a = \frac{R1*C1}{N}$	$E_b = \frac{R1*C2}{N}$
x=BNC	O_c	O_d	R_2	$E_c = \frac{R2*C1}{N}$	$E_d = \frac{R2*C2}{N}$
	C_1	C_2	N		

corpus of text specific to the domain in question versus the frequencies of the same VPs in a balanced general language corpus. As described in section 3.1 the 100 M word British National Corpus[29] (BNC) is a commonly used general language corpus.

This subsection presents two experiments carried out in order to examine to which degree the two UMLS relation types, "induces" and "may_prevent", actually appear to be instantiated in a large biomedical corpus, namely the 90 M word BioMed Central's open access full-text corpus[30]. The first step in the experiments is to extract all VPs from the text corpora and record their frequencies. The VPs are extracted from lemmatized, POS tagged[31] and chunked[32] corpora using the following CQL[33] search template.

[chunk='.-VP']+ [chunk='.-PP']?

This template will extract even complex VPs like

being/being/B-VP possibly/possibly/I-VP influenced/influence/I-VP by/by/B-PP

where the positional attributes are wordform/lemma/chunk.

Having extracted two sets of VPs (one for the analysis corpus, BioMed, and one for the reference corpus, BNC), it is time to consider relevant statistical measures. A number of statistical measures have been used to assess the degree of association between two lexical items (e.g. finding collocations) or between one lexical item and two corpora (e.g. term recognition). The approaches are typically based on a so-called contingency table (cf. table 8) but differ with regard to the importance attached to rare events[34], i.e. rare words or phrases. To illustrate how association measures based on simple contingency tables can help identify VPs (and thus to some extent semantic relation types) which are characteristic of a domain-specific corpus, this subsection will apply the following two association measures to the BioMed using the BNC as reference corpus.

The first association measure is the so-called log odds ratio which is given by:

[29] see www.natcorp.ox.ac.uk for details

[30] www.biomedcentral.com/info/about/datamining/

[31] www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[32] http://chasen.org/~taku/software/yamcha/

[33] Corpus Query Language - see www.ims.uni-stuttgart.de/projekte/CorpusWorkbench

[34] see [Evert, 2004] and www.collocations.de for details

$$log - odds - ratio(VP) = log(\frac{O_a * O_d}{O_c * O_b})$$

where the variables O_a, O_b, O_c, O_d are the observed frequencies as defined in table 8. Log-odds ratio[35] ignores the expected frequencies (defined in the same table) and thus emphasizes the importance of the observed ones implying a bias towards rare events.

The second association measure is the log-likelihood ratio given by

$$log - likelihood - ratio(VP) = 2 * \sum_{i=a}^{d} O_i * log(\frac{O_i}{E_i})$$

Since log-likelihood[36] takes the expected frequencies into consideration, it provides a more conservative estimate of the association between a particular VP and the two corpora. For both association measures minimum VP frequencies of 100 in the biomedical corpus and 1 in the BNC were enforced so as to prevent overly specialized verbs, typos and so on from cluttering the lists. Tables 9 and 10 show the top 20 (and bottom 5) VPs in the BioMed corpus as ranked by their degree of association with this corpus versus the BNC (as measured by log-likelihood and log-odds ratio, respectively).

When comparing the two lists of VPs most characteristic of the BioMed corpus, it is quite clear that log-odds ratio (table 10) is useful in finding the most peculiar VPs which are relatively rare even in the biomedical corpus but hardly occur at all in the BNC. For example, the verb "lyse" ("be lysed in") is indeed one such peculiar VP which is unique to the domain of Biomedicine or Biology. However, since the high-ranking VPs based on log-odds ratios on average only occur a few hundred times in the 90 M word BioMed corpus, this association measure appears to be a less appropriate tool than log-likelihood when trying to identify the most significant semantic relation types in a domain. It is interesting, though, that most of the VPs in this table are in the passive voice, but this is, of course, a general feature of academic writing rather than of Biomedicine as such. Finally, the two VPs, "can be downloaded from" and "can be accessed", probably only rank high due to the age of the BNC.

The VPs in table 9, on the other hand, are much more common. There are quite a few verbs of communication and existence[37], which are characteristic not of Biomedicine but of academic writing in general. One way of eliminating these academic VPs from the lists would be to compare VP frequencies in the biomedical corpus with their frequencies in a whole range of other specialized but non-biomedical corpora. However, such a comprehensive experiment would fall outside the scope of this thesis, and the Academic Word List discussed in subsection 2.2.2 might, in fact, be a useful filter. What is important in this pilot experiment is that six[38] out of the top twenty VPs can, in fact, be categorized as causal (these are the VPs capitalized in table 9). Three of

[35] See perl script in appendix 8.3

[36] See perl script in appendix 8.2

[37] e.g. "show", "indicate", "suggest", "demonstrate", "report"

[38] "induce", "inhibit", "result in", "increase", "decrease", "treat with"

Table 9: characteristic VPs in the BioMed corpus ranked by log-likelihood versus the BNC

f(biomed)	f(BNC)	LL	lemmatized VP
111,800	38,245	52,040	use
75,911	28,469	31,700	show
37,999	7,175	28,951	indicate
42,631	12,360	23,481	contain
37,999	844	21,454	INDUCE
17,087	1,014	20,297	compare to
43,312	17,498	16,416	suggest
13,405	827	15,780	participate in
14,222	1,254	15,190	be associate with
17,837	2,796	15,086	demonstrate
9,099	2	14,278	distribute under
10,054	308	13,417	encode
10,118	561	12,200	INHIBIT
15,921	3,197	11,668	base on
14,252	2,679	10,887	perform
13,695	2,614	10,364	RESULT IN
21,774	7,418	10,176	INCREASE
8,460	575	9,726	DECREASE
7,479	281	9,689	TREAT WITH
20,960	7,244	9,639	report
...	...	...	...
2,119	90,915	-93,662	think
13,492	134,041	-93,962	do
3,310	103,993	-102,113	know
2,143	236,395	-266,490	say
715,189	1,810,867	-319,581	be

Table 10: Characteristic VPs in the BioMed corpus ranked by log-odds versus the BNC

f(biomed)	f(BNC)	LO	lemmatized VP
9,099	2	8.60	distribute under
709	1	6.74	compare to control
620	1	6.61	be stimulate with
965	2	6.36	harbor
4,555	12	6.12	be permit in
364	1	6.08	be downloaded from
346	1	6.03	be highly express in
337	1	6.00	be quantify use
308	1	5.91	silence in
301	1	5.89	be extract use
292	1	5.86	be overexpressed in
546	2	5.79	be conduct use
4,127	16	5.73	can be access
510	2	5.72	profile of
499	2	5.70	be evaluate use
247	1	5.69	be lyse in
231	1	5.62	generate use
226	1	5.60	be evaluate with
598	3	5.47	be detect use
180	1	5.37	be transiently transfected with
...	...	...	...
531	35,924	-4.04	look
118	9,597	-4.22	buy
124	10,211	-4.23	watch
2,143	236,395	-4.53	say
109	26,145	-5.30	have get

the VPs are likely to instantiate the "induces" relation ("result in", "increase" and "induce") and three will typically instantiate the "may_prevent" relation ("inhibit", "decrease" and "treat with"). This would seem to corroborate the fact that causal relations indeed play an important role in Biomedicine. Incidentally, this claim is also backed by [Girju and Moldovan, 2002] who observe in a study of causality markers that 58% of all NP pairs linked by causal verbs in WordNet 1.7 are tagged as belonging to the domain of Medicine.

In conclusion, looking at frequent, domain-specific verbs can help us identify which relation types are important to a particular domain, but any statistically induced list should, of course, be supplemented by knowledge of the actual domain. For Biomedicine, it is a well-known fact that all new drugs have to be tested for potential side effects and these must constantly be monitored and registered. Moreover, copycat drugs introducing additional side effects are also becoming an increasing problem. Thus the UMLS "induces/induced_by" relation type is clearly a very important one in the biomedical domain. Since the purpose of most drugs is to prevent or reduce the severity of various pathological conditions, the "may_prevent" relation type is also important. The statistical association measures for VPs in a biomedical corpus seemed to align fairly well with the actual ontological design of the UMLS.

In the KP discovery experiments which follow we will focus on term pairs instantiating the two causal relations just mentioned, the terminologically fundamental ISA relation and synonymy. In other words KPs will be discovered for

- ISA
- induces
- may_prevent
- synonymy

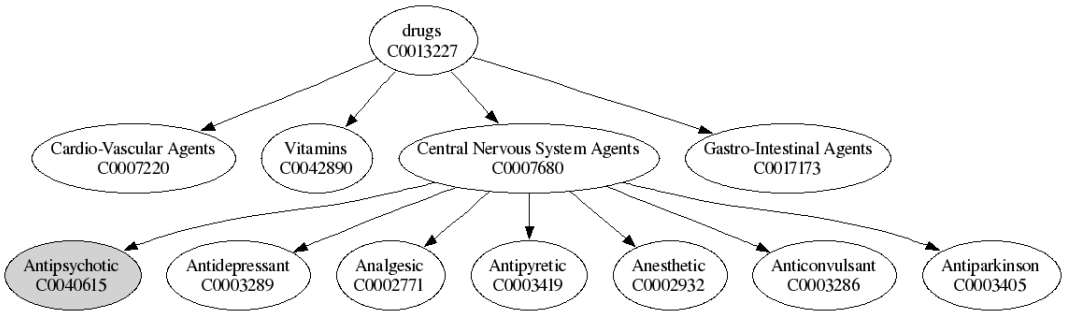
4.1.3 Select term pairs instantiating these relation(s)

Before any actual web search engine queries can be executed and a training corpus be built, the following steps must be completed.

1. Extract UMLS concept pairs[39] from database for target relation(s)
2. Apply term variant expansion
3. Apply lexical filtering
4. Apply frequency filtering

[39] or single concepts in the case of synonymy

Figure 11: Drug hyponyms from UMLS used to discover ISA KPs



Extracting concept pairs Initially, a completely random extraction of term pairs from the UMLS Metathesaurus was considered (see the script in appendix 8.4). However, for the ISA relation there are no less than 318,391[40] concept pairs recorded in the database. Extracting all recorded term variants for each of these concepts to produce all possible term pair combinations would be rather time consuming and unnecessary. Thus it was convenient to enforce certain restrictions on the semantic types of the ISA arguments to reduce the size of the search space and make the setting more realistic.

One possible application of disvering knowledge patterns for the "may_prevent" and "induces" relations is to implement a system which can identify side effects and beneficial, intended effects given a drug and vice versa. Accordingly, it seemed natural to restrict the extraction of concept pairs for the ISA relation to a fragment of the drug subontology, although any fragment of the UMLS would presumably have been equally adequate. Also, "Clinical Drug" happens to be the most frequent semantic type in both the "may_prevent" and "induces" relation (cf. subsection 5.2.4). Figure 11 visualizes a small ontological fragment containing frequent hyponyms of the UMLS concept "drug". Thus the first step of KP discovery for the ISA relation is to extract concept pairs from the UMLS in which one argument is either a vitamin, a gastro-intestinal agent, a cardio-vascular agent or one of the subtypes of a central nervous system agent. This is done using the script in appendix 8.5.

The numbers in the figure are the Concept Unique Identifiers, or CUIs, by which the concepts are indexed in the UMLS Metathesaurus, and the filled circle in the figure marks the category of drugs, namely antipsychotics, for which the ISA KPs will be tested. Of course, as is standard practice in Machine Learning no antipsychotics are used when discovering KPs (neither by the system nor by the author!).

As described under the following headings, a few search space restrictions are also enforced when extracting causal pairs (see appendix 8.6) and synonym pairs (see appendix 8.14) from the database.

Term variant expansion Most concepts recorded in the UMLS are associated with 4-5 term variants, and when compiling a training corpus, term variance raises the question of whether to randomly select just a single term variant for each concept, or to

[40] In the 2006AB edition of the UMLSKS

Table 11: Effect inducing drugs (examples)

Ingredient	Strength	Dose form
Phenylephrine	2.5 MG/ML	Nasal Spray
Cetirizine	5 MG	Oral Tablet

search for all possible term variants. Since some term variants are much more commonly used (e.g. "Vitamin C") than others (e.g. "ascorbic acid"), but frequency information is not recorded in the UMLS, it was decided to produce all possible combinations including all term variants.

Lexical filtering However, in the case of Biomedicine some term variants are extremely specialized. For example, drug entities in the UMLS Metathesaurus are characterized by the following three components.

Active ingredient + Strength + Dose form

Two examples can be seen in table 11. Clearly, searching for exact strings containing all three elements will result in data sparseness at best. One way of reducing the level of detail would be to ask the UMLS database for the immediate hypernyms of each term. For example, the pair "phenylephrine 6 mg/ml oral solution <=> mydriasis" becomes "methylparaben <=> pupil disorders" when querying for the hypernyms of the term pair. The induces relation between the latter pair is not as clear as the one between the former pair, and thus using hypernyms will not necessarily overcome the data sparseness issue, but will certainly introduce invalid relations.

Another way of reducing the level of detail would be to use the UMLS relation "has_tradename" and query for the actual drug trade names. However, the active ingredients are likely to be more universally used than specific drug trade names, which go in and out of fashion. In conclusion, the best way of reducing the level of detail in instances of "induces" and "may_prevent" relations is to make use of the "has_ingredient" relation because this will give us the first column of names in table 11 and ignore strengths and dose forms (see also appendix 8.6).

Lexical filtering may further normalize the term variants and boost recall. [Aronson, 2001] discusses the challenges involved in developing the MetaMap application which maps term variants found in biomedical text to the UMLS Metathesaurus concepts they represent. He suggests a number of ways of conflating strings which essentially refer to the same term variant. The techniques listed below include most of them.

- Removing non-essential parentheticals
- Syntactic uninversion[41]
- Removing case variation
- Removing hyphen variation

[41] for example "Acid, ascorbic" => "ascorbic acid"

Table 12: Example term variants for an induces/induced_by relation

concept1 variants	concept2 variants
Ascorbic Acid	Hyperoxaluria
Acid, Ascorbic	Oxaluria
L-Ascorbic Acid	
Vitamin C	
C Vitamin	
ascorbic acid preparation	

In the following experiments all possible combinations of all term variants are produced and the only lexical filtering applied to the variants are the four points listed above, including the removal of duplicates and the deletion of punctuation marks, which are not recognized by web search engines.

Frequency filtering Table 12 lists all the UMLS term variants for each of the concepts in an example "induces" relation. A practical problem is that many term variants (even lexically filtered ones) are too rare, and when combined as a pair in the search for candidate knowledge patterns, they will generate queries yielding zero hits on the search engines. A simple way of avoiding data sparseness problems is to select at random term variants which co-occur with a certain minimum frequency. An arbitrary threshold of 100 hits (on Google) has been set as a minimum co-occurrence frequency of all term pairs. The hit counts are obtained by the script in appendix 8.11.

Table 13 illustrates the impact of the lexical filtering and frequency filtering on the number of actual term variant pairs used for discovering ISA patterns. It also shows that using the WWW to expand a special ontology does not completely banish all data sparseness problems. With the exception of ISA relations having "Vitamin" as the hypernym, only a small fraction of the lexically filtered term pairs have Google co-occurrence frequencies exceeding 100 hits (see column f(Google)>100 in table 13). Overall, only about 4.7% of the lexically filtered term pairs met this co-occurrence frequency threshold which was deemed necessary to get a sufficient number of snippets from which to reliably extract knowledge patterns.

For a hierarchical relation like ISA one way of reducing the sparseness problem would be to simply use more general hypernyms. For example, changing the hypernym in <X;Antidepressant> to <X;drug> would likely yield more hits on Google. However, this strategy is not an option with non-hierarchical relations like "induces" and "may_prevent" (cf. the example with "phenylephrine above). Since ISA is such a fundamental and domain independent relation another approach to coping with data sparseness would be to find training pairs in other terminological resources outside the specialized ontology, but in the case of less universal relation types, training pairs must typically be found in the existing ontology one wishes to augment. In such cases, the data sparseness problem appears to affect even WWW-based approaches to KP discovery.

Finally, synonym training pairs were found by retrieving all UMLS concept pairs

Table 13: Effects of variant expansion, lexical and frequency filtering for ISA

ISA relation	#concept pairs	#variant pairs	#filtered pairs	f(Google)>100
X;GI Agent	64	2,492	1055	7 (0.7%)
X;vitamin	54	1,400	375	211 (56.3%)
X;CV Agent	43	1,526	435	16 (3.8%)
X;Antidepressant	49	1,400	678	55 (8.1%)
X;Analgesic	85	4,944	2,616	31 (1.2%)
X;Anesthetic	29	960	464	6 (1.3%)
X;Anticonvulsant	171	7,502	2,600	95 (3.7%)
X;Antipyretic	10	210	87	13 (15%)
X;Antiparkinson	56	2,850	973	3 (0.3%)
TOTAL	507	23,284	9,283	437 (4.7%)

Table 14: Lexically filtered, relatively frequent UMLS term pairs for KP discovery (examples)

INDUCES	MAY_PREVENT
alcohol;unconsciousness	prilocaine;pain
alcohol;vomiting	flunisolide;asthma
...	...
SYNONYMY	ISA
hypotension;low blood pressure	analgesic;gabapentin
pregancy;gestation	valproic acid;anticonvulsants

for a specific relation type, in this case "induces", generating all term variants of each concept and randomly selecting pairs on the basis of their term type code (TTY). To be included one term variant must have the term type "PN", or preferred name, and the other variant must be explicitly marked as a synonym[42]. Also, the pair must co-occur at least 100 times on Google. Enforcing these restrictions and looking only at concepts which are arguments of the "induces" relation resulted in the list which can be seen in appendix 8.29.

4.1.4 Build a training corpus

Table 14 gives a few examples of the training pairs which will now be used to compile four web corpora, one for each of the four relation types. The pairs have been lexically filtered and co-occur at least 100 times on Google. The complete lists can be seen in appendices 8.27, 8.28, 8.29 and 8.30. In order to identify KPs for the four relation types, one needs a large number of actual contexts in which term pairs instantiating these relations co-occur. The simplest approach is to download a number of text

[42]see the script in appendix 8.14 and the UMLS website for details.

Table 15: Knowledge patterns in context

left	term1	middle	term2	right
<causes of>	diarrhea	<include>	parasites	, some cancers ...
	diarrhea	<induced by>	bacteria	
...to minimize	the stomach irritation		aspirin	<can cause>
a <side effect of>	nicotinic acid	<is>	flushing	
	contaminated water	and the	diarrhea	it <can cause>

snippets for each of the training term pairs using Google[43] queries like:

1. ISA: "ketamine * analgesic"
2. INDUCES: "carbon dioxide * headache"
3. MAY_PREVENT: "mineral oil * constipation"
4. SYNONYMY: "dyspnea * breathlessness"

in which the "*" is a word wildcard representing at least one word. Two search parameters are specified: matches must be "allintext" and automatic results filtering is activated to weedout duplicate content and host crowding. The script in appendix 8.7 provides the details of the procedure.

Although some training pairs co-occur thousands of times on Google, most pairs co-occur only hundreds of times. Consequently, it was decided to retrieve only the top 100 snippets (hits) for each training pair so as to make sure that all pairs are equally represented in the training corpora. The individual snippets typically contain only 1-3 sentences (on average some 22 tokens per snippet), but this is enough context for our purposes.

4.1.5 Identify pattern candidates for the target relation(s)

From the small collections of term pairs in context, pattern candidates can now be extracted. The main question which remains is which part of the term contexts to examine. [Alfonseca et al., 2006b] defines a term pair co-occurrence as having three contexts: left context, middle context and right context. As table 15 illustrates, useful knowledge patterns can occur in all three contexts, although, intuitively and empirically, the middle contexts seem more useful than the left and right contexts[44].

> From our experiments with English-language documents, we have found the middle context to be the most indicative of the relationship between the elements of the tuple. [Agichtein and Gravano, 2000]

While [Alfonseca et al., 2006b, p54] propose a maximum distance of eight tokens between the two terms in each pair, the limit used in [Turney, 2006] is three tokens.

[43] Google is used because Yahoo offers no word wildcards

[44] Although it can be void in case of relative clauses

[Ravichandran and Hovy, 2002] and [Pantel and Pennacchiotti, 2006] use a suffix tree constructor, which finds the longest common substrings of all lengths within the entire sentence.

Rather than specifying a specific distance in terms of number of tokens, it seems more intuitive to let the distance be either linguistically determined or indefinite. In two of the experiments which follow, the distance is linguistically determined ("may_prevent" and "induces"), and the other two cases (ISA and synonymy) it is indefinite. It should be mentioned that there is an upper limit on the number of words matched by the Google word wild card, "*". Although the exact number is not documented, it appears to be about eight.

Finally, it should be noted that [Ravichandran and Hovy, 2002], for example, enforce a frequency threshold and discard low frequency pattern candidates. However, these patterns may, in fact, be very good, and thus in the experiments which follow, all pattern candidates are kept (as advocated in [Pantel and Pennacchiotti, 2006]).

The following query templates are used to extract knowledge pattern candidates from the middle contexts of instances representing the four relation types.

1. "induces" and "may_prevent"
   (a) <t1> ($dummy{0,2} $vp+ $pp?) $np* <t2>
   (b) <t2> ($dummy{0,2} $vp+ $pp $np* <t1>
2. synonymy
   (a) <t1> .* <t2>
   (b) <t2> .* <t1>
3. ISA
   (a) <hypernym_sing> .* <hyponym>
   (b) <hypernym_plur> .* <hyponym>
   (c) <hyponym> .* <hypernym_sing>
   (d) <hyponym> .* <hypernym_plur>

The reason more query templates are needed for the ISA relation than for synonymy is that ISA is not a symmetrical relation like synonymy is. For synonymy, there are no particular interdependencies between the patterns and the linguistic form of their arguments. Also, changing the position of the two arguments typically has no impact on the choice of knowledge pattern. Synonymy KPs are, for the most part, bidirectional. For example, "emesis <also known as> vomiting" and "vomiting <also known as> emesis"[45] are equally grammatical. For the ISA relation, however, most patterns are unidirectional and switching the position of the arguments will typically also necessitate a change of pattern. The pattern "and other", for example, requires a hyponym as

[45] some low frequent patterns like "emesis <is a technical term for> vomiting" may, however, be unidirectional

Table 16: Query templates, training pairs and corpus sizes per relation type

relation type	#query templates	#training pairs	#snippets (#tokens)
induces	1	40	4,054 (94,000)
may_prevent	1	40	3,993 (91,000)
synonymy	2	20 x 2 = 40	2,444 (56,000)
ISA	4	40 x 4 = 160	9,519 (205,000)

a first argument and its hypernym as its second argument. Additionally, this particular pattern requires the second argument to be a noun in the plural.

As for the actual processing of the text snippets these are first term annotated by the script in appendix 8.8 so that the actual terms are substituted by <t1> and <t2> tags. In the case of "induces" and "may_prevent" the snippets are then part-of-speech tagged using the IMS TreeTagger[46] and chunked using Yamcha[47] before the actual KP extraction takes place. The script in appendix 8.9 has the details. For the synonymy and ISA snippets, KPs are extracted from the middle contexts using a simple regular expression.

Table 16 summarizes the corpus sizes and the number of training pairs and query templates employed for each of the four relation types. Data sparseness is the explanation why less than the target 100 text snippets per term pair are compiled for synonymy and the ISA relation. One reason is that the ontological fragment used to randomly select training pairs for the ISA relation (see figure 11) is a little too specialized. Another reason is that Google sometimes returns only 60 or 70 snippets due to duplicate content, even if a hit count of 100 was orginally reported. However, this sparseness affected the 160 pairs in a balanced manner, so there should be no bias towards one specific template or specific types of pairs.

Using linguistic information or not The query templates illustrate how linguistic information may be allowed to play a role in the identification of knowledge patterns. First of all, one may specify the linguistic form of the patterns by requiring that they contain a verb (this solution can be attractive for reasons explained below). Secondly, the linguistic form of the arguments may correlate with the form of the pattern, for example due to noun-verb agreement in number. The latter case is particularly relevant for the ISA relation. Allowing for positional and morphological flexibility, a total of four[48] query templates are used to extract knowledge pattern candidates for the ISA relation. For the causal relations the query template specifies that patterns should be continuous sequences of VP elements ($vp+) flanked by the term pair (<t1> and <t2>).

The rationale for forcing a verb in the middle context is provided by studies like [Girju and Moldovan, 2002], [Barrière, 2001] and [Christensen, 2002, Christensen, 2005]. [Girju and Moldovan, 2002] use <NP1 verb NP2> search templates to automatically

[46] www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[47] http://chasen.org/~taku/software/yamcha/

[48] to simplify the experiment morphological flexibility is not allowed for the hyponym. The names of drugs, however, will typically not occur in plural, so the limitation does not affect the performance of the system in this particular setting.

identify explicit intra-sentential syntactic patterns which instantiate simple causative relations in free text. [Barrière, 2001] presents a detailed manual study of causality patterns which reveals verbs to be the second most frequent category instantiating these patterns, and more importantly verbs to be far-and-away the most efficient (in terms of precision) of four major word classes. Finally, [Christensen, 2002] has compiled a catalogue of Danish verbs which has proven to be effective knowledge probes in the detection of terms.

While the advantage of implementing a linguistic filter is that also the less frequent, but precise, patterns are not ignored, there are two obvious disadvantages. First of all, a linguistic filter makes the system language dependent. Although accurate taggers and chunkers are available for most major languages, language independent applications driven purely by statistics are currently in vogue. Secondly, when enforcing a linguistic filter specifying specific morpho-syntactic KP structures one runs the risk of being too restrictive. To avoid this pitfall and ensure structural flexibility, the VP can maximally be preceeded by any two tokens ($dummy{0,2}) and optionally be followed by a PP element ($pp?). Finally, by allowing any number of optional NP elements ($np*) immediately before the rightmost term, premodifiers like "*very severe* stomach pains", for example, will not reduce recall. Similarly, the optional dummy tokens make sure that constructions with adverbs, complementizers, parentheses and other punctuation are not ignored.

Forcing a verb in the middle context for synonymy and ISA relations seems too restrictive (for example the patterns "i.e." or "such as" contain no verbs), so for these relations any number of tokens of any kind are allowed between the term pairs.

4.1.6 Example patterns

Before discussing how imprecise KP candidates can be automatically filtered out, one might ask to what extent noise really poses a problem at this stage. In other words it will be interesting to see examples of pattern candidates as ranked by simple frequency of occurrence in the text snippets (the column, "f", in the tables). Such examples of the top ten most frequent patterns extracted for synonymy, "induces" and "may_prevent" are listed in table 17, and the ten most frequent patterns for the four ISA templates are given in table 18.

For all four relation types there are unfiltered patterns which intuitively seem reliable, for example "prevents", "can cause", "such as" and "i.e.". However, the noise in both table 17 and table 18 is almost overwhelming. For example, "ann" and "j" would intuitively not be good synonymy markers. The prepositions "for", "on" and "in" are unlikely to be reliable markers of "may_prevent" instances, "anti" and "intolerance" are probably equally imprecise patterns for extracting "induces" instances and the table of candidate ISA markers contain a number of drugs, for example. The patterns, "k" and "d" are indeed a bit puzzling, but they can be explained by the fact that many of the training pairs for this particular ISA template involved the hypernym "vitamin". When searching on Google for "vitamin * retinol", for example, it is thus very likely that "k" and "d" suceed "vitamin" and form co-hyponyms of "retinol" rather than establish any hypernymic links.

Table 19 lists the top 10 pattern candidates for the two causal relations when using

Table 17: Top 10 unfiltered patterns by frequency of occurrence in snippets

f	synonymy	f	may_prevent	f	induces
122	or	73	for	203	induced
58	acute	71	and	45	causes
22	chronic	60	resistant	42	induces
16	recurrent	49	prevents	34	anti
16	and	36	on	32	intolerance
14	i.e.	36	in	29	can cause
13	severe	35	for acute, severe	29	are fever
13	ann	33	injection	25	non
12	j	32	cream for	22	to induce
11	called	24	prevents febrile	21	and

Table 18: Top 10 unfiltered ISA patterns by frequency of occurrence in snippets

f	hyper_sing;hypo	f	hyper_plur;hypo
89	drug	79	such as
68	k	52	e.g.
58	effect of	24	including
42	drugs	21	and
41	d	20	eg
37	agent	19	phenobarbital
31	sodium	19	especially
29	effects of	18	carbamazepine
23	cymbalta	17	phenytoin
20	properties of	14	include
...	...	...	...
f	hypo;hyper_sing	f	hypo;hyper_plur
123	an	101	and other
48	is an	34	tricyclic
34	as an	28	or other
29	has no	27	other
24	as a novel	26	as
24	has	21	and tricyclic
14	pharmacokinetics and	16	and
13	as adjunctive	10	as adjunctive
11	as	9	or

Table 19: Top 10 “induces” and “may_prevent” patterns containing a verb

may_prevent	induces
prevents	induces
reduces	does not cause
to prevent	can cause
prevent	to induce
in preventing	induced
had	include
prevented	to cause
decreases	causes
to treat	produces
reduced	may cause

the search template from subsection 4.1.5, i.e. when requiring that the patterns contain a verb. As hypothesized in section 1.4 enforcing this very simple formal restriction indeed has both positive and conspicuous effects on precision. Virtually all pattern candidates for the two relation types now appear to be highly reliable even when just ranking them by their frequency of occurrence in the text snippets.

In conclusion, it would appear that almost no further filtering is necessary when forcing a verb in the pattern search template for the causal relations, whereas further filtering is certainly required to eliminate noise when using completely unrestrictive search templates (ISA and synonymy). How such a filtering can be executed is the topic of section 4.2.

4.2 KP filtering

This section introduces two additional techniques (beyond forcing a verb) which can be used to filter out noisy KP candidates automatically. The first relies on the use of a set of negative term pairs instantiating non-target relation types. This technique is relatively common in the literature (see for example [Etzioni et al., 2004, Popescu et al., 2004]). The second technique is a novel idea of using a byproduct of the ten-fold-validation tests, namely the ten sets of positive term pairs, to measure the “iteration range” of each KP candidate (see subsection 4.2.5 for details). The iteration range of a pattern is a number between 1 and 10, and this measure is, by the author’s knowledge, unique to this thesis. Finally, although the expression “precision” will be used in what follows, this should not be understood as the actual precision of the KPs, but only as a very crude approximation. Computing actual precision scores requires a manual evaluation of the performance of each KP. The analysis of such an evaluation is provided in section 6.3.

[Pantel and Pennacchiotti, 2006] distinguish between generic and reliable patterns, ie. broad coverage noisy patterns[49] and highly precise patterns with low recall[50]. Reliable patterns are those which, on average, occur much more often with valid term

[49] for example, “the door <of> the car” for meronymy

[50] for example, “the door <is part of> the car” for meronymy

pairs than invalid term pairs. When using the entire WWW as text source optimizing precision is more important than optimizing recall, and thus the following experiments focus on identifying reliable patterns.

The basic approach mirrors that of [Alfonseca et al., 2006b, Alfonseca et al., 2006a] described in subsection 3.4.7, but it does not consider the complete context, $C_1xC_2yC_3$, of the related elements, x and y, but only the middle context, t_1KPt_2, in which t_1 and t_2 are two terms which instantiate the target semantic relation, and KP is a candidate knowledge pattern for this relation type. In other words, the precision (or reliability) of each candidate knowledge pattern, KP, is approximated as follows.

$$prec(KP) \approx \frac{\sum_{t_1;t_2 \in R} C_{Google}(t_1 KP t_2)}{\sum_{t_1;t_2 \in R} C_{Google}(t_1 KP t_2) + \sum_{t_1;t_2 \in \neg R} C_{Google}(t_1 KP t_2)}$$

The approximated precision of a KP is thus the combined Google co-occurrence frequency of the KP with a set of term pairs instantiating the target relation, R, divided by the combined co-occurrence frequency of the KP with a set of pairs instantiating both target, R, and non-target relations, $\neg R$.

To filter out low precision KPs for the four selected relations (ISA, "induces", "may_prevent" and synonymy), the four sets of term pairs from subsection 4.1.3 can be used as positives. The selection of negative term pairs, however, is the topic of subsection 4.2.1.

4.2.1 Selecting non-target relations

The simplest and most objective way of selecting term pairs instantiating *non-target* relations would be to extract term pairs at random from all of the 53 non-target UMLS relation types. However, the UMLS Metatheasurus is a huge database and also contains a great number of very specialized relation types, for example "scale_of" or "mechanism_of_action_of". Term pairs instantiating such relations cannot be expected to occur often enough on the Internet to provide reliable precision measurements. Using pairs instantiating such obscure relations, it will remain unknown whether a Google co-occurrence frequency of zero means that the KP has a high precision, or whether the negative pair is simply too rare to ever co-occur with the KP as a natural language string outside the database. One way of overcoming this problem is to require that the co-occurrence frequency of the negative term pairs exceed a certain threshold (see below).

Another completely randomized approach to the selection of negative term pairs would be to scramble the UMLS term pairs of the target relation type as described in [Mukherjea and Sahay, 2006] so that scrambled and non-scrambled "induces" instances, for example, are used in the precision measurements for this relation. There are two reasons why this technique may be suboptimal. Firstly, the technique is likely to produce many negatives which will never co-occur. For example, given the two positive instances "alcohol <induces> unconsciousness" and "vitamin c <induces> diarrhea", the technique would produce the negative pair "vitamin c;unconsciousness" which has zero co-occurrences on Google with an interpolated word wild card. Secondly, the technique is also likely to produce term pairs which, in fact, are *positive*. For

example, it may be the case that "alcohol <induces> diarrhea" is correct even if this instance is not recorded in the UMLS Metathesaurus and thus could not be automatically removed from the set of negatives.

There are additional ways in which the selection of negative pairs can have a big impact on the subsequent precision measurements. Two factors which may be important when selecting negative pairs are word order and morphology. If the form of the pattern candidates learned for a specific relation type depend on the position and grammatical number of the entities they connect, then the negative pairs should be selected so as to mirror this morphosyntactic environment. If not, it might be morphosyntactically impossible for a candidate pattern to occur with any of the negative pairs.

Finally, not all relation types are equally easy to distinguish from one another. Experience has shown that synonymy and ISA relations, for example, can be quite hard to tell apart, not just for machines but even for human judges, including domain experts. Especially for very specialized concepts with rich intensions, but narrow extensions, it can be difficult to establish whether one is indeed dealing with two distinct concepts (possibly a hypernym-hyponym pair) or synonyms of the same concept. Subsection 5.5.5 provides several examples which illustrate the proximity of the synonymy and ISA relations in Biomedicine. The semantic proximity of the synonymy and ISA relations is presumably also reflected in the knowledge patterns which instantiate the two relation types and for this reason synonym pairs are presumably very useful negatives when assessing the precision of ISA patterns and vice versa.

In summary, the following points are important to consider when selecting non-target relation types and thus negative term pairs to compute the precision of KP candidates for a relation type R.

1. Include relation types which are semantically close to R.
2. Include fundamental relation types like ISA, PART_OF and synonymy.
3. Examine whether the KP form of instances of R is morphosyntactically sensitive.
   (a) If yes, verify that the morphosyntax of the negative term pairs matches that of the positives.
4. Make sure that the negative term pairs co-occur relatively frequently in the test data (in this case on the WWW).

Out of the four relation types investigated in the thesis, only the KP form of ISA instances is morphosyntactically sensitive. For this reason the selection of negative term pairs for the ISA relation will be be dealt with separately.

Induces, may_prevent and synonymy Following the principles in the above list (points 1, 2 and 4), negative pairs for "induces" and "may_prevent" are provided by the two classical relations, ISA and PART_OF, and either of the relations themselves, which, both being causal, are semantically close. For synonymy, instances of both "induces" and "may_prevent" are selected as negatives as well as ISA instances.

More specifically, the negative pairs are selected by the following procedure.

Table 20: Negative term pairs for the "induces" relation

Relation	term1;term2	$f_{Google}(term_1 * term_2)$
ISA	lung diseases;cystic fibrosis	15,900
ISA	cystic fibrosis; lung disease	32,700
PART_OF	eyes;pupils	160,000
may_prevent	melatonin;jet lag	28,800

Table 21: Negative term pairs for the "may_prevent" relation

Relation	term1;term2	$f_{Google}(term_1 * term_2)$
ISA	lung diseases;cystic fibrosis	15,900
ISA	cystic fibrosis; lung disease	32,700
PART_OF	eyes;pupils	160,000
induces	niacin;flushing	19,400

1. For each non-target relation type pick the most frequent semantic type occurring as argument.
2. Extract all concepts (or term variants in the case of synonymy) associated with these semantic types from the target ontology (in this case the UMLS).
3. Produce all term variants of these concepts.
   (a) Optionally accept only preferred names (PN) to reduce the set.
4. Query Google to obtain term pair co-occurrence statistics.
5. Select X random negative pairs co-occurring more than Y times on Google and representing as many semantically close or fundamental non-target relations as possible.

Steps 1 and 2 may be skipped if the target ontology lacks a top ontology with information on semantic types.

For the ISA relation the most prominent semantic type in the UMLS Metathesaurus is "Diseases", and for the PART_OF relation the most prominent type is "Body Part". X is set to 4 so as to match the number of positive pairs in each of the ten-fold iterations. Finally, Y is set to 15,000, which is an aribitrary threshold value established

Table 22: Negative term pairs for the synonymy relation

Relation	term1;term2	$f_{Google}(term_1 * term_2)$
ISA	lung diseases;cystic fibrosis	15,900
ISA	cystic fibrosis; lung disease	32,700
may_prevent	melatonin;jet lag	28,800
induces	niacin;flushing	19,400

Table 23: Non-ISA pairs

relation	ISA template	term1;term2	$f_{Google}(term_1 * term_2)$
PART_OF	singular-singular	tongue;mouth	725,000
PART_OF	singular-singular	mouth;tongue	831,000
induces	singular-singular	disease;symptom	278,000
synonymy	singular-singular	illness;disease	1,010,000
induces	plural-singular	drugs;disease	681,000
synonymy	plural-singular	diseases;illness	123,000
PART_OF	plural-singular	fingers;hand	1,110,000
synonymy	plural-singular	illnesses;disease	270,000
PART_OF	singular-plural	hand;fingers	816,000
induces	singular-plural	drug;diseases	536,000
induces	singular-plural	disease;symptoms	976,000
synonymy	singular-plural	illness;diseases	144,000

empirically. While examples of the positive training pairs can be found in appendices 8.27 ("induces") and 8.28 ("may_prevent") and 8.29 (synonymy), the four negative pairs used to approximate pattern precision for each of the three relations are given in tables 20, 21 and 22.

ISA As argued above negative pairs for the ISA relation need to be selected so as to match the morphosyntactic environment of the templates used to discover the knowledge patterns in the first place (see subsection 4.1.5). For this reason the selection procedure deviates slightly from that of synonymy and the two causal relation types. However, with exception of ensuring that the negative pairs match the templates in terms of position and number, most steps are the same. As before two classical non-target relation types (PART_OF and synonymy) along with the two causal relations provide the negative term pairs. To simplify the selection process the negative term pairs are simply ranked by their Google co-occurrence frequency, and the four most frequent pairs matching the target template are selected. These pairs are listed in table 23. The complete list of positive term pairs for the ISA relation is given in appendix 8.30.

Conclusion In conclusion, it must be admitted that the best approximation of the true precision of a particular KP supposedly instantiating a particular relation type will be achieved by using a much wider range of negative term pairs than were selected in this subsection. Unfortunately, as can be seen from table 24, the number of queries already run high with just four negative pairs per template per relation type. Using a wider range of negative pairs on the WWW to get a more accurate approximation of KP precision would thus, in practical terms, necessitate a frequency threshold on the

Table 24: Querying Google

Relation	#KP candidates (ave.)	#total queries (app.)
ISA (template a)	404	3,232
ISA (template b)	203	1,624
ISA (template c)	363	2,904
ISA (template d)	316	2,528
synonymy (both templates)	459	3,672
induces	333	2,664
may_prevent	380	3,040
TOTAL	2,458	19,664

number of KPs investigated so as to limit the number of search engine queries. Since the purpose at this point is not to measure the exact precision of the KP candidates, but to enforce a crude filter which will eliminate the most noisy ones, it seems justifiable to use only a few negative pairs, but to make sure that these pairs meet the four recommendation points outlined above.

4.2.2 Query flexibility

A methodological question when querying Google to obtain frequency counts for the positive and negative term pairs is whether to look for exact phrases or whether to allow a certain degree of contextual flexibility in-between the KP candidate and the terms. Contextual flexibility can be allowed by using the same word wild card ("*") used when discovering the KP candidates in the first place. A simple example reveals the difference flexibility can make. While the query "cystic fibrosis <is a> lung disease" yields 16 hits, the flexible query "cystic fibrosis <is a> * lung disease" yields 60 hits. The disadvantage of allowing query flexibility is that it necessarily introduces a lot of noise, because the number of words allowed by "*" cannot be specified, but may range from one to about eight. Also, introducing query flexibility would require eight times as many queries to be sent to Google, because the "*" can be positioned to the left, right, both or no sides of the KP in both the negative and positive sets of queries.

All in all, it was decided to look only for exact phrases because allowing query flexibility would require a substantial amount of time and would introduce a host of new variables while not necessarily increasing the accuracy of the precision measurements.

4.2.3 Precision of all, unfiltered patterns

As a first experiment it will be interesting to see whether simply applying the complete, unfiltered[51] list of patterns discovered for "induces" and "may_prevent" can be used to distinguish target from non-target relation instances (when the patterns are simply ranked by frequency of occurrence in the training snippets). The positive pairs used in this test are extracted randomly from the UMLS (making sure that none of the pairs were used to discover KPs). They are listed in table 25. The negative pairs were also

[51] i.e. only filtered by forcing a verb as specified in subsection 4.1.5

Figure 12: “induces”: Average precision of unfiltered KPs

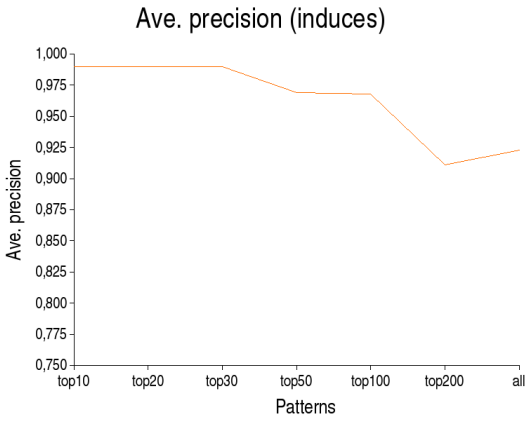


Figure 13: “may_prevent”: Average precision of unfiltered KPs

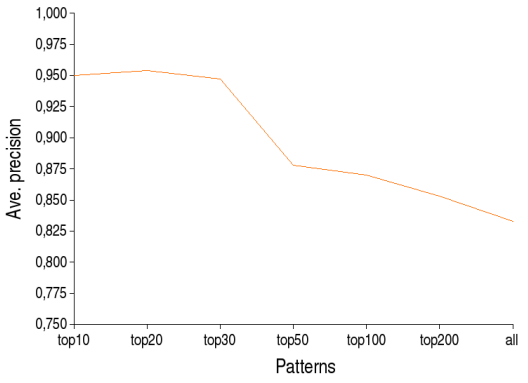


Table 25: Positive pairs

induces	may_prevent
alcohol;vomiting	prilocaine;pain
plague;headache	metoclopramide;nausea
candida albicans;allergy	psyllium;constipation
alcohol;unconsciusness	calcium;magnesium deficiency

Table 26: Negative pairs

Relation	term pair
ISA1	proteins;aprotinin
ISA2	ketones;acetone
PART_OF	small intestine;duodenum
may_prevent	feverfew;migraine
induces	oxytocin;preterm labor

extracted randomly from the UMLS, but so that two pairs represent the ISA relation, one pair represents the PART_OF relation and the last pair represents the opposite causal relation (i.e. "induces" in the case of "may_prevent" and vice versa). These pairs can be seen in table 26.

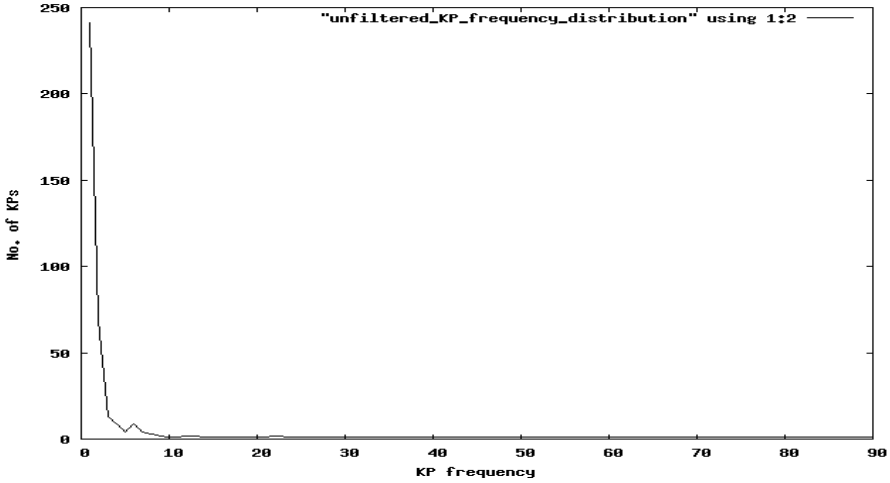
By using the standard 10-fold validation technique known from Machine Learning precision scores are computed as averages over ten iterations. Although the approach is inspired from Machine Learning, no true learning takes place because the term contexts in the *training* snippets are not annotated as either true or false, but are all considered as positive (albeit noisy). Precision is computed by inserting Google hit counts into the formula displayed at the beginning of this section, but as the co-occurrence frequencies of the randomly selected term pairs (both positives and negatives) vary, these hit counts are normalized as follows.

$$score(iteration) = \sum_{i=1}^{4} \frac{\sum_{KP \in LIST} C_{Google}(pair_i, KP)}{C_{Google}(pair_i)}$$

That is, in each of the ten iterations the total number of Google hits for all combinations of a specific term pair (1 out of 4) with the patterns in the list is divided by the co-occurrence frequency of this term pair (as measured using the Google word wild card query: "term1 * term2"). Figures 12 and 13 show the average precision of the complete list of discovered patterns for "induces" and "may_prevent" and when taking only the top 10, 20, 30, 50, 100, 200 most frequent patterns. Not surprisingly, it is a clear tendency that precision drops as less and less frequent patterns are used when testing. Interestingly, the drop in precision is not very pronounced, as it stays in the 85% to 99% range. For "induces" there is even a slight *increase* in precision from top 200 to all patterns. This seems to corroborate the claim that no further KP filtering is really necessary for the two causal relations when forcing a verb.

Figure 14 reveals how the pattern frequencies follow a typical Zipfian distribution,

Figure 14: Frequency distribution of unfiltered pattern candidates (“induces”)



in that 241 out of the 367 different patterns discovered for the “induces” relation occur only a single time in the corpus of snippets. Combining the information in figures 12 and 14 we can conclude that the majority of these low-frequency patterns must have a very high precision (although in some cases presumably a low recall). It may also be concluded that even if the number of Google queries could be more than halved simply by disregarding such KP singletons, this would entail the loss of many useful patterns. The patterns “will actually cause”, “which promotes”, “to bring about” and “triggering” are but a few examples.

4.2.4 Individual pattern precision

Using the sets of positive and negative term pairs introduced in subsections 4.1.3 and 4.2.1, respectively, the average precision of the individual patterns and their average total frequency can now be computed by inserting Google hit counts into the formula displayed at the beginning of this section. Again, precision scores are computed as averages over ten iterations making sure that the term pairs used to *discover* the patterns in each iteration are not also used when *testing* these patterns to assess their precision.

As for the actual implementation, the script used to divide the corpus of term pair contexts into ten-fold validation sets is replicated in appendix 8.8. The script in appendix 8.10 executes the formation of Google queries, and appendix 8.12 holds the implementation of the average precision scoring. Finally, appendices 8.31 and 8.32 contain the complete lists of the average frequencies and average precision scores of all pattern candidates for the two relations “induces” and “may_prevent”, respectively. Appendix 8.33 contains the results of automatic precision ranking of the candidate ISA patterns, and appendix 8.34 the results for synonymy. Examples from these complete lists are given in tables 27, 28, 29 and 30, respectively.

Judging from the results in the appendices and the examples in the tables, the crude

Table 27: Induces pattern candidates (examples)

ave. prec(KP)	ave. f(KP)	KP
100%	163	may cause
100%	150	to induce
100%	86	produce
100%	15	does not cause
99.7%	1514	induced
87.4%	57	causes
78.6%	82	can cause
46.9%	104	is
43.1%	68	include
29.9%	11	are

Table 28: May_prevent pattern candidates (examples)

ave. prec(KP)	ave. f(KP)	KP
100%	444	to prevent
100%	326	for relieving
100%	308	for preventing
100%	281	helps prevent
100%	261	relieves
100%	185	in preventing
60.9%	374	reduces
53.1%	35	can reduce
22.2%	195	causes
8.3%	364	is
3.2%	28	are

Table 29: Synonymy pattern candidates (examples)

ave. prec(KP)	ave. f(KP)	KP
100%	188.1	also called
100%	72.1	means
100%	68.1	see
100%	43.2	also known as
100%	0.6	causing
100%	0.2	often called
99.81%	5899.5	or
18.55%	521.8	is
3.28%	102.3	induced
0.28%	105.2	include
0.00%	7.2	which causes

Table 30: ISA pattern candidates (examples)

ave. prec(KP)	ave. f(KP)	KP
99.96%	184.2	an
89.57%	53.4	has
88.59%	28.7	activity
79.83%	22.6	is a new
79.52%	82.3	such as
76.72%	585.7	is
40.52%	134.4	are
19.63%	906.2	with
9.77%	1717.1	on
7.34%	604633.7	and
5.91%	19.2	but
0.00%	9.7	both

precision filter does seem to be working. For example, prepositions like "on" and conjunctions like "and" get very low precision scores for the ISA relation in table 30. However, the relatively high precision scores for patterns like "is", "are" and "include" for the "induces" relation in table 27 strike the eye. These cases can be explained by the fact that many knowledge patterns are, in fact, discontinuous templates extending beyond the middle context of term pairs. In this case the template is presumably:

"(a)? side effect(s)? of <cause> (include|are|is) <effect>"

Interestingly, the negated pattern, "does not cause", gets a precision score of 100% in table 27, presumably because it does not occur with any of the four negative term pairs. However, it could be the result of dosage information given in the left context, for example "small doses of X <does not cause> Y". Worse yet, completely misleading patterns like "causing" for synonymy (see table 29) still make it through the crude filter. Subsection 4.2.5 will introduce a way of eliminating this kind of residual noise.

4.2.5 Using iteration range to eliminate noisy KPs

As was indicated by the lists of unfiltered KPs and precision scored KPs in subsections 4.1.6 and 4.2.4, a disadvantage of using a totally unrestrictive search template (as is the case for ISA and synonymy) is that a number of noisy patterns are found. A simple but effective strategy for eliminating residual noisy patterns is to examine the "iteration range" of all KP candidates. This is to be understood as the number of iterations (a number from one to ten) in which a particular candidate occurs during the ten-fold-validation process.

Tables 31 and 32 illustrate the effects of applying a iteration range filter to KPs discovered for the four ISA templates and synonymy, respectively. Looking at the patterns in the two tables, it seems that adding this filter has eliminated virtually all noise. Patterns like "drugs such as" are clearly domain dependent, but a closer analysis of

Table 31: ISA KPs filtered by iteration range, average sample frequency and precision

pattern	precision	range	pattern	precision	range
ISA-b:			ISA-c		
e.g.	99.98%	10	exerts its	100.00%	9
such as	99.88%	10	as an	100.00%	9
including	99.16%	10	is an	100.00%	10
like	89.60%	10	is an effective	100.00%	10
i.e.	77.15%	9	an	99.96%	10
include	69.02%	10	has	89.57%	10
			is a new	79.83%	10
ISA-a:			is	76.72%	10
efficacy of	100.00%	9	a new	69.65%	10
action of	100.00%	9	as	63.94%	10
drugs	100.00%	9	has an	59.99%	10
actions of	100.00%	8	another	57.58%	10
agents	100.00%	7	and other	55.96%	10
agents such as	100.00%	7			
called	100.00%	7	ISA-d:		
drugs such as	100.00%	7	and other	99.01%	10
properties of	100.00%	7	or other	69.48%	10
effects of	100.00%	10	other	68.38%	10
effect of	100.00%	10	with other	67.60%	10
activity of	100.00%	10	see	59.15%	10
drug	99.94%	10	as	58.84%	10
activity	88.59%	10			
such as	79.52%	10			

Table 32: synonymy KPs filtered by iteration range, average sample frequency and precision

pattern	precision	range
or	99.13%	10
see	100.00%	9
also known as	100.00%	9
ie	100.00%	9
means	100.00%	8
also called	100.00%	8
acute	100.00%	7
called	100.00%	6
aka	100.00%	6
is also called	100.00%	5
mild	100.00%	4
is known as	100.00%	4
refers to	100.00%	4
severe	100.00%	4
was defined as	100.00%	4

Table 33: Number of filtered KPs used in system evaluation

type of filter	induces	may_prevent	ISA	synonymy
(1) "iteration range"	no	no	yes	yes
(2) ave. frequency	no	no	yes	yes
(3) linguistic	yes	yes	no	no
(4) ave. precision	yes	yes	yes	yes
#unfiltered KPs	367	380	1,286	459
#filtered KPs	71	101	41	13

KP domain dependence is deferred to the empirical experiment in section 6.5. Finally, while the three adjectives, "mild", "severe" and "acute", in table 32 may be useful KPs for retrieving synonyms of diseases (cf. "... pruritus (severe itching) ..."), they appear less useful for retrieving synonyms of substances like those tested in the system evaluation (see subsection 5.2.4). Based on this observation one may ask to what extent KPs can be not only domain dependent, but also dependent on the semantic types of the arguments instantiating the target relation.

Table 33 summarizes the effects of the combined filtering on the number of KPs for the four relation types. For all relation types a mimimum average precision threshold of 50% was enforced. However, based on the findings in subsection 4.2.3, the average frequency and "iteration range" filters were only deemed necessary for the ISA and synonymy relations. For both relations the minimum average frequency threshold was set to 1, and the minimum iteration range was set to 7 for the ISA KPs, but for

synonymy it was lowered to 4 so as to allow a greater number of patterns to be investigated (see appendix 8.34). These threshold values were empirically established, and the impact of the individual filters and threshold values on system performance certainly merit further investigation in future work.

4.2.6 Conclusion

The experiments in this section have shown that using text snippets on the WWW appears to be an effective strategy not just for KP discovery, but also for KP filtering. In summary, section 4.1 described the WWW-based discovery of KPs instantiating four different relation types. For "induces" and "may_prevent" a search template forcing a verb appeared to reduce noise so that the patterns could, in fact, be used with no further filtering as a high-precision relationship recognizer (see subsection 4.2.3). For synonymy and ISA, totally unrestrictive search templates were used, and predictably this resulted in much more noisy patterns.

However, it was demonstrated how this residual noise can be minimized by computing approximate KP precision scores on the WWW based on a principled selection of positive and negative term pairs (subsection 4.2.4) and by measuring a so-called "iteration range" (subsection 4.2.5) based on the number of iterations (out of ten) in which each KP candidate is found.

While chapter 4 illustrated and discussed how KPs can be automatically discovered and filtered from text snippets on the WWW, chapter 5 describes the implementation and thorough evaluation of an actual relation extraction system based on these sets of filtered KPs.

5 Relation instance extraction

This chapter describes the implementation and evaluation of WWW2REL, an automatic relation extraction system which operates exclusively on WWW text snippets and is equipped with the four sets of KPs discovered and filtered in chapter 4. While the system implementation is the topic of section 5.1, system evaluation issues are discussed in section 5.2. WWW2REL features a range of instance ranking schemes which are devised in section 5.3 and comprehensively evaluated against a manually established gold standard (sections 5.4 and 5.5).

5.1 System implementation

This section provides a brief overview of the technicalities and the challenges of the system implementation, including its initialization and test phase. It also discusses certain system strengths and limitations affecting the output which four domain experts are subsequently asked to evaluate. This manual evaluation in turn provides the foundation for all subsequent analyses of the performance of various ranking schemes (section 5.4) as well as the performance of individual knowledge patterns (section 6.3).

The figure in appendix 8.35 displays a schema of the database used to store all information about candidates, patterns and the mappings between them (the table called

"np2kp"). All graphs displayed in sections 5.4 and 5.5 are produced by querying this database. Consisting of only three different tables, the database has a very simple structure. The table, "patterns", is a static table containing all the filtered patterns obtained for each of the four relation types. The table "candidates" is activated during the system test and contains all NPs returned when querying Google using templates like "<input term> {KP} ?" and "? {KP} <input term>", where {KP} is the set of knowledge patterns discovered for the target relation type. Finally, the table "np2kp" holds information on each and every <term-KP-candidate> triplet occurring in the text snippets retrieved for all experiments. In other words, it maps every candidate to the range of KPs with which it co-occurs.

5.1.1 Discovering and filtering KPs

Initializing the system essentially means discovering and filtering KPs for the target relation type(s). The actual steps of the implementation presented in this thesis are as follows.

1. Discover KPs
   (a) Select random seed term pairs from the target ontology (appendices 8.4, 8.5, 8.6 and 8.14)
   (b) Keep only pairs with a co-occurrence frequency exceeding an appropriate threshold (appendix 8.11)
   (c) Download text snippets containing these pairs from Google (appendix 8.7)
   (d) Term tag snippets and prepare 10-fold-validation sets (appendix 8.8)
   (e) Optionally POS tag and chunk snippets (appendix 8.9)
   (f) Extract pattern candidates from term pair contexts (appendix 8.9)
2. Filter KPs
   (a) Combine the term pairs from the 10-fold-validation sets with KP candidates to form search engine queries (appendix 8.10)
   (b) Retrieve hit counts from Google for positive and negative pairs (appendix 8.11)
   (c) Normalize hit counts and compute crude pattern "precision" (appendix 8.12)
   (d) Measure "iteration range" (for synonymy and ISA) (see subsection 4.2.5)
3. Store KPs in database

Table 34: Automatic NP conflation

original NP	transformed NP
bleeding in the stomach	stomach bleeding
a buildup of toxins	toxin buildup
increased risk of haemorrhage	*haemorrhage increased risk
stomach ulcers in some people	*people stomach ulcers

5.1.2 Discovering relation instances

Having discovered a set of filtered patterns for a particular relation type, it is time to implement a simple extraction system which can search the WWW to discover, for example, possible side effects given a certain drug as input.

The system implementation has the following steps.

1. Read input term and target relation type[52] from user
2. Form "<input_term> <KP> [?]" queries for each of the filtered KPs for the target relation type
3. Retrieve top X snippets for each of these queries (appendix 8.15)
4. Remove markup, tag and chunk snippets (appendix 8.16)
5. Extract NPs in position [?], delete determiners/prepositions and attempt PP fronting (appendix 8.17)
6. For each NP compute instance reliability, r(i), using one of several ranking schemes (see section 5.3)
   (a) Optionally ignore hapax legomena (singletons)
   (b) Optionally apply BNC discounting heuristic (subsection 5.3.1)
   (c) Optionally group NPs by their heads (subsection 5.3.2)
7. Output top X most relevant candidates as ranked by scheme Y

In light of the discussion in subsection 2.2.2 about the linguistic properties of academic writing, it was decided to split NPs postmodified by a PP, delete the preposition and determiners and attempt PP fronting. This way variant NPs referrring to the same concept can be conflated, system output can be generalized and recall should be boosted. This technique will result in transformations like the examples in table 34.

As witnessed by the two examples marked by "*" in table 34, however, the simplistic transformation technique may also produce ungrammatical NPs. While these errors could be minimized using a more sophisticated tranformation technique, it was deemed sufficient to simply accept only those transformed NPs which have at least one untransformed counterpart in the given data. In other words, if the transformation

[52] so far, the system has only been initialized for 4 relation types

"*people stomach ulcers" does not occur as an untransformed NP, the system ignores the PP ("in some people") and registers only "stomach ulcers". Using this heuristic a compromise is made between losing lots of information (ignoring all PP postmodifiers) and flooding the user with NP variants which essentially instantiate the same concept (keeping all PP postmodifiers as is).

However, the current system implementation has the following NLP limitations.

1. lack of anaphora resolution (outside the scope of the thesis and of text mining)
2. only the first PP is analyzed and fronted (if possible)
3. NPs with conjunctions are not decomposed
4. input term modifiers are ignored
5. no lemmatization of candidates

As for the second limitation, full parsing of the text snippets would presumably allow the extraction of "stomach bleeding" from a phrase like "aspirin [may cause] patients to experience [stomach bleeding]". In these cases only the NP head immediately following the KP (i.e. "patients") will be extracted even if this is semantically vague and terminologically irrelevant. It is doubtful, however, whether full parsing will be worth the effort or even be possible when processing sentence fragments. Also, the ranking heuristics devised in section 5.3 should ensure that semantically vague heads like "patients" be penalized.

The third limitation may have had the effect of lowering the overall recall of the system. If a "glucose" synonym candidate like "dextrose or corn sugar" (which is very infrequent but nevertheless correct) were to be split into its constiuent parts, system performance would presumably be boosted.

The fourth limitation can explain why candidates like "blood sugar" and "hyperglycemia" are suggested by the system as synonyms of "glucose" in subsection 5.5.5 (table 54). It is simply because the input term premodifiers marked by angle brackets in the following sentence fragments are allowed.

1. ... <blood> [glucose] also known as *blood sugar* ...
2. ... <high blood> [glucose], or *hyperglycemia*, ...

While it is correct that "high blood glucose" and "hyperglycemia" are synonyms, "glucose" and "hyperglycemia" are not. Future versions of the system should explore whether disallowing input term modifiers will result in a performance gain or may cause data sparseness situations.

Finally, there are two reasons for not including a lemmatization module in the system implementation. First of all, it would make the system language dependent. Secondly, many of the candidates (for example antipsychotic drugs) are so specialized that they will not be recorded in any freely available, machine-readable lexicons and thus will be left in their plural form by the lemmatization module anyway. Thirdly, adding biomedical NER modules would, as discussed in section 3.4, reduce system portability.

5.2 System evaluation

This section contains a description of the manual evaluation setup for the relation extraction system which was outlined in section 5.1, initialized in sections 4.1 and 4.2 and which will be tested in sections 5.4 and 5.5. The section will both discuss issues particular to the evaluation of WWW2REL, but also challenges which are relevant when evaluating AKA systems in general.

5.2.1 Evaluation setup

> Evaluation of ontology learning is an important but largely unsolved issue, as reported at the workshop [5] upon which this volume is based. Two evaluation stages are typically performed when evaluating an ontology learning method. First, *term level evaluation* assesses the performance of extracting domain relevant terms from the corpus. Second, an *ontology quality evaluation* stage assesses the quality of the extracted ontology. While term level evaluation can be performed by using the well-established recall/precision metrics, ontology quality evaluation is more subtle and there is no standard method for performing it. One approach is to compare an automatically extracted ontology with a Gold Standard ontology which is a manually built ontology of the same domain [...] Another approach is to evaluate the appropriateness of an ontology for a certain task. [Sabou, 2005, p131]

While it is perhaps debatable whether applying precision/recall metrics to the task of term extraction is not equally problematic as applying them to the task of ontology learning, the above quote does highlight a problem which is highly relevant to the evaluation which is to follow. The task of automatically extracting semantic relation instances from free text and evaluating their usefulness in enriching an existing ontology is an intermediate step in-between the two steps of "term level evaluation" and "ontology quality evaluation" mentioned in the quote. However, since semantic relations are essentially the building blocks of ontologies, observations about the evaluation issue in the literature on ontology learning will also be pertinent to the relation extraction task. In short, the main problem is the lack of a standard methodology and framework for evaluating and comparing applications which extract semantic relations or build ontologies from text.

The performance measurements of a number of ranking schemes in section 5.4 will also be based on a Gold Standard. This standard is not an ontology but a great number of manually annotated relation instances proposed by WWW2REL given eleven different input terms (table 38) and the four sets of filtered KPs produced in section 4.2.

In effect the system was run for the eleven different experiments *without any kind of instance ranking or filtering*. In this way it produced approximately 2,000 candidate relation instances (cf. table 38) which were then given to the four domain experts who judged their correctness. Establishing a Gold Standard by means of unfiltered system output is perhaps a bit unorthodox, but given time and financial constraints it was unfeasible to ask the domain experts to produce an ontology for each experiment. Even if

the experts had had plenty of time to introspectively devise such ontologies, questions could be raised as to the completeness and appropriateness of the result. Measuring precision/recall against introspectively created ontologies would be as errorprone as measuring precision/recall automatically against the UMLS Metathesaurus.

The main reason why it is problematic to measure performance in this way is that it will completely rule out the possibility of retrieving *new knowledge* which just happens not to be recorded in the UMLS or to have materialized through domain expert introspection (see section 6.4 for examples). Since finding "new" terminological knowledge is the main purpose of the implemented system, establishing our gold standard by means of unfiltered, system output appeared to be the only viable solution. This decision is also supported by [Faatz and Steinmetz, 2005] who observe that

> From the moment we consider words or phrases which do not come from the given ontology, there is no automatic way of judging about their quality: from our point of view quality statements are only allowed for the descriptors we already met with the given concepts. [Faatz and Steinmetz, 2005, p84]

An important question now remains. How are the domain experts supposed to evaluate the (terminological) relevance of the relation instances produced by the system?

> During the concept per concept analysis of the extracted ontologies the domain experts rated concepts *correct* if they were useful for ontology building and were already included in the Gold Standard. Concepts that were relevant for the domain but not considered during manual ontology building were rated as *new*. Finally, irrelevant concepts, which could not be used, were marked as *spurious*. [Sabou, 2005, p132]

[Pantel and Ravichandran, 2004] also use three categories for judging the correctness of automatically harvested semantic relations (the ISA case). Unlike [Sabou, 2005] they distinguish between "correct", "partially correct" and "incorrect" candidates. The evaluation setup used in the following experiments also include three possible categories for judging the relevance of each relation instance extracted by the system. Unlike the experiments described in [Sabou, 2005], however, the experts in this thesis setup were not asked to assess the novelty of the extracted relations, since this can be assessed automatically by querying the UMLS knowledge sources (if it is not recorded, it is novel). Instead, the four domain experts were asked to use the three categories listed in table 35 when judging the correctness of instances extracted by WWW2REL.

The category "4" is a special case which is only used in the two experiments "X induces vomiting" and "X induces emesis" due to the greater search space of these queries. Finally, it may seem crude to group instances for which the expert is "unsure" whether the target relation holds with instances in which the argument is deemed semantically vague. This was done in order to speed up the manual evaluation process by reducing the number of possible categories. On average each expert was only allotted about 30 seconds to decide on each of the approximately 2,000 candidate instances. An unfortunate consequence of grouping the two categories is that no separate analysis of the semantically vague arguments can be carried out, although such an analysis

Table 35: Categories used in manual evaluation of relation correctness

category	meaning
1	relation is correct AND relevant
2	unsure OR argument of relation is fuzzy/vague/incomplete
3	relation is incorrect
4*	relation is correct AND argument is a drug

might have contributed to the theoretical debate outlined in section 2.2.4. Nevertheless, the main focus of the experiments is to optimize system precision for *terminologically* relevant arguments of the target relations, i.e. instances annotated as category "1".

5.2.2 Manual evaluation issues

When setting up a manual evaluation of an automatic relation extraction system there are a number of potential pitfalls, including but not limited to the following.

The human factor Even domain experts are not infallible. There may be white spots in their otherwise comprehensive knowledge about the subject area, but worse than that, there may be cases in which an expert is wrong, but does not stop to consider this possibility during the evaluation. By asking four experts to look at the same data, an inter-annotator agreement can be computed (see subsection 5.2.5), and this should reduce the overall effect of individual misjudgments. Nevertheless, the judgments of the domain experts should not always be taken as the holy grail. It is not totally inconceivable that the judgments of all four experts could be wrong in a few cases.

Simplified evaluation scale By forcing the experts to select among only three possible verdicts (1=correct, 2=vague/unsure, 3=incorrect), borderline cases can be hard to assess. However, time constraints forced the merger of the "argument is semantically vague" and "unsure" categories into one as mentioned above.

Objectivity Especially when the developer and the thesis writer is the same individual, a potential problem is that

> When the developers are deeply involved in an assessment, the readers may rightfully question the impartiality of the entire process and the outcome measures, and thus also the conclusion. Therefore, a description of stakeholders participating in an assessment study is important for the interpretation of an assessment study. [Brender, 2006, p291]

In this case the thesis writer and the developer is the same person, but to secure impartiality in the evaluation of system performance, this task was exclusively assigned to four domain experts who had no part in the system design or implementation.

Old or incorrect knowledge A potentially more problematic aspect of extracting bits of knowledge from the WWW is that this source contains millions of old documents which were created years ago and perhaps never updated. It also contains numerous documents containing incorrect information for one reason or another. Although restricting queries to particular web sites, domains or date ranges might reduce the risk of encountering incorrect or old knowledge, it is not obvious how such a delimitation of data sources could be carried out while avoiding that individual sources bias the results. It is obvious, however, that restricting queries to specific web sites will severely reduce the portability of the system to other domains and exacerbate data sparseness problems.

Another issue is that the efforts of painfully establishing a truly balanced, authoritative and up-to-date collection of URLs might be partly in vain due to the dynamic nature of the WWW where URLs shift in and out of existence over time. Finally, the distribution of such a list of URLs to other researchers can be deemed illegal (see [Sharoff, 2006] and subsection 3.1.2 for more details).

In the present case it is decided to employ totally unrestrictive Google queries, with the exception of setting the language to English. This choice is partly motivated by the data sparseness problems experienced in subsection 4.1.3, but also by the fact that old (but correct) knowledge will often be required to clarify concepts and build coherent concept systems in practical terminology work. Domain experts, on the other hand, will tend to focus on recent scientific developments rather than the long established facts which are relevant to terminologists.

As for the problem of incorrect knowledge this is a more severe challenge which may reduce the precision of the knowledge patterns unfairly. Even if someone asserts (for example in a blog) that "aspirin causes cancer" and the relation is deemed incorrect by the four experts, this need not be the fault of the KP, "causes". Thus the individual KP performance figures reported in section 6.3 are presumably underestimated when compared to the performance of most other applications for the domain of Biomedicine which typically operate only on academic papers (cf. subsection 3.4.11).

Observer bias Finally, domain experts will necessarily have slightly different educational backgrounds, and they may also have either positive or negative preconceptions about the the system to be evaluated. In the latter case "the participants may start collecting extra data to prove themselves right and the system wrong or vice versa" [Brender, 2006, p296]. In either case, the only way to minimize observer bias is to include as many human evaluators as logistically possible. In the present case that number is four evaluators. All evaluators are pharmacists educated at the Danish University of Pharmaceutical Sciences[53].

5.2.3 Precision targets

Having discussed a variety of pitfalls in manual system evaluations, it is time to ask a fundamental question concerning system performance. In short, what level of precision can be expected?

[53] www.dfuni.dk

Table 36: “may_prevent” - most frequent STY combinations

frequency	STY1	STY2
9,745	Clinical Drug	Disease or Syndrome
2,037	Clinical Drug	Pathologic Function
1,522	Clinical Drug	Sign or Symptom
996	Food	Disease or Syndrome
681	Clinical Drug	Injury or Poisoning
679	Clinical Drug	Finding
462	Organic Chemical	Disease or Syndrome
248	Drug Delivery Device	Disease or Syndrome
242	Amino Acid, Peptide, or Protein	Pathologic Function
198	Clinical Drug	Congenital Abnormality
...	...	...

> If one assumes that relationship extraction requires identification of three biomedical terms (two entities and one relationship), the performance of relationship extraction should be approximately equal to the cube of the performance of NER[54]. [...] the assumption does not seem to hold for biological relations. It may be easier to extract concepts in combination with the relationship between them owing to the increased local context that relationships provide. [Cohen and Hersh, 2005, pp60-61]

If state of the art performance in Named Entity Recognition tasks is about 90%, then relation extraction performance could be expected to achieve about 70%. However, in the case of WWW2REL, one term is required as input, so the system only has to identify one relation and one biomedical term. This raises the expected performance to 80%. On the other hand, working exclusively with noisy text on the WWW renders the task more difficult and should lower the performance expectations somewhat. As will become apparent in sections 5.4 and 5.5 this level of precision is certainly attainable in most experiments, while performance falls short of this level in other experiments where the input term is not really a term but a word (see the “vomiting” experiment in subsection 5.5.3).

5.2.4 Selecting input terms

In order to select appropriate input terms with which to test WWW2REL, it will be relevant to examine which entities typically occur as arguments of the target relations (for example “may_prevent” and “induces”) in the UMLS Metathesaurus, the most comprehensive ontology of Biomedicine. Extracting statistics directly from the UMLS we get the results listed in tables 36 and 37. The tables reveal that for both “induces” and “may_prevent” the most frequent semantic type (STY for short) combination is “Clinical Drug” and “Disease or Syndrome”. Hence, drugs, diseases and syndromes will be selected as input terms in the system tests.

[54] Named Entity Recognition

Table 37: “induces” - most frequent STY combinations

frequency	STY1	STY2
611	Clinical Drug	Disease or Syndrome
590	Clinical Drug	Pathologic Function
346	Clinical Drug	Sign or Symptom
313	Clinical Drug	Finding
211	Amino Acid, Peptide, or Protein	Pathologic Function
97	Food	Disease or Syndrome
49	Drug Delivery Device	Disease or Syndrome
45	Clinical Drug	Injury or Poisoning
41	Organic Chemical	Finding
25	Organic Chemical	Disease or Syndrome
...	...	...

Table 38: System inputs for evaluation

input term	relation	#snippets (tokens)	#candidates	min_frq
aspirin	induces	3,376 (98,000)	365	>1
selenium	may_prevent	4,967 (121,000)	421	>=1
vomiting	induces	5,110 (127,000)	317	>1
emesis	induces	1,641 (40,000)	76	>1
formaldehyde	synonymy	2,028 (48,000)	46	>2
vitamin C	synonymy	2,684 (70,000)	63	>2
lactose	synonymy	2,291 (57,000)	41	>2
glucose	synonymy	3,171 (80,000)	100	>2
progesterone	synonymy	2,631 (62,000)	61	>2
antipsychotic(s)	ISA (hyponymy)	4,270 (95,000)	225	>1
haloperidol	ISA (hypernymy)	3,940 (90,000)	141	>2
11	4	36,109 (888,000)	1,856	

As for the input terms, these were randomly selected from the UMLS Metathesaurus among relatively frequent[55] active ingredients from “induces”, “may_prevent” and synonymy relations. For the ISA relation one hypernym and one hyponym were selected. The hypernym being “antipsychotic” (cf. the ontology fragment in table 11), and the hyponym being the single most frequent term variant of an antipsychotic, namely “haloperidol”. Table 38 summarizes the input terms and relation types for which the extraction system was run and for which the domain experts were asked to evaluate output. It also summarizes the number of snippets (and tokens) compiled from the WWW in each of the 11 experiments.

The system tests and system evaluations were carried out in six sessions[56] in November and December of 2006. For each session system output was produced from the

[55] i.e. having a Google hit count exceeding 1 million hits

[56] the five synonyms constituted a single session

Table 39: Interpretations of kappa values

Kappa	Interpretation
< 0	poor agreement
0.0-0.20	slight agreement
0.21-0.40	fair agreement
0.41-0.60	moderate agreement
0.61-0.80	substantial agreement
0.81-1.00	almost perfect agreement

database and e-mailed individually to the four domain experts in the form of a spreadsheet with three columns. One column listing all the candidate relation instances for the experiment, one column listing instance ID numbers making it easy to load the judgments into the appropriate table fields in the database, and one empty column for the judgments themselves.

For each session the experts were given 2.5 hours to evaluate the output. On average, the experts were able to evaluate 300 to 400 candidates in 2.5 hours, i.e. spending about 30 seconds per instance. For this, purely pragmatic, reason the minimum sample frequency in table 38 (min_frq) varies between 1 and 3 so as to arrive at a number of candidates in this interval. In the special case of the two ISA experiments, the minimum sample frequency is to be understood as the combined frequency of a candidate across all the four ISA query templates.

Although it would have been interesting to have had several input terms per relation type, priority was given to covering all the four relation types in the first place.

5.2.5 Inter-annotator agreement

Inter-annotator agreement is assessed by using an implementaion (cf. appendix 8.18) of the Fleiss' kappa test mesure for inter-rater reliability [Fleiss, 1971]. The kappa measure, k, is given by $k = \frac{\overline{P}-\overline{P_e}}{1-\overline{P_e}}$, where the denominator indicates the degree of agreement that is attainable above chance, and the numerator indicates the degree of agreement acually observed above chance. The measure ranges from negative numbers to 1 (perfect agreement), but no universally accepted interpretation of kappa score intervals exists. Table 39 lists an interpretation from [Landis and Koch, 1977]. Even if there is no agreement on the interpretation of agreement levels in absolute terms, the kappa measure is still useful as a measure of *relative* agreement, i.e. for comparing whether there is greater agreement in one versus another experiment. Kappa measures for all system evaluations are summarized in table 40.

Generally speaking, the fewer the categories the higher the agreement and thus the kappa values. As described in subsection 5.2.1 the domain experts were asked to assign each candidate relation proposed by the system to one of three categories, "1" for correct, "2" for "unsure/too vague" and "3" for "incorrect"[57]. However, as some

[57] the additional category "4", meaning correct AND argument is a drug was used in two experiments, but is counted as "1" when computing inter-annotator agreement

Table 40: Inter-annotator agreement across all experiments

Experiment	kappa (strict)	kappa (lax)	observations
synonyms of formaldehyde	0.54	0.75	46
X <ISA> antipsychotic	0.34	0.62	225
haloperidol <ISA> X	0.36	0.57	141
synonyms of lactose	0.40	0.57	41
synonyms of glucose	0.42	0.56	100
synonyms of vitamin C	0.38	0.45	63
X <induces> emesis	0.19	0.42	76
synonyms of progesterone	0.34	0.40	61
aspirin <induces> X	0.18	0.28	365
selenium <may_prevent> X	-0.01	0.28	421
X <induces> vomiting	0.07	0.23	317

of the experts seemed a little hesitant to make use of the category "3" and seemed to prefer "2", the kappa values in table 40 are given in two columns, "strict" and "lax". "Strict" means that all four experts assign a relation to the same of the three categories for there to be perfect agreement. "Lax" means that the categories "2" and "3" have been merged, and perfect agreement is attained if the four experts all agree whether a relation is "correct" ("1") or "unsure/too vague/incorrect" ("2" or "3").

As predicted kappa values are higher when using only two categories instead of three. Regardless of the number of categories it is apparent that inter-annotator agreement in the synonymy and ISA experiments is far better than for the causal relations ("induces" and "may_prevent") and the "X induces vomitinglinduces". However, in the latter case using the more technical synonym "emesis" rather than "vomiting" does boost agreement considerably. In terms of kappa value interpretations, agreement ranges from fair to substantial (formaldehyde) when using two categories and from poor (selenium) to moderate when using three categories. With a kappa score of a mere -0.01 and 0.28, the "selenium may_prevent X" experiment appears to have caused the domain experts considerable trouble.

Table 41 lists the proportion of "2" judgments used in the individual experiments. Again, it should be stressed that the "2" judgment also covers cases where the candidate proposed by the system was considered semantically too vague to be relevant or simply did not make sense[58]. The figures are thus artificially high if interpreted exclusively as the degree of expert uncertainty in each experiment. Nevertheless, the percentages indicate that the two causal relations along with the "X induces vomitinglemesis" experiments were the hardest to evaluate. The synonymy experiments all have a conspicuously low proportion of "2" judgments. Surprisingly, it appears to have been easier for the experts to assess whether a candidate is an antipsychotic drug than whether "haloperidol" is an antipsychotic. In the case of "haloperidol ISA X" most of the "2" judgments should probably be interpreted as "too vague" (for example candidates like "medication" or "drug") rather than "unsure".

[58] for example when PP fronting fails and the semantic core of the NP happened to be the PP head

Table 41: How unsure are the experts?

experiment	#unsure/vague	%unsure/vague
vitamin c synonyms	18	7%
progesterone synonyms	20	8%
lactose synonyms	17	10%
formaldehyde synonyms	19	10%
glucose synonyms	75	19%
X ISA antipsychotic	244	27%
haloperidol ISA X	222	40%
X induces emesis	132	43%
X induces vomiting	587	46%
aspirin induces X	803	55%
selenium may_prevent X	1,001	59%

Both in terms of inter-annotator agreement and proportion of "2" judgments the "selenium may_prevent X" experiment appears to have been the most difficult one of all eleven. One explanation is the banal fact that there *is*, in fact, very little agreement in the field as such as to the correctness of various proposed beneficial effects of selenium. This fact was revealed through private talks with the four experts, and in hindsight another (or an additional) input term should probably have been selected for testing the KPs discovered for the "may_prevent" relation.

5.3 Devising instance ranking schemes

The arsenal of instance ranking schemes devised for WWW2REL and evaluated in sections 5.4 and 5.5 are listed below. It should be noted that all these ranking schemes operate exclusively on the samples of text snippets retrieved by WWW2REL and thus (unlike e.g. the Espresso system) require no further WWW querying to establish their ranking order.

1. $frq = C_{sample}(i)$
2. $kpr = KPR_{sample}(i)$
3. $fkpr = C_{sample}(i) * KPR_{sample}(i)$
4. $pkpr$: like 2) but including passive KPs in KPR
5. $pmi = \frac{\sum_{p \in P} \frac{pmi(i,p)}{max_{pmi}}}{|P|}$
6. $pmi2 = \frac{\sum_{p \in P} \frac{pmi2(i,p)}{max_{pmi2}}}{|P|}$

The first ranking scheme, named "frq", is a baseline method in which instances are simply ranked by their frequency of occurrence in the corpus of text snippets. The

second scheme ranks relation instances by their KP range, i.e. the number of *different* filtered KPs with which each instance occurs in the snippets corpus. This scheme will be referred to as "kpr". The third ranking scheme, "fkpr", is a hybrid scheme in which instance frequency is multiplied by instance KP range.

As for the fourth scheme, "pkpr", this is identical to the "kpr" scheme except that a small set of passive KPs (see the list in appendix 8.32.1) are added to the active construction KPs. This scheme is only implemented for the "induces" relation and is used to test the hypothesis that passive KPs may identify more terminologically relevant instances and improve precision because boosted use of the passive voice is one of the most characteristic and universal features of academic language (see e.g. [Biber et al., 1998, Biber et al., 1999]). The "pkpr" of an instance is computed by querying Google to see how many of the passive patterns the instance co-occurs with. This number is then added to the "kpr" of the instance. The procedure is implemented in the script in appendix 8.24.

Finally, the fifth scheme is essentially a slightly modified version of the instance reliability formula used in the Espresso system [Pantel and Pennacchiotti, 2006]. It is also based on a sum of the pointwise mutual information (pmi) scores of all KPs with each instance, but as mentioned in subsection 3.4.4, the reliability of all KPs in WWW2REL are either set to 1 (i.e. 100%) if they are accepted by the combination filter (see table 33) or to 0 if they are not accepted. In Espresso a continuous reliability scale is used.

It has been observed that "a well-known problem is that pointwise mutual information is biased towards infrequent events" [Pantel and Ravichandran, 2004]. In order to discount the significance of low-frequency pairs and compensate for this bias one often squares the observed frequencies as in the variant pmi formula given below (see e.g. [Evert, 2004]).

$$pmi2(i,p) \approx \frac{C_{google}(t,p,i)^2}{C_{google}(t,*,i) * C_{google}(*,p,*)}$$

With the regular pmi and this pmi2 variant (see appendix 8.21) the number of ranking schemes is brought up to six in total.

5.3.1 BNC discounting heuristic

The ranking schemes are essentially attempting to do two things at once, namely

1. automatic relation instance extraction
2. automatic term recognition

The first objective is met by measuring the KP range which is assumed to be an indication of the reliability of the target relation. The second objective can be met by the use of "BNC discounting" which is a heuristic introduced in this thesis to penalize arguments which are too general and thus likely to be terminologically irrelevant (see the discussion in subsection 2.2.3 on termhood and fuzziness). BNC discounting can be applied to any of the basic ranking schemes, "frq", "kpr" or "fkpr", and is computed by means of the following formula, exemplified by "kpr".

$$kpr_{bnc} = \frac{KPR_{sample}(i)}{log(C_{BNC}(i))}$$

When an argument is not seen in the BNC, the scheme will default to its main ranking style, i.e. "frq", "kpr" or "fkpr". In all other cases BNC filtering will *reduce* the overall reliability score of the candidate by dividing the main score by the logarithmized BNC frequency of the instance, i. For example, a term occurring 10,000 times in the BNC will have its ranking score divided by log(10,000) = 9.21 while one occurring 100 times in the BNC is only penalized by a factor of 4.60. In this way BNC discounting is a somewhat more conservative way of assessing the termhood of an argument than using, for instance, ratios of relative frequencies in a general versus a specific corpus (also known as term "keyness" or "weirdness" (see [Ahmad, 1993]).

It should be noted that the BNC discounting is based on unigrams and thus is not applied to instance ngrams where n > 1. The main reason for this is that the BNC is a relatively small corpus by today's standards (it contains only 100 million tokens), so a bigram frequency list produced from the BNC will likely be even more affected by data sparseness than a unigram list, and thus be unreliable. Had the author been aware that Google published comprehensive ngram statistics based on a trillion web pages in the Fall of 2006, true ngram discounting might have been employed, but this will be implemented in future versions of WWW2REL. Finally, BNC filtering is expected to be most useful in combination with an additional heuristic which happens to provide it with unigrams. This heuristic is a head noun grouping strategy to be explained in subsection 5.3.2.

5.3.2 Head grouping heuristic

The head noun grouping heuristic can also be applied to the main ranking schemes. It works by grouping all candidate instances by their NP heads and executing a *primary ranking* of these heads based on the main scheme, i.e. "frq", "kpr" or "fkpr". Based on the same scheme it then ranks all NPs sharing a particular head (see appendix 8.20 for details). [Gillam, 2004, Gillam et al., 2005] report on a system which induces taxonomies automatically from text by identifying prominent "mother terms" (i.e. statistically "weird" unigrams) and then recursively expanding these into more complex terms by identifying words which collocate with the NP head. However, the idea of grouping relation instances, including *non-taxonomical* ones, by their NP head to boost system precision (and recall) is unique to this thesis as far as the author is aware. It is hypothesized that head grouping will not only improve system performance but may also provide the user (i.e. the terminologist) with a better interface in which possible hyponyms of selected candidates can be expanded or hidden as need be.

That grouping candidates by head noun could be a useful strategy for boosting performance is illustrated by the examples in table 42. In this case all the candidate hypernyms of "haloperidol" which have "antipsychotics" as their head noun are, in fact, judged to be correct by the four experts. Of course, grouping by head noun may also create clusters of incorrect candidates, but given the hypothesis that the KP range of an instance (or a cluster of instances) is a useful indicator of its reliability, this should

Table 42: All candidates of "haloperidol ISA X" where the head is "antipsychotics"

judgment	candidate	head
1,1,1,1	classical antipsychotics	antipsychotics
1,1,2,1	conventional antipsychotics	"
1,1,1,1	first generation antipsychotics	"
1,1,1,2	older antipsychotics	"
1,1,2,1	traditional antipsychotics	"
1,1,1,1	typical antipsychotics	"

be no problem, because when grouping instances by their head, the KP range is simply computed for the heads of the candidates rather than the individual NPs.

5.3.3 Hypotheses

Concerning the effectiveness of the proposed ranking schemes the following hypotheses will be tested in sections 5.4 and 5.5.

1. The "kpr" scheme will achieve higher precision than the baseline "frq" scheme as KP range is a more appropriate measure of relation strength than pure frequency.
2. Grouping by noun head will boost performance because it conflates non-restrictive linguistic variations.
3. Applying BNC-based instance discounting will boost performance because a high BNC frequency is an indicator of low termhood.
4. "pkpr_bnc" is the best reliability measure because it uses two key features of LSP to determine the termhood of arguments.
5. The pmi-based schemes will not be worth the extra (computational) effort because their expected frequencies are unreliable when based on a small sample of text.
6. Using more technical (domain-specific) synonyms as input terms will improve both recall and precision.
7. Identifying non-taxonomical, conceptual relations ("induces" and "may_prevent") is a harder task than identifying ISA and synonymy relations.

5.4 Evaluation of ranking schemes

The data analyses in this section will compare the performance of the different ranking schemes and the two heuristics proposed in section 5.3 to determine which schemes and which heuristics perform the best across all eleven experiments and also to test some of the hypotheses proposed in subsection 5.3.3. For each of the eleven experiments precision, recall and F scores are computed using the three formulae introduced in section

Table 43: correct candidates in individual experiments

term	relation	correct/total
aspirin	induces	148/365
selenium	may_prevent	50/421
vomiting	induces	59/317
emesis	induces	25/76
formaldehyde	synonymy	4/46
vitamin C	synonymy	5/63
lactose	synonymy	1/41
glucose	synonymy	4/100
progesterone	synonymy	2/61
antipsychotic(s)	ISA (hyponymy)	88/225
haloperidol	ISA (hypernymy)	57/141

3.2 as implemented in appendices 8.19 (no head grouping), 8.20 (head grouping) and 8.21 plus 8.23, respectively.

In text mining, IE and IR literature the performance of applications is often given only as an F-score. However, in the context of this evaluation high precision is considered all-important. The reason is that the intended users of WWW2REL are terminologists needing assistance in the practical work of augmenting an ontology. If there is much noise among the top X relation instances returned by the system, the intended users are unlikely to benefit from using the system.

Also, although recall figures are interesting, one should not forget that in the terminological context (unlike the typical IE context) such figures will always be somewhat artificial in that terminological gold standards can rarely claim to be exhaustive. This is also the case for the gold standard established from system output and even the more comprehensive UMLS Metathesaurus as such. In fact, since WWW2REL is conceived as a knowledge discovery application, measuring to what extent it can find relation instances already recorded in the target ontology is less important than assessing its ability to extract *unrecorded* but relevant relation instances with high precision. Recall versus the UMLS and the proportion of unrecorded knowledge returned by the system is the topic of section 6.4.

Finally, system users should have the option of raising or lowering the minimum candidate frequency threshold (see table 38), but the effects of this parameter on system performance are not explored in this thesis.

Table 43 provides statistics on the number of correct and relevant candidates in the individual experiments as a reference point for the following analyses. Correct candidates are defined as candidates having an average judgment equal to or less than 1.50. As will become apparent when analyzing the results of the individual experiments in section 5.5, the five synonymy experiments involve far fewer correct candidates than the other experiments.

In order to measure the correlation between the ranking performed by the four human experts (the gold standard) and the ranking performed by the various automatic ranking schemes, the use of a matched pairs rank test was considered (see e.g.

[Oakes, 1998]). Rather than producing a single number of the overall degree of correlation between the gold standard and the system ranking, however, it seemed more appropriate to visualize the degree of correlation as a graph (or rather lots of graphs) by plotting the accumulated precision scores of the system ranking versus the gold standard. As the graphs can get too cluttered, however, tables are also provided.

5.4.1 Ranking by frequency ("frq")

Figure 15 plots precision scores using the baseline "frq" ranking scheme for all experiments involving the causal relations, the ISA relation and synonymy, respectively. In the case of the ISA relation, precision is high for the first five or so candidate instances, but then drops fairly quickly. When applied to the synonymy experiments the precision of "frq" also seems to trail off quickly. In these experiments, however, there are only very few correct candidates (see table 43) so a quick drop in precision is to be expected. Finally, in three out of the four causality experiments, namely "selenium may_prevent X", "X induces vomiting" and "X induces emesis", the "frq" scheme overall performs extremely poorly with precision scores in the 0.30 and 0.20 range. The "X induces vomiting | emesis" experiments are not displayed here but discussed in subsections 5.5.3 and 5.5.4.

5.4.2 Ranking by KP range ("kpr")

Again, figure 16 plots precision scores for experiments involving the causal relations, the ISA relation and synonymy, but this time using the "kpr" scheme in which instances are ranked by the range of different KPs with which they occur in the sample. In the case of the ISA relation both experiments suggest that "kpr" does give considerably higher precision rates than "frq" also in the longer run (top graph in figure 15). In the "selenium" experiment the precision boost is negligible, and for "aspirin" simple frequency ranking is actually preferable. However, as will be discussed in subsection 5.5.4 there is a drastic precision boost for "emesis" when using "kpr" rather than "frq". Finally, for the five synonymy experiments "kpr" is only slightly better than "frq".

5.4.3 Ranking by "fkpr"

This time instances are ranked by their KP range multiplied by their sample frequency, or "fkpr" for short. Figure 17 plots the precision scores for experiments involving the causal relations, the ISA relation and synonymy, respectively. For the causality experiments "fkpr" is slightly better than "frq" in the "selenium" experiment but more or less the same for "aspirin". While "fkpr" performs better than the baseline in both ISA experiments, it performs worse than "kpr". Finally, for synonymy "fkpr" seems to be slightly better than both "kpr" and "frq".

5.4.4 Ranking by "pmi"

As evidenced by figure 18 the "pmi" ranking scheme makes little difference in the "aspirin" and "selenium" experiments, although it is somewhat better than the baseline

Figure 15: Ranking by “frq” (ISA, causality and synonymy)

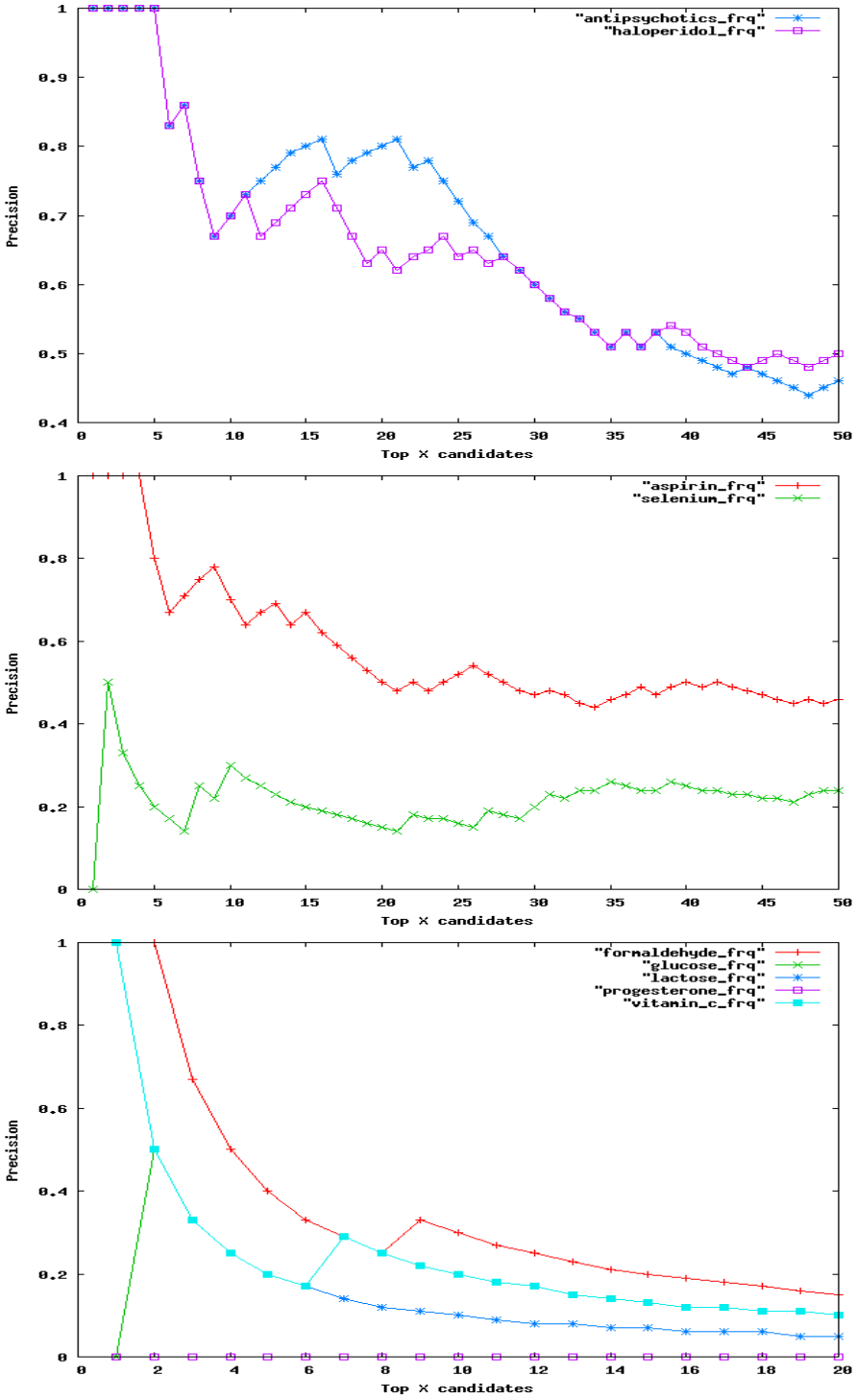


Figure 16: Ranking by “kpr” (ISA, causality and synonymy)

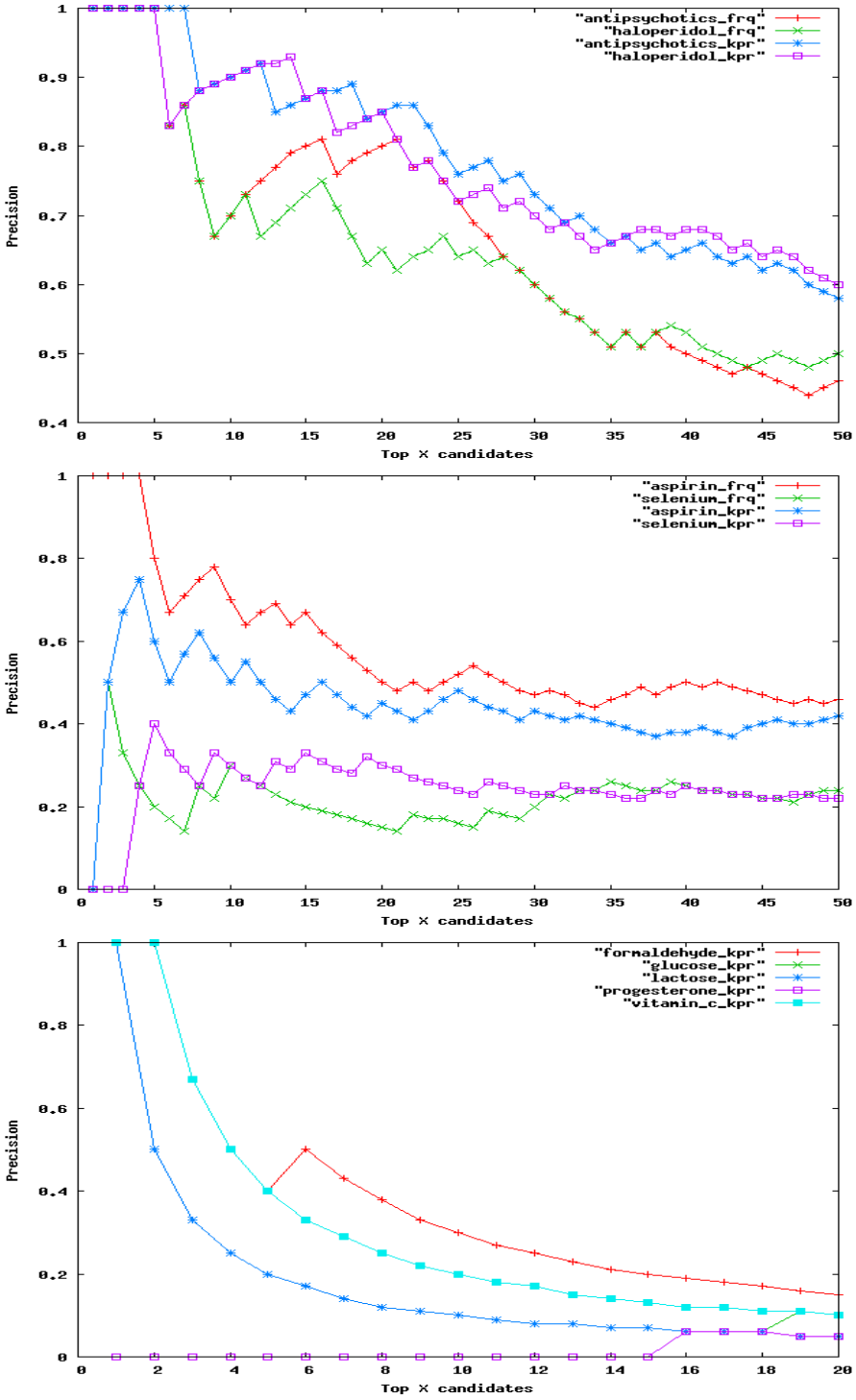


Figure 17: Ranking by “fkpr” (ISA, causality and synonymy)

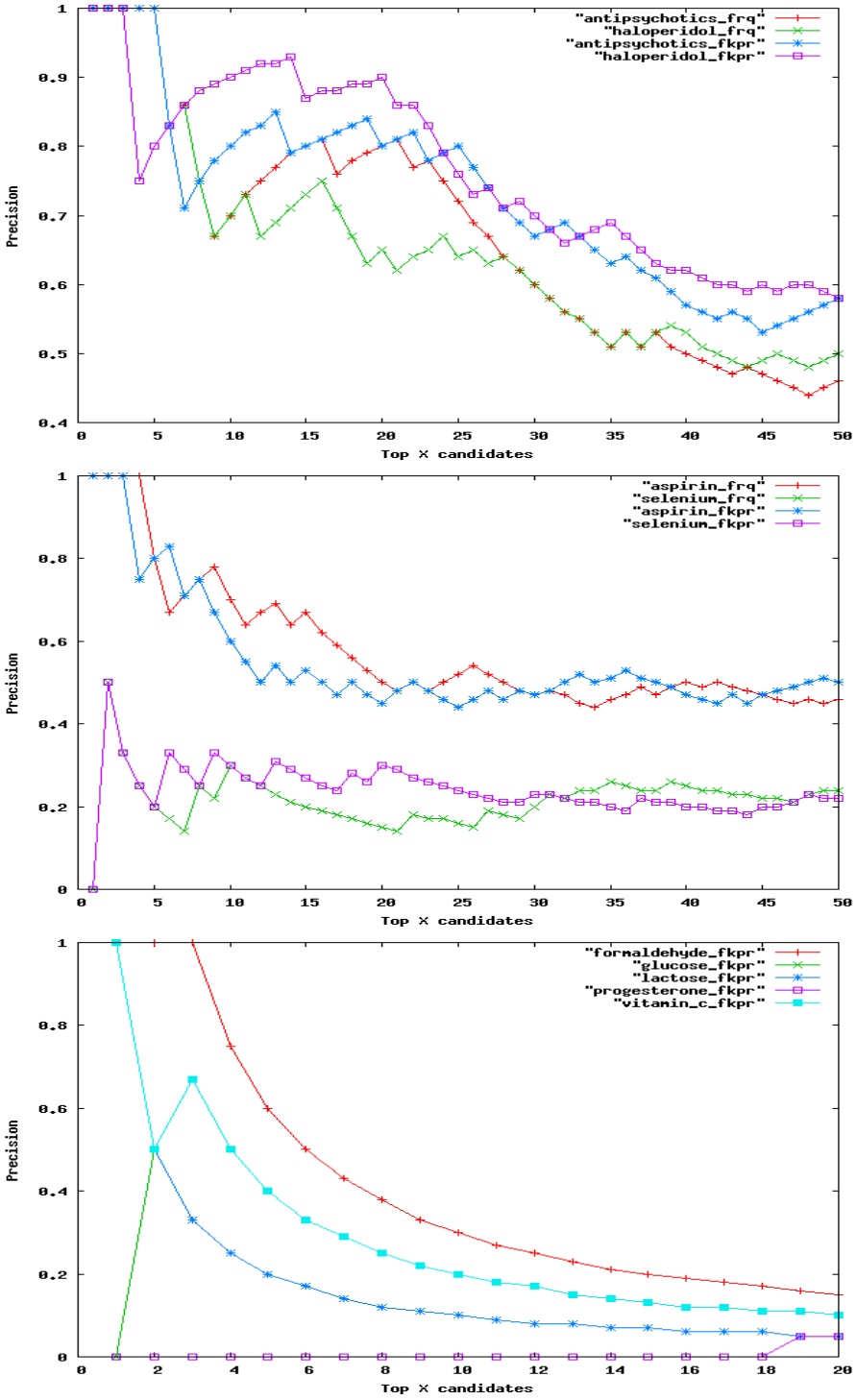
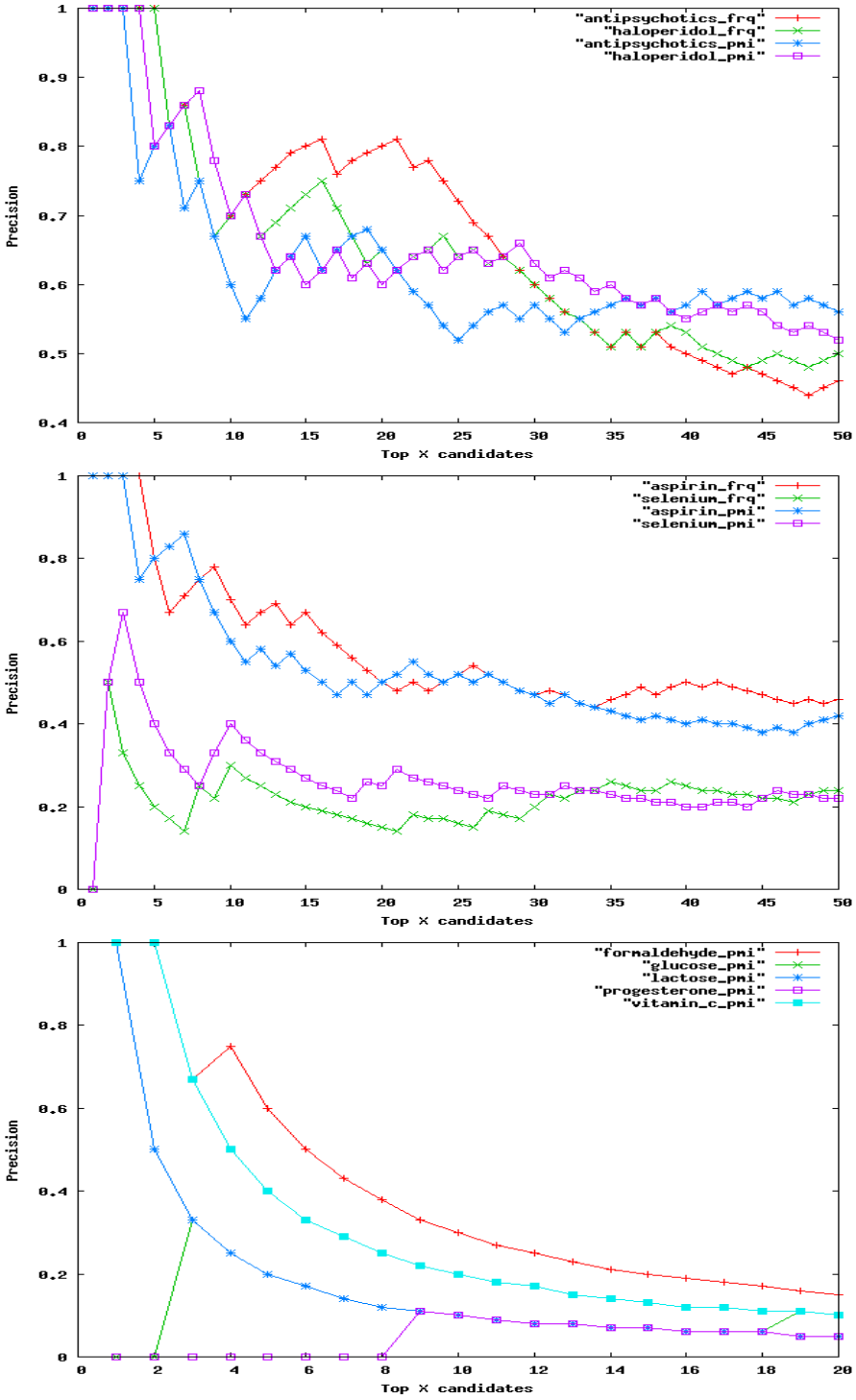


Figure 18: Ranking by “pmi” (ISA, causality and synonymy)



in the latter case. However, in subsections 5.5.3 and 5.5.4, it will be discussed how "pmi" is, in fact, one of the best ranking schemes in the "X induces vomiting | emesis" experiments.

For the ISA relation, "pmi" performs worse than the baseline "frq" scheme as precision rates drop very quickly. In this case "kpr" appears to be the best choice. Finally, for synonymy "pmi" appears slightly better than the other schemes in that it is the first scheme to identify a correct candidate in the "progesterone" experiment.

5.4.5 Applying BNC-based discounting

Seeing as "kpr" was one of the best ranking schemes so far, the BNC-based discounting filter is applied to this particular scheme so as to test its effect on precision. Figure 19 indicates that discounting relation instances which occur frequently in the general language corpus, the BNC, does indeed improve precision. The tendency is very clear in the two ISA experiments (the top graph in figure 19). Especially when finding hyponyms of the input term "antipsychotic(s)" BNC discounting boosts precision tremendously. Even if less conspicuous the positive impact on precision is also visible when looking for hypernyms of "haloperidol". It makes good sense that BNC discounting has a greater impact at the lower levels of an ontology which is typically where the more specialized terms are found.

In the causal experiments BNC-based discounting clearly boosts precision in the "selenium may_prevent X" experiment. This experiment scores quite poorly with most other ranking schemes, but now gets good precision rates in the 0.70 to 1.00 range. Given the low degree of inter-annotator agreement and the high proportion of "2" judgments (i.e. unsure/vague) in this particular experiment, higher precision rates could probably not be expected. However, the BNC-based filter does not seem to boost precision for the "aspirin induces X" experiment. When taking a closer look at the output candidates for this experiment in subsection 5.5.1 it will become apparent why the BNC-based filter does not boost precision in this case. Finally, for the five synonymy experiments (the bottom graph in figure 19) the BNC-based filter only has a marginally positive effect on precision.

5.4.6 Applying head grouping

To test whether or not grouping instances by their noun head will improve precision, head grouping is activated for the "kpr" scheme and the results are compared across all experiments (see figure 20). Across all experiments and all three relation types there is a clear tendency that head grouping boosts the precision of the "kpr" scheme.

In the experiments for the two causal relations head grouping has a significant positive and sustained impact on precision in both the "aspirin" and "selenium" experiments, although the impact is much more conspicuous in the former case. Head grouping applied to the "kpr" scheme also works brilliantly in the experiment of finding hypernyms of "haloperidol". When finding hyponyms of "antipsychotic(s)" head grouping has a negative impact on the first 10 candidates, but after this initial drop in precision grouping candidate heads by their "kpr" also proves the best ranking scheme for the extraction of hyponyms. Also in the case of synonymy head grouping boosts

Figure 19: Ranking by “kpr_bnc” (ISA, causality and synonymy)

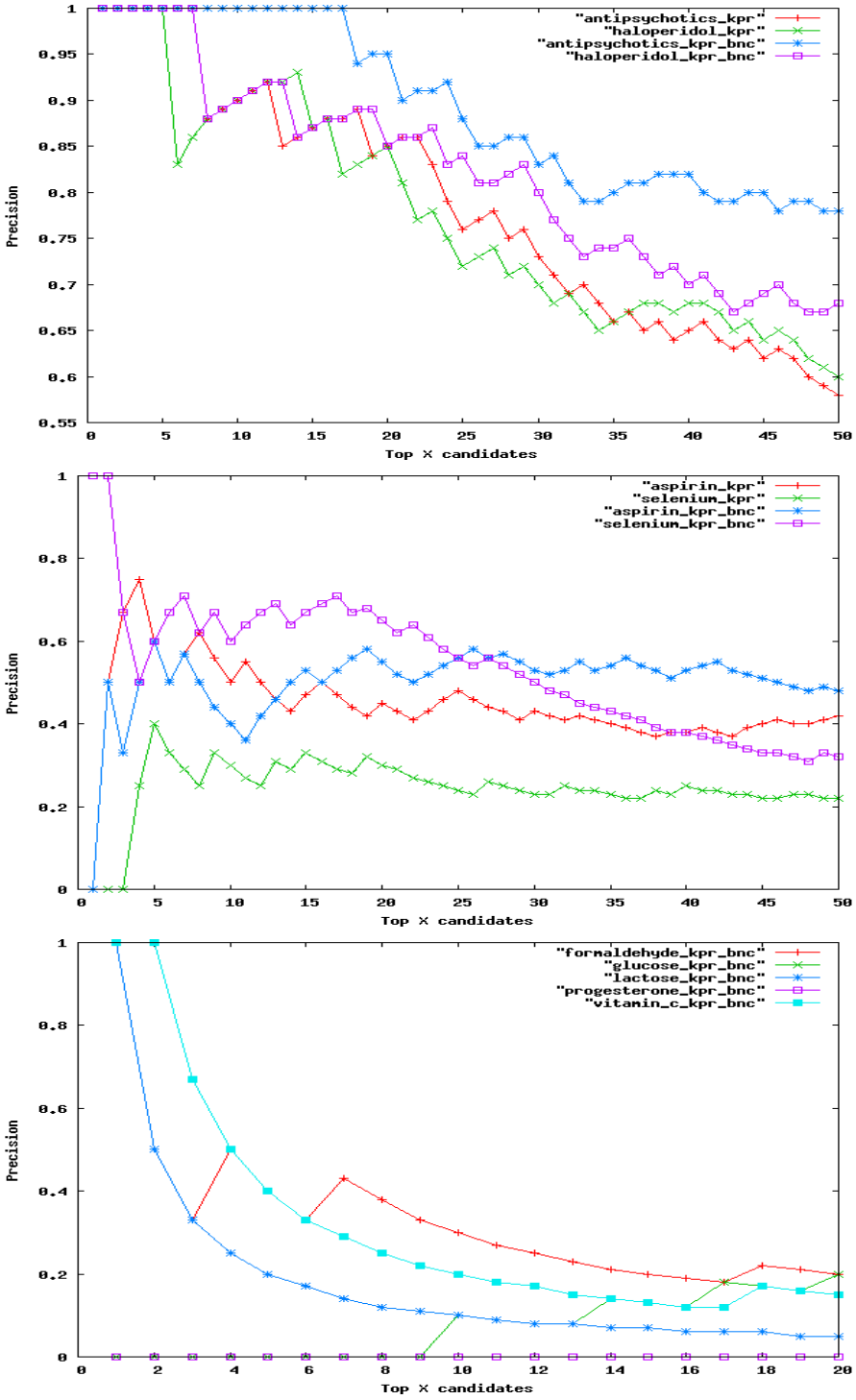
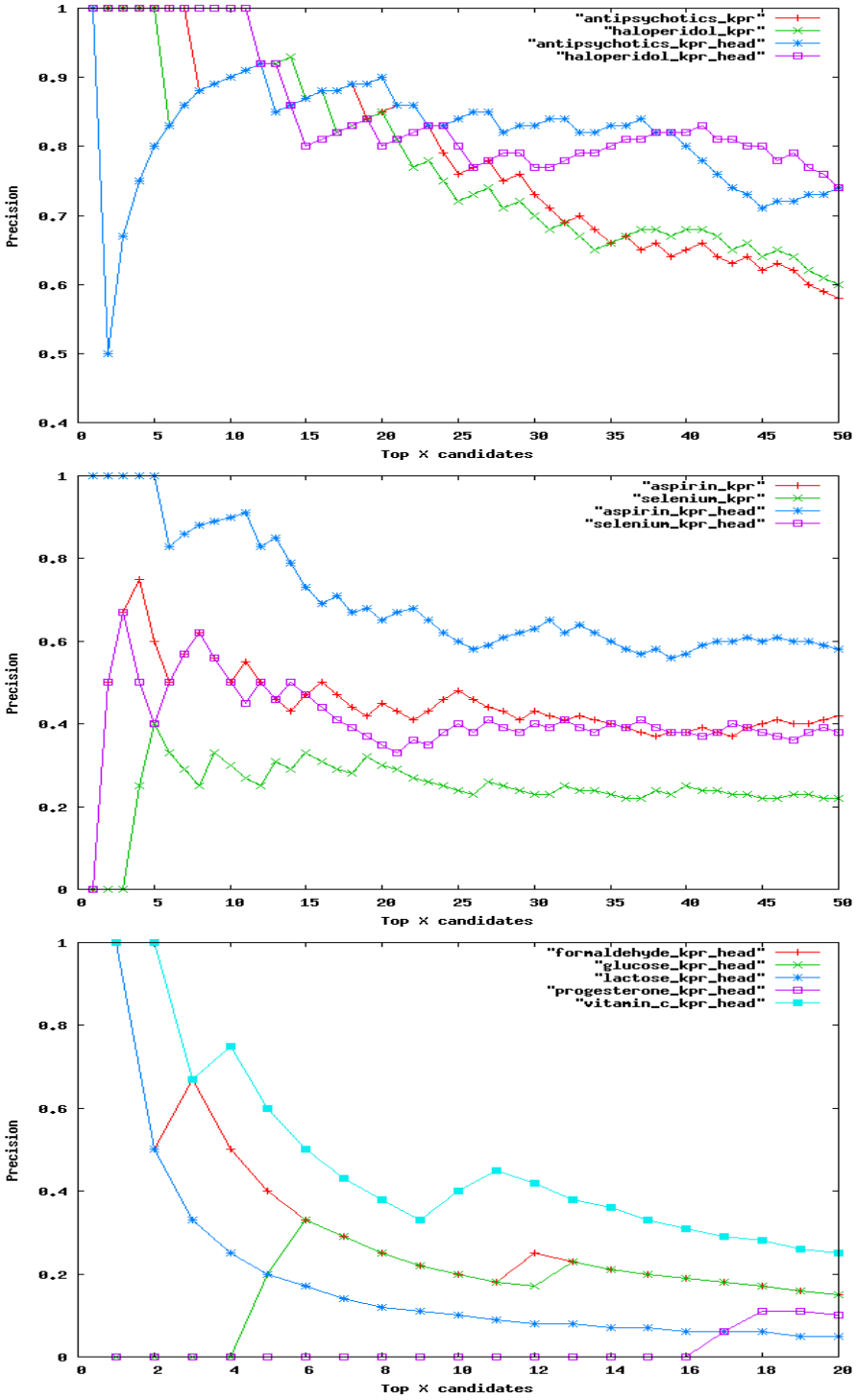


Figure 20: Ranking by “kpr_head” (ISA, causality and synonymy)



precision, especially for the "vitamin c" experiment in which more correct synonyms are among the high ranking candidates.

5.4.7 Combining both heuristics

Since both head grouping and BNC-based discounting appears to have a significant and positive impact on the precision of the "kpr" ranking scheme, the two heuristics will now be activated simultaneously so as to explore whether they may work in combination. The results of applying this combined filter across all experiments (except "X induces vomiting | emesis") can be seen in figure 21.

Overall, combining the two heuristics does boost precision for the causal relations and the ISA relation as compared to the individual application of each heuristic. Thus for the "aspirin" (and also "emesis" and "vomiting") precision is boosted, but in the case of "selenium" higher precision rates are achieved by using the BNC-based filter *without* head grouping. In the two ISA experiments combining the two filters yields the best precision scores presented in this section. Indeed, as evidenced by figure 21 precision is now nearly 90% for the top 50 hyponyms of "antipsychotic(s)". In the case of synonymy combining the two filters appears to be a bad idea. Only in a single experiment ("progesterone") is precision boosted, but in two experiments ("vitamin c" and "lactose") the combination causes precision rates to be lowered.

5.4.8 Conclusion

The graphs presented in this section revealed that while ranking relation instances by their simple frequency (the baseline scheme) may yield high precision in some cases (for example for synonymy and ISA relations), precision quickly deteriorates. While "pmi" is a useful scheme for identifying a few instances with high precision, its precision rate also deteriorates rather quickly. In comparison with the baseline ranking scheme both "kpr" and "fkpr" achive considerably higher precision scores, while "pmi" is dubious. The graphs also indicated that applying either head grouping or BNC-based instance discounting can provide significant precision boosts as compared to the unmodified "kpr" ranking scheme. In fact, combining the two heuristics boosted precision even further for all the investigated relation types except synonymy. As evidenced by the complete F-score plots in appendices 8.40, 8.41, 8.42 and 8.43 the two heuristics not only boost precision but also recall, both when activated individually and even more so when used in combination.

The first three hypotheses formulated in section 5.3 are thus validated.

5.5 Evaluation of experiments

In the data analyses presented in this section the perspective is changed from evaluating the performance of the various ranking schemes to evaluating performance in the individual experiments. The analyses not only test the remaining hypotheses proposed in subsection 5.3.3, but also discuss challenges particular to each experiment. Each subsection presents examples of the top 10 relation instances proposed by WWW2REL for the particular experiment along with the domain expert judgments they were assigned.

Figure 21: Ranking by “kpr_bnc_head” (ISA, causality and synonymy)

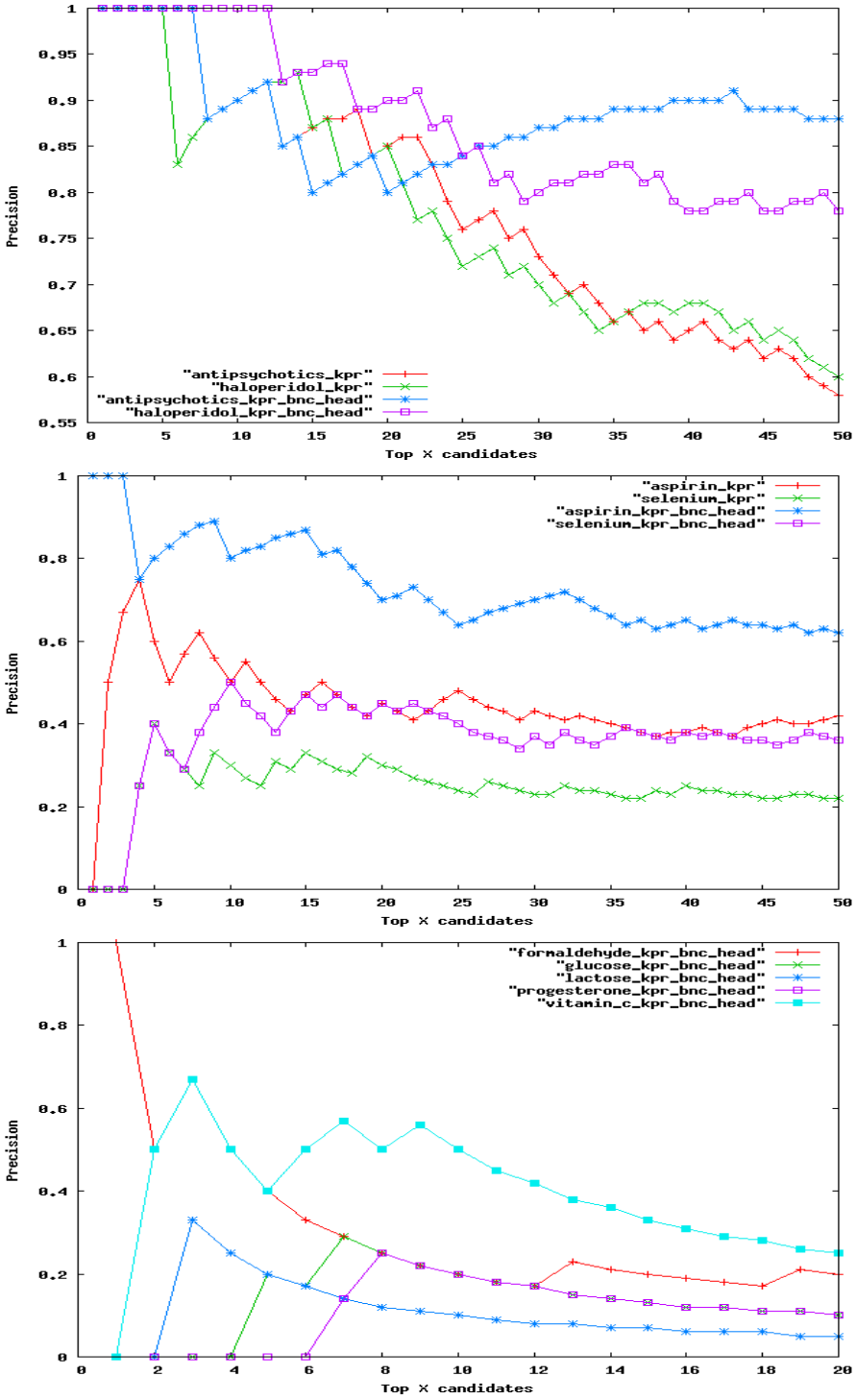
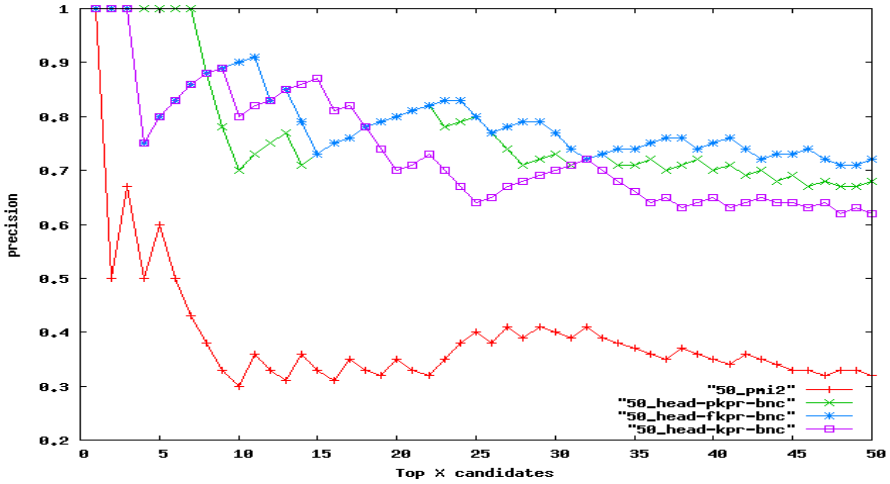


Figure 22: Aspirin induces X - assorted ranking schemes



As in section 5.4 precision is considered critical in the evaluation. Nevertheless, occasional references will be made to the recall or F-score of individual ranking schemes. Plots will only be provided to elucidate experiment particularities not illustrated by the plots in section 5.4.

5.5.1 Aspirin induces X

Figure 22 and table 45 summarize the performance of all ranking schemes applied in the "aspirin induces X" experiment. In the very short run "head-pkpr-bnc" is the top performing scheme, i.e. adding the small set of 13 passive KPs in appendix 8.32.1 actually boosts precision. However, this scheme quickly runs out of steam and in the longer run, "head-fkpr-bnc" proves to be the best scheme of them all, achieving a precision of 0.80 for its top 25 candidates. Even without head grouping "fkpr-bnc" is still the best scheme when looking beyond the top 10 candidates. The "pmi2" variant is the worst ranking scheme.

Interestingly, "head-fkpr-bnc" is considerably better than "head-kpr-bnc" which was considered the overall best ranking scheme in section 5.4. The explanation is presumably that the high ranking head "bleeding" gets penalized by the BNC discounting heuristic, but less so when its KP range is multiplied by its frequency (fkpr) than when it is not (kpr).

Table 44 lists the top 10 instances returned by the two ranking schemes "frq" and "head-fkpr-bnc" along with their expert judgments. While ranking by simple frequency is not a bad scheme (for the top 10 candidates), it is evident from the table that the top 10 candidates returned by "head-fkpr-bnc" are judged to be more correct.

As for performance in terms of F-scores the figure in appendix 8.42 shows that ranking schemes using the BNC-based filter outperform the other schemes throughout the sample. A second observation which can be made from these plots is that grouping

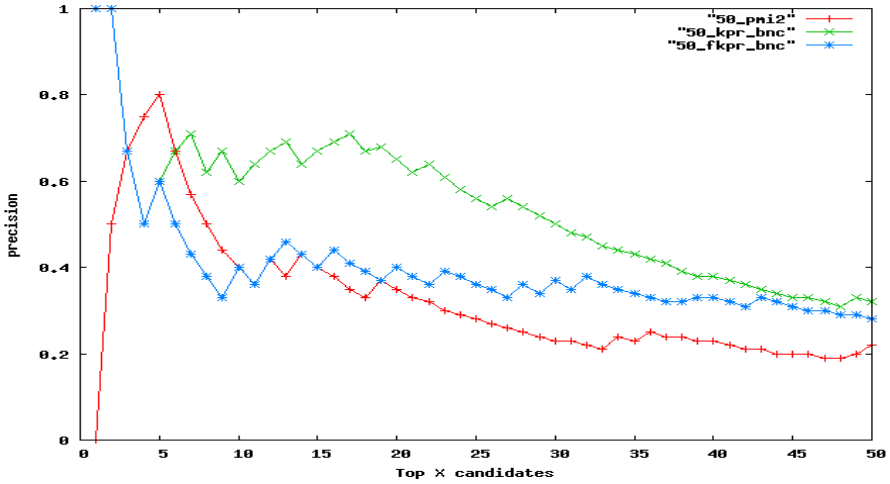
Table 44: Aspirin induces X - top 10 candidates

rank	candidate ("frq")	judgment	candidate ("head-fkpr-bnc")	judgment
1	apoptosis	1,1,2,1	bleeding	1,1,2,2
2	bleeding	1,1,2,2	gastrointestinal bleeding	1,1,1,1
3	asthma	1,1,2,1	stomach bleeding	1,1,1,1
4	ulcers	1,2,2,1	more bleeding	1,1,2,3
5	ringing	1,2,2,3	internal bleeding	1,1,2,1
6	patency	2,2,2,1	increased postoperative bleeding	1,1,1,3
7	headache	1,1,1,1	ulcer bleeding	1,1,1,1
8	gastrointestinal bleeding	1,1,1,1	gastric bleeding	1,1,1,1
9	tinnitus	1,1,2,1	increased bleeding	1,1,2,1
10	reye's syndrome	1,1,2,3	liver damage and stomach bleeding	1,1,2,1
...	...	...	...	

Table 45: "Aspirin induces X": precision of sample-based schemes

scheme	top 3	top 10	top 25	scheme	top 3	top 10	top 25
kpr	0.67	0.50	0.48	kpr_head	1.00	0.80	0.60
kpr_bnc	0.33	0.60	0.56	kpr_bnc_head	1.00	0.80	0.64
fkpr_bnc	0.67	0.40	0.64	fkpr_bnc_head	1.00	0.90	0.80
pkpr_bnc	0.67	0.50	0.56	frq_bnc_head	0.67	0.80	0.68
frq	1.00	0.70	0.52	frq_head	0.67	0.90	0.68
fkpr	1.00	0.60	0.44	fkpr_head	1.00	0.90	0.64
pmi	1.00	0.60	0.52				
pmi2	0.67	0.30	0.40				

Figure 23: Selenium may_prevent X - assorted ranking schemes



by head noun improves performance not only in terms of precision but also recall.

5.5.2 Selenium may_prevent X

Figure 23 and table 47 illustrate how the various ranking schemes fare in the "selenium may_prevent X" experiment.

The first thing which strikes the eye is that precision rates drop much faster than in the "aspirin" experiment. Secondly, it does not seem as if grouping by head noun has as positive and lasting an effect on performance as it had for "aspirin". As reflected by the figure and the numbers in table 47, the best performing ranking scheme is "kpr-bnc", which performs dramatically better (56% for top 25) than when turning the BNC filter off (24% for top 25). Similarly, the "pmi" and "pmi2" schemes, which are also tuned to detect rare events, perform relatively well in this experiment.

Table 46 lists the top 10 candidates returned by "kpr-bnc" and its filterless version "kpr". With BNC discounting highly relevant relation candidates are returned, but when the filter is turned off the picture is less promising. One reason for this is the crude NLP performed by the system. When NPs are postmodified by PPs, fronting is attempted (as illustrated in table 34), but when this fails the PP is simply omitted in an attempt to reduce the number of term variants and boost recall. Candidates like "incidence", "rate" and "number" are an unfortunate consequence of this strategy. They are themselves semantically vague, but in the original text snippets they were presumably followed by one or more PPs holding the core meaning of the complex NP. Fortunately, these meaningless NP fragments are eliminated by the BNC filter which makes up for the system's lack of syntactic processing.

In terms of F-scores the figure in appendix 8.43 reveals "kpr-bnc" to be the best scheme in the short run, while "kpr-bnc-head" is the best scheme in the slightly longer run. Interestingly, it also that performance peaks at a much earlier stage for selenium

Table 46: Selenium may_prevent X - top 10 candidates

rank	candidate (“kpr”)	judgment	candidate (“kpr-bnc”)	judgment
1	risk	3,2,2,2	prostate cancer risk	1,1,1,1
2	cancer	2,1,2,2	prostate cancer	1,1,1,1
3	incidence	3,2,2,2	cancer risk	2,1,2,2
4	prostate cancer risk	1,1,1,1	toxic effects	2,1,2,2
5	prostate cancer	1,1,1,1	prostate cancer incidence	1,1,1,1
6	development	3,2,2,2	lung cancer risk	2,1,1,2
7	cancer risk	2,1,2,2	oxidative damage	2,1,2,2
8	rate	3,2,2,3	cancer incidence	2,1,2,1
9	toxicity	2,1,2,2	dna damage	1,1,2,2
10	number	3,2,2,3	tumor growth	2,1,1,1
...	...	...	...	

Table 47: “Selenium may_prevent X”: precision of sample-based schemes

scheme	top 3	top 10	top 25	scheme	top 3	top 10	top 25
kpr	0.00	0.30	0.24	kpr_head	0.67	0.50	0.40
kpr_bnc	0.67	0.60	0.56	kpr_bnc_head	0.00	0.50	0.40
fkpr_bnc	0.67	0.40	0.36	fkpr_bnc_head	0.33	0.20	0.40
frq	0.33	0.30	0.16	frq_head	0.67	0.20	0.40
fkpr	0.33	0.30	0.24	fkpr_head	0.33	0.20	0.40
pmi	0.67	0.40	0.28	frq_bnc_head	0.67	0.30	0.32
pmi2	0.67	0.40	0.28				

than for aspirin. By the top 19 candidates "kpr-bnc" achieves an F-score of 0.38 for "selenium", but at the same point for "aspirin" its F-score is only 0.15. In comparison with "aspirin induces X", top performance in this experiment is achieved about one fourth of the way through the 421 candidates. This presumably reflects the fact that in this particular experiment no frequency threshold was enforced and singletons were allowed in the sample. In fact, 238 out of the 421 candidates for "selenium may_prevent X" occurred only once in the sample snippets. In perfect agreement with Zipf's law of word frequency distributions in natural language it might be added. Since many of these singletons appear to have been judged "incorrect" by the experts, enforcing a minimum sample frequency threshold of 2 would have drastically reduced the number of candidates presumably with only a minimal reduction of recall.

An expert observation must be mentioned at this point. During the experiment the annotators commented that the beneficial effects of selenium are still being investigated by the scientific community and that many proposed effects of selenium are being debated and undergoing scrutiny. This made their evaluation difficult and explains the poor degree of inter-annotator agreement in this particular experiment as indicated in table 40. It also stresses that many of the results reported in this evaluation are, in fact, underestimated in that it is often the retrived knowledge rather than the system itself which is questioned. In the "selenium" experiment there are thus many cases of the fourth type of noise listed in section 1.1. Namely where the KPs realize the target semantic relation and the arguments of this relation are indeed domain specific, but where the correctness of the knowledge expressed by the relation is questionable.

5.5.3 X induces vomiting

Using a disease (or rather a symptom) as input and trying to identify the causes of this disease is inspired by [Mukherjea and Sahay, 2006] who identify the causes of Typhoid. The numbers in table 43 show how the query template "X <induces> vomiting" results in a much larger number of candidates for X (317) than the template "X <induces> emesis" (76). This, of course, reflects the fact that the occurrence of the more technical of the two synonyms, i.e. "emesis", is somewhat rarer on the WWW at large. As the following plots will reveal, however, relative rarity need not affect the quality of the query results in any negative manner. On the contrary, in fact.

Figure 24 and table 49 illustrate how precision is remarkably low almost regardless of which ranking method is applied. "Pkpr-bnc" is the best scheme in terms of precision, but it quickly trails off, so in the longer run "fkpr-bnc" is a safer option. Table 48 lists examples of the top 10 candidates proposed by the "fkpr" and "pkpr-bnc" schemes, respectively. Clearly, the range of things which induce vomiting is so large that most of them are considered too vague to be biomedically relevant. Also in many cases it is possible to envision circumstances in which the argument might cause someone to vomit (for example chocolate), but this effect is not a distinctive feature of the argument so to speak. No wonder, the experts make heavy use of the judgments "2" and "3" in this experiment.

Also, in terms of F-scores performance is poor throughout the sample as reflected by the figure in appendix 8.44. Applying the two heuristics of BNC discounting and head grouping in the "head-kpr-bnc" scheme does boost overall performance, however.

Figure 24: X induces vomiting - assorted ranking schemes

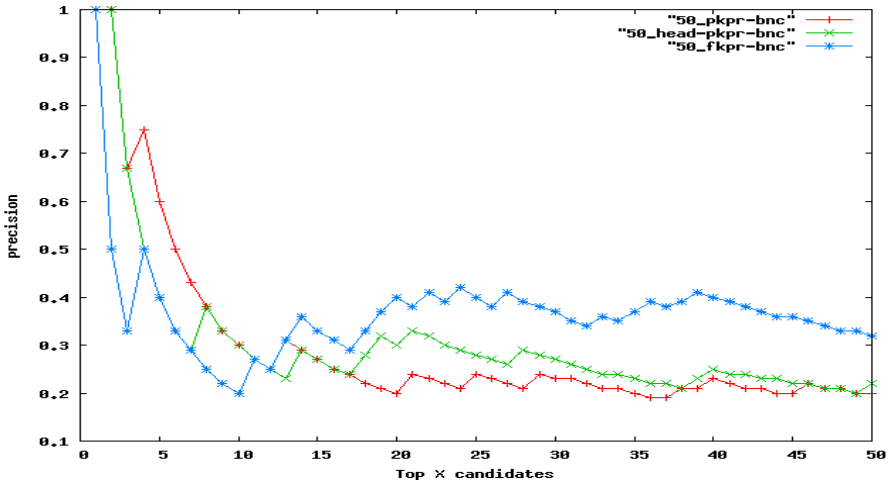


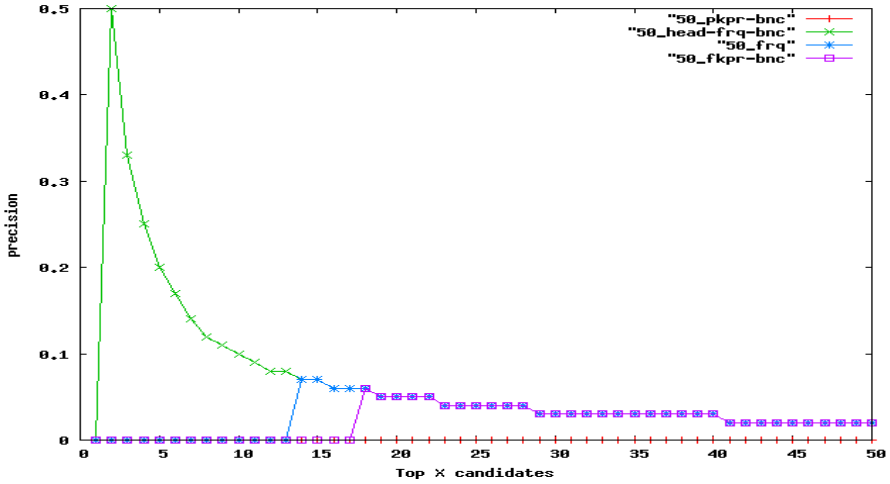
Table 48: X induces vomiting - top 10 candidates

rank		judgment ("fkpr")	candidate ("pkpr-bnc")	judgment
1	4,4,2,2	ipecac	ipecac	4,4,2,2
2	3,3,3,2	water	rotavirus	1,1,1,1
3	1,2,2,3	ingestion	gastrointestinal tract	2,2,3,3
4	1,2,3,2	medicine	large doses	1,2,2,2
5	2,2,2,3	stomach	ipecac syrup	4,4,4,3
6	2,3,2,3	your child	your child	2,3,2,3
7	1,3,1,1	chocolate	intracranial pressure	1,2,2,3
8	1,2,2,2	food	binge drinking	1,2,2,3
9	2,2,3,3	gastrointestinal tract	your veterinarian	2,2,2,3
10	4,4,4,3	ipecac syrup	your dog	1,3,2,3
...	...	...	...	

Table 49: "X induces vomiting": precision of sample-based schemes

scheme	top 3	top 10	top 25	scheme	top 3	top 10	top 25
kpr	0.33	0.20	0.28	kpr_head	0.33	0.20	0.24
kpr_bnc	0.67	0.30	0.20	kpr_bnc_head	0.67	0.30	0.28
fkpr_bnc	0.33	0.20	0.40	fkpr_bnc_head	0.33	0.20	0.20
pkpr-bnc	0.67	0.30	0.24	frq_bnc_head	0.67	0.30	0.24
frq	0.33	0.30	0.36	frq_head	0.00	0.00	0.20
fkpr	0.33	0.20	0.24	fkpr_head	0.33	0.10	0.16
pmi	0.33	0.30	0.28				
pmi2	0.00	0.40	0.24				

Figure 25: {drugs} induces vomiting - assorted ranking schemes



When enforcing the additional requirement that the candidates must also be drugs, performance drops even lower (cf. figure 25 and appendix 8.44.1). The four drugs in the sample are simply too infrequent to be effectively identified (as part of top 10) by most of the ranking schemes. However, when using head grouping and BNC discounting the single scheme, "head-frq-bnc", is able to identify "baclofen" as its second candidate. All other schemes fail. It should be stressed though that the experts disagreed whether "ipecac" and "ipecac syrup" are drugs. Had all experts assigned drug status to these two candidates, performance would have been less miserable.

5.5.4 X induces emesis

Figure 26 and table 50 trace the precision of the same arsenal of schemes as before only this time using the more technical synonym for vomiting, namely "emesis". First of all, it is obvious that performance, both in terms of precision but also F-scores (see appendix 8.45), is much higher when using "emesis" rather than "vomiting" as input term. The "kpr" and "pmi" schemes, neither of which rely on the BNC filter, are best in the short run, but they are outperformed by "fkpr-bnc" in the longer run. Head grouping does not appear to be particularly helpful, only in the very short run (top 3 candidates).

Table 50 provides a breakdown of the precision scores of the various ranking schemes at selected points. As can be seen by the top 10 candidates for "pmi" and "head-kpr-bnc" in table 51 part of the reason why the precision of "head-kpr-bnc" suddenly drops is due to the limited NLP of the system when dealing with candidates following the "ADJ doses of <drug>" template where the PP containing the drug is simply skipped because it cannot be fronted (cf. subsection 5.1.2). If the system were to be custom-tailored for the biomedical domain, this would be an important point for improvement.

In this experiment the experts were also asked to use the additional judgment "4" in

Figure 26: X induces emesis - assorted ranking schemes

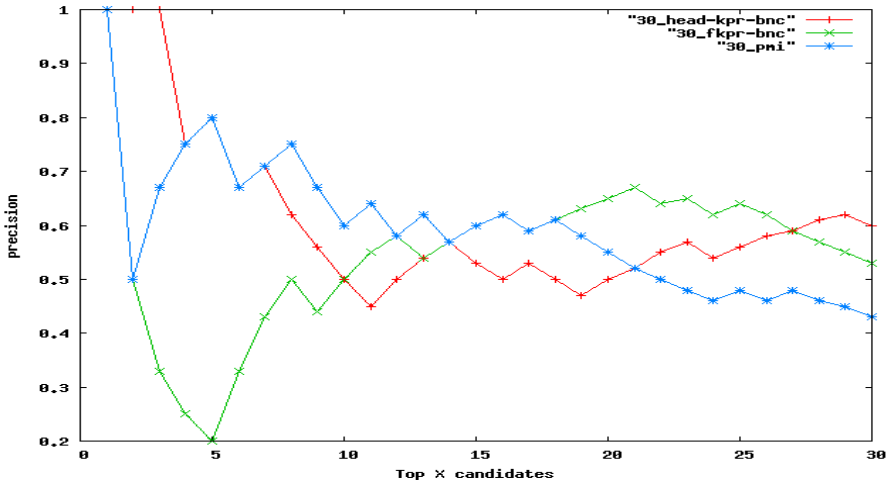


Table 50: "X induces emesis": precision of sample-based schemes

scheme	top 3	top 10	top 25	scheme	top 3	top 10	top 25
kpr	0.67	0.60	0.48	kpr_head	0.00	0.50	0.52
kpr_bnc	0.67	0.50	0.32	kpr_bnc_head	1.00	0.50	0.56
fkpr_bnc	0.33	0.50	0.64	fkpr_bnc_head	0.33	0.50	0.56
pkpr-bnc	0.67	0.50	0.40	frq_bnc_head	0.67	0.40	0.52
frq	0.00	0.50	0.60	frq_head	0.00	0.20	0.48
fkpr	0.67	0.50	0.56	fkpr_head	0.00	0.50	0.40
pmi	0.67	0.60	0.48				
pmi2	0.67	0.60	0.36				

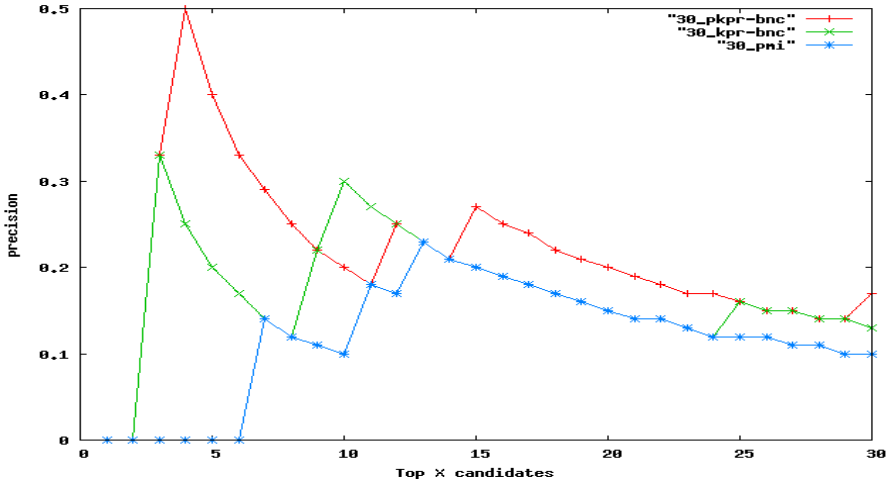
Table 51: X induces emesis - top 10 candidates

rank		judgment ("pmi")	candidate ("head-kpr-bnc")	judgment
1	4,4,4,1	ipecac	ipecac	4,4,4,1
2	2,2,2,2	large doses	carboplatin	4,4,4,4
3	4,2,2,1	drug	apomorphine	4,4,4,4
4	4,2,2,1	drugs	xylazine	3,2,1,4
5	1,1,1,1	chemotherapy	cisplatin	4,4,4,4
6	3,2,2,3	studies	pid	2,2,3,3
7	4,4,4,4	cisplatin	chemotherapy	1,1,1,1
8	1,2,1,1	chemoreceptor trigger zone	large doses	2,2,2,2
9	2,2,2,2	high doses	divided doses	2,2,2,3
10	3,2,3,3	brain	high doses	2,2,2,2
...	...	...	...	

Table 52: Drugs which induce emesis|vomiting

emesis (perfect agr.)	emesis	vomiting (perfect agr.)	vomiting
methotrexate	ipecac	loperamide	ipecac syrup
levodopa		aspirin	iron
tramadol		lithium	guaifenesin
paraplatin		baclofen	phenylephrine
carboplatin			moricizine
apomorphine			zoloft
cisplatin			
morphine			
TOTAL: 8/76	1/76	4/317	6/317

Figure 27: {drugs} induce emesis - assorted ranking schemes



those cases where X not only induces the effect ("emesis"), but also is a drug. Table 52 lists those drugs retrieved by the two query templates and having either perfect agreement[59] or near-perfect agreement[60]. Interestingly, there is almost no overlap between the drugs retrieved by the two query templates. In absolute numbers they yield nearly the same number of drugs, but relatively speaking 12% of all candidates are drugs when using the synonym "emesis", but only 3% when using the more domain-independent synonym "vomiting". When accepting only perfect agreement, as is done in the graphs in the appendices, these percentages drop to 11% and 1%, respectively.

While the task of identifying *drugs* which induce a particular effect is obviously more difficult than just finding *things* which induce this effect, figure 27 and table 53

[59] all four experts gave the judgment "4"

[60] three out of the four experts gave the judgment "4"

Table 53: {drugs} induce emesis": precision of sample-based schemes

scheme	top 3	top 10	top 25
kpr	0.00	0.10	0.16
kpr_bnc	0.33	0.30	0.16
fkpr_bnc	0.00	0.10	0.24
pkpr-bnc	0.33	0.20	0.16
frq	0.00	0.00	0.16
fkpr	0.00	0.10	0.16
pmi	0.00	0.20	0.12
pmi2	0.00	0.20	0.12

show that when using the input term "emesis" rather than "vomiting", the task is by no means impossible. The two schemes, "pkpr-bnc" and "kpr-bnc", which are both based on BNC discounting, are the best options, and they outperform the two pmi-based variants by a wide margin. Adding the passive construction KPs indeed appears to strengthen the reliability of the causal relation instances retrieved.

Again, the main reason why "pkpr-bnc" and "kpr-bnc" are the best ranking schemes in the task of identifying drugs with a particular effect, is that drug names are relatively rare in a general language corpus like the BNC. They are thus largely unaffected by the BNC frequency discounting because they have a high "weirdness" to use the terminology of [Ahmad, 1993]. That the pmi-based schemes are outperformed is perhaps surprising seeing as they are also tuned to identify rare events. However, the relatively small text sample may make it difficult to get a correct assessement of the expected frequencies on which pmi relies.

5.5.5 Synonymy

Before analyzing in detail the results of the five synonymy experiments it is worthwhile to consider the fact that identifying synonyms in the strict, terminological sense is harder than simply identifying semantically related words. For example, many of the words clustered in Wordnet's so-called synsets would not be considered true synonyms when exposed to a thorough conceptual analysis.

> Work in computational linguistics related to synonym detection has mainly focused on detecting semantically related words rather than exact synonyms [...] [Yu et al., 2002]

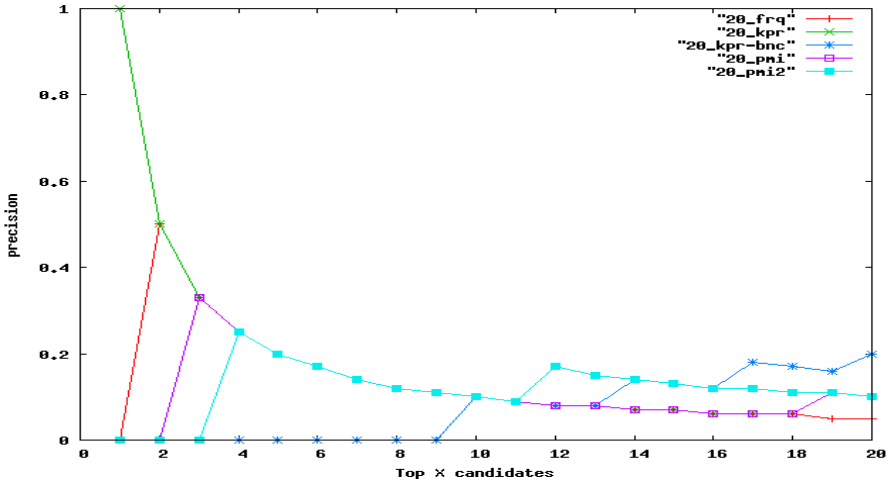
The above quote from the developers of the SGPE system which extracts gene and protein synonyms from biomedical text stresses that this is indeed also the case for the domain of Biomedicine, and the problem of detecting synonymy in the strict sense of the term will become apparent in the following discussions.

Synonyms of "glucose" In the analysis of this first of the five synonymy experiments we will take a detailed look at the information in the test database in which all the experimental data is stored and from which all precision graphs are generated. Table 54

Table 54: Ranking of “glucose” synonyms

candidate (correct?)	frq	kpr	...	kpr-bnc	pmi
blood sugar (no)	121 (#1)	9 (#3)	...	9.00 (#2)	1.72 (#42)
dextrose (yes)	*83 (#2)*	*10 (#1)*	...	2.91 (#14)	1.82 (#56)
sugar (no)	72 (#3)	10 (#2)	...	0.60 (#71)	1.29 (#31)
hypoglycemia (no)	31 (#4)	8 (#4)	...	11.54 (#1)	1.74 (#41)
hyperglycemia (no)	29 (#5)	8 (#5)	...	8.00 (#3)	1.80 (#32)
...	...	...	...	...	
d-glucose (yes)	*5 (#33)*	*3 (#19)*	...	3.00 (#10)	2.14 (#8)
corn sugar (yes)	*4 (#49)*	*2 (#36)*	...	2.00 (#17)	1.88 (#30)
dextrose or corn sugar (yes)	*3 (#67)*	*2 (#44)*	...	2.00 (#20)	1.98 (#23)

Figure 28: Synonyms of “glucose” - assorted ranking schemes



illustrates how the candidate synonyms for “glucose” are ranked by a selection of the various schemes. It is observed how three of the four correct candidate synonyms of “glucose”, namely “d-glucose”, “corn sugar” and “dextrose or corn sugar” are both infrequent and associated with a very limited range of KPs in the sample text snippets (3, 2 and 2). Consequently, they rank low by the “frq”, “kpr” and “fkpr” schemes, but higher when the BNC filter is turned on (“kpr-bnc”) or with the pmi-based schemes which are biased towards rare events. Also, the infrequency of the candidate “dextrose or corn sugar” illustrates how system performance (primarily recall) could be improved by adding more advanced morphosyntactic transformations so as to decompose coordinated NPs into their constituent parts.

Figure 28 displays the precision of assorted ranking schemes for synonyms of “glucose”. There are four correct candidates in the sample (three of which are very infrequent as mentioned above), and only one of the ranking schemes returns a correct candidate as its first choice, namely “kpr”. The “pmi2” scheme is the first to identify a

Table 55: Synonyms of "glucose" - top 10 candidates

rank	candidate ("kpr")	judgment	candidate ("frq")	judgment
1	dextrose	1,1,1,1	blood sugar	3,2,2,3
2	sugar	3,1,3,1	dextrose	1,1,1,1
3	blood sugar	3,2,2,3	sugar	3,1,3,1
4	hyperglycemia	3,3,3,2	hypoglycemia	3,3,3,3
5	hypoglycemia	3,3,3,3	hyperglycemia	3,3,3,2
6	diabetes	3,3,3,3	amount	2,3,2,2
7	glycogen	3,3,2,2	insulin	3,3,3,2
8	blood	3,3,3,2	glycogen	3,3,2,2
9	corn syrup	3,3,3,2	simple sugar	3,1,3,2
10	gestational diabetes	3,3,3,3	monosaccharide	2,2,2,3
...	...	...	...	

second synonym as candidate number 12. Table 55 lists the top 10 candidates returned by the "kpr" and "frq" schemes, respectively. While most of the candidates (with the exception of "dextrose") are not synonyms of glucose, they appear to be highly related to this concept in various ways. So related, in fact, that even the experts disagree as to the number of synonyms in the sample (see for example the "sugar" candidate).

At this point it will be helpful to visualize a small fragment of the UMLS sugar ontology (see figure 29) as this may illustrate the relative complexity of the knowledge required on the part of the experts in this (and the following) experiment. Incidentally, the UMLS sugar ontology is slightly simplified as compared to that presented in textbooks on Chemistry[61].

The UMLS Metathesaurus contains no synonyms for "lactose", but the two synonyms, "dextrose" and "D-glucose", for "glucose". Only one of these (dextrose) make it into top 10 in the system ranking, but the ontology fragment explains why candidates like "glycogen", "monosaccharide", "simple sugar" etc. appear among the top 10 candidates. They are not synonyms of "glucose" but hypernyms (for example "monosaccharide" and its synonym "simple sugar") or other hyponyms of carbohydrate. Further down the list of candidates a wide range of these, including "galactose", "glycogen", "malt sugar" and "maltodextrin", appear.

This suggests that the synonymy KPs perhaps are somewhat ambiguous and to some extent overlap with the ISA KPs (more on this ambiguity in section 6.3). Nevertheless, the system also retrieves candidates so closely related to "glucose" that they could be called near-synonyms. For example, the candidate "corn sugar" is indeed a synonym of "glucose", but what about "corn syrup"? Another example of near-synonymy is the candidate "FDG" which ranks fairly low with a frequency of 3 and a KP range of 2. FDG is an acronym for "fluorodeoxyglucose", which is a glucose analogue. This means that "FDG" ($C_6H_{11}FO_5$) is structurally similar to "glucose" ($C_6H_{12}O_6$), but has replaced atoms.

As mentioned in the discussion about the Communicative Theory of Terminology

[61] glucose is, in fact, an "aldohexose" [McMurry, 1998, p441]

Figure 29: UMLS sugar ontology fragment

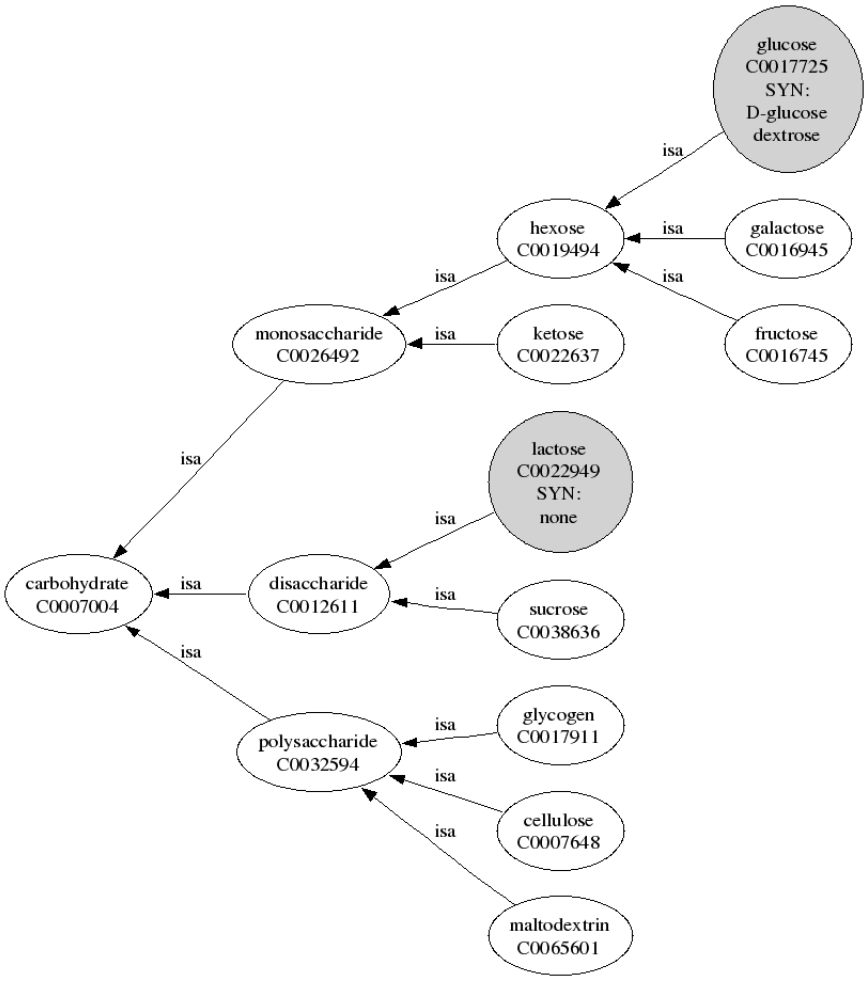
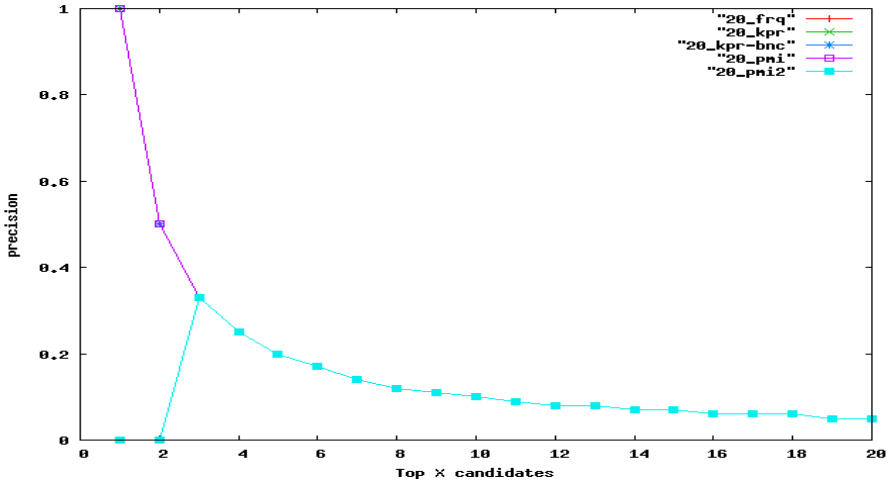


Table 56: Levels of precision in chemical nomenclature

nomenclature	features	example
common name		corn sugar
systematic name	+semantic transparency	glucoseldextrose
chemical formula	+atom number/type	$C_6H_{12}O_6$
IUPAC system	+atomic configuration	6-(hydroxymethyl)oxane-2,3,4,5-tetrol

Figure 30: Synonyms of “lactose” - assorted ranking schemes



(see subsection 2.2.4), [Sager, 1990] argues that the wording of messages in specialized discourse is determined by the three factors of precision, economy and approprioateness. Although a maximum level of precision can easily be attained by using complete and unambiguous definitions of the concepts used in the discourse, the factor of economy will tend to shorten and compress the complete definitions into more ambiguous phrases with the passage of time. In short, appropriateness is the balance between precision and economy which has been achieved over time in a specific discourse community. In the case of Chemistry (which is partly overlapping with the domain of Biomedicine), this balance can be struck with at least four degrees of precision. Table 56 illustrates how these four degrees of precision are realized by four different ways of expressing the concept “corn sugar”. In the example “corn sugar” is the common name for “glucose | dextrose” and is typically used by laymen. While “glucose | dextrose” is the systematic term used by domain experts (and knowledgeable non-experts), these experts often have an even greater need for precision and will use either the chemical formula, indicating the number and type of atoms in the particular molecule, or the IUPAC[62] expression if the configuration of these atoms makes a semantic (or rather chemical) difference. All this to say that as for synonymy Biomedicine is perhaps a unique domain because synonymy can be eliminated by using IUPAC notation.

Synonyms of “lactose” Figure 30 and the complete F-score plot in appendix 8.39.4) illustrate that “lactose” is a special case in that it has only one synonym, namely “milk sugar”. Nearly all schemes manage to identify this as their number one choice. There is one exception, though, “pmi2” fails miserably and ranks “milk sugar” as its third choice. In this case pmi discounting by squaring the observed frequency proved less effective than the unmodified pmi measure.

[62] International Union of Pure and Applied Chemistry

Table 57: Synonyms of "lactose" - top 10 candidates

rank	candidate ("frq")	judgment	candidate ("pmi2")	judgment
1	milk sugar	1,1,1,1	milk	3,3,3,2
2	milk	3,3,3,2	galactose	3,2,3,3
3	lactose intolerance	3,3,3,3	milk sugar	1,1,1,1
4	condition	3,3,3,3	lactose intolerance	3,3,3,3
5	sugar	3,3,1,1	inducer	3,3,3,3
6	lactose-free milk	3,3,3,3	yoghurt	3,3,3,3
7	lactase	3,3,3,2	lactase deficiency	3,3,3,3
8	galactose	3,2,3,3	lactose maldigestion	3,3,3,3
9	iptg	3,3,3,3	iptg	3,3,3,3
10	milk allergy	3,3,3,2	lactose-free milk	3,3,3,3
...	...	...	...	

Table 57 lists the top 10 candidates as ranked by "frq" and "pmi2". The candidates in the table again suggest that synonymy KPs may have something in common with ISA KPs. As can be seen from the sugar ontology in figure 29 "galactose", for example, is a carbohydrate like "lactose" albeit a monosaccharide and not a disaccharide. Interestingly, there may also be an overlap between synonymy KPs and meronymy KPs. Lactose and the two candidates, "milk" and "yoghurt", are meronyms, so they are certainly semantically related, but not synonymous. This again illustrates how using high precision KPs is required to distinguish between instances of closely related relation types, whereas identifying "semantically related" terms can be done through various other measures of distributional similarity as in e.g. [Ananiadou and McNaught, 2006].

Synonyms of "formaldehyde" While figure 31 plots the precisions of assorted ranking algorithms in the task of finding synonyms of "formaldehyde", the figure in appendix 8.39.3 plots their F-scores. Looking at the two graphs reveal that several algorithms are not only highly precise, but also place 75% of all the synonyms in the sample (3 out of 4) among their top 10 candidates. In fact, most of the action takes place in the top 5 candidates and "fkpr-bnc" and "fkpr" are the best ranking schemes. However, "fkpr-bnc" is the first scheme to identify the fourth and final synonym, namely the infrequent "formic aldehyde or methyl aldehyde" (at rank 17). "Kpr-bnc", on the other hand, is a surprisingly poor choice. Table 58 has a breakdown of precision and recall scores, and table 59 lists the top 5 candidates for "kpr-bnc" versus "fkpr".

In conclusion, identifying synonyms of "formaldehyde" and also "lactose" appears to be a considerably easier task than "glucose".

Synonyms of "progesterone" Figure 32 reveals "progesterone" to be the most challenging of all five synonymy experiments. As can be seen from table 60, the only schemes which rank correct candidates among their top 10 are "pmi2" and "pmi". All other schemes perform even worse.

Table 61 lists the top 10 candidate synonyms as ranked by "frq" versus "pmi2". The

Figure 31: Synonyms of “formaldehyde” - assorted ranking schemes

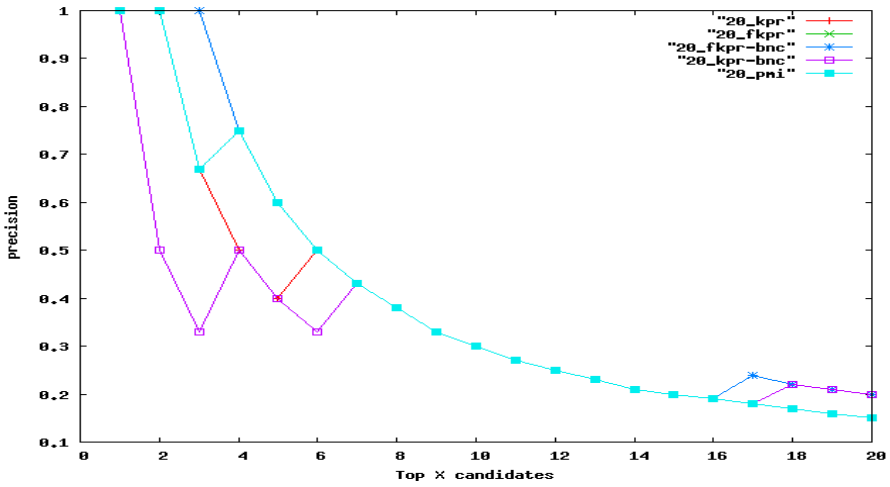


Table 58: Synonyms of “formaldehyde”: precision and recall of sample-based schemes

scheme	top 3 (P)	top 5 (P)	top 10 (P)		top 3 (R)	top 5 (R)	top 10 (R)
kpr	0.67	0.40	0.30		0.50	0.50	0.75
kpr_bnc	0.33	0.40	0.30		0.25	0.50	0.75
fkpr_bnc	1.00	0.60	0.30		0.75	0.75	0.75
frq	0.67	0.40	0.30		0.50	0.50	0.75
fkpr	1.00	0.60	0.30		0.75	0.75	0.75
pmi	0.67	0.60	0.30		0.50	0.75	0.75
pmi2	0.67	0.40	0.30		0.50	0.50	0.75

Table 59: Synonyms of “formaldehyde” - top 5 candidates

rank	candidate (“kpr-bnc”)	judgment	candidate (“fkpr”)	judgment
1	methanal	1,3,1,1	formalin	1,2,1,1
2	paraformaldehyde	3,3,3,3	methanal, methylene oxide	1,1,1,1
3	embalming fluid	2,3,3,2	methanal	1,3,1,1
4	ureaform	3,3,3,3	bakelite	3,3,3,3
5	methanal, methylene oxide	1,1,1,1	paraformaldehyde	3,3,3,3
...	...	...	...	

Figure 32: Synonyms of “progesterone” - assorted ranking schemes

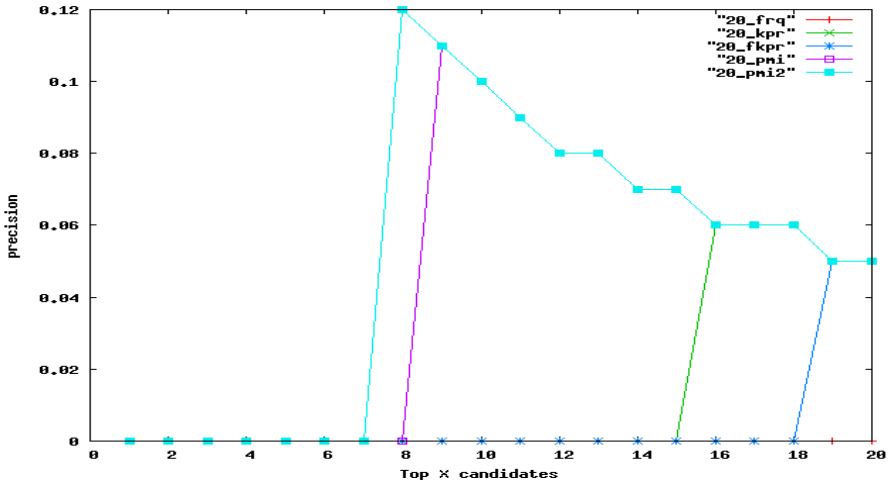


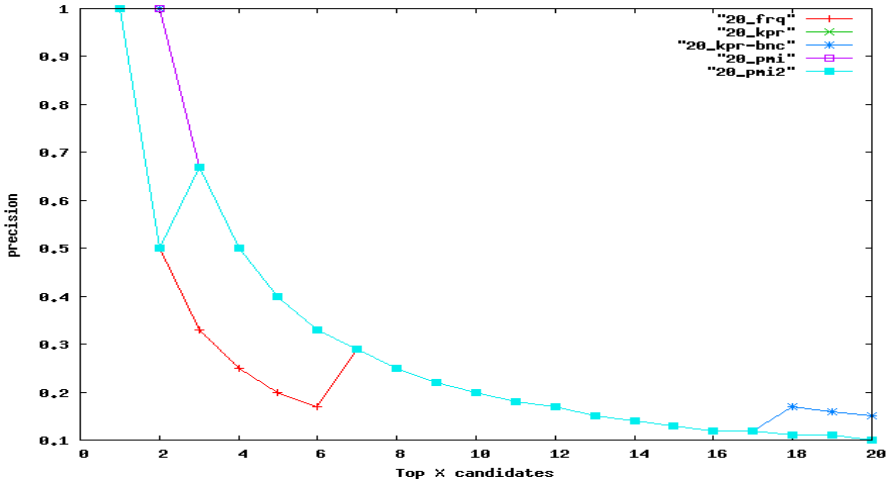
Table 60: Synonyms of “progesterone”: precision of sample-based schemes

scheme	top 3	top 5	top 10
kpr	0.00	0.00	0.00
kpr_bnc	0.00	0.00	0.00
fkpr_bnc	0.00	0.00	0.00
frq	0.00	0.00	0.00
fkpr	0.00	0.00	0.00
pmi	0.00	0.00	0.10
pmi2	0.00	0.00	0.10

Table 61: Synonyms of “progesterone” - top 10 candidates

rank	candidate (“pmi2”)	judgment	candidate (“frq”)	judgment
1	medroxyprogesterone acetate	3,2,3,3	progestin	1,2,1,3
2	progestogens	1,2,1,3	hormone	3,3,3,3
3	progestins	1,2,1,3	progestins	1,2,1,3
4	treatment	3,3,3,3	estrogen	3,3,3,3
5	progestin	1,2,1,3	estrogen dominance	3,3,3,3
6	estradiol	3,3,3,3	new osteoblasts	3,3,3,3
7	hormone	3,3,3,3	molecule	2,3,2,2
8	progestogen	1,2,1,1	prometrium	3,3,3,3
9	prometrium	3,3,3,3	provera	3,3,3,3
10	estrogen dominance	3,3,3,3	progesterone substance	3,3,2,3
...	...	...	...	

Figure 33: Synonyms of “vitamin c” - assorted ranking schemes



candidates in the table reveal two things. First of all, there is considerably disagreement among the experts as to the status of “progestin(s)” as a synonym of progesterone. Secondly, it would appear that a lemmatization module might have improved system performance in this case. While the candidate “progestogen” is considered correct by the experts, its plural form, “progestogens”, which is ranked second by “pmi2” ends up with an average correctness score exceeding the 1.50 threshold. The lack of lemmatization may have confused an expert here, thus reducing the apparent performance of the system.

Quite a few correct hypernyms are retrieved by the system, for example “molecule” and “hormone”. Again this indicates an overlap between synonymy and ISA KPs. As for “progestin(s)” this is indeed a near-synonym of “progesterone”. Progestins are synthetic progestogens, whereas progesterone is the only natural progestogen. Chemically speaking, the two terms refer to the same compound/substance, but terminologically speaking, progestin is a cohyponym of progesterone and not a synonym, because a distinctive feature (the nature of their genesis) can be identified. Indeed this semantic closeness is also reflected in the expert judgments of “progestin” (1,2,1,3) and “progestins” (1,2,1,3). Only one of the four experts does not consider these terms synonymous with “progesterone” and one is in doubt. In fact, although progesterone has two synonyms, (“progestogen” and “progestogens”) according to the domain experts, neither of these are among the ones listed as synonyms in the UMLS. The UMLS does consider “progestogen” and “progestin” synonymous, but they are both recorded as *hypernyms* of “progesterone”. These inconsistencies suggest that WWW2REL may also be used to *revise* and not just augment existing ontologies.

Synonyms of “vitamin c” In this experiment almost all ranking schemes perform really well and identify two out of the five synonyms of “vitamin C” among their top 5 candidates (see figure 33 and table 62). The only scheme which performs poorly is the

Table 62: Synonyms of “vitamin C”: precision of sample-based schemes

scheme	top 3	top 5
kpr	0.67	0.40
kpr_bnc	0.67	0.40
fkpr_bnc	0.67	0.40
frq	0.33	0.20
fkpr	0.67	0.40
pmi	0.67	0.40
pmi2	0.67	0.40

Table 63: Synonyms of “vitamin C” - top 10 candidates

rank	candidate (“kpr”)	judgment	candidate (“frq”)	judgment
1	ascorbic acid	1,1,1,1	ascorbic acid	1,1,1,1
2	l-ascorbic acid	1,1,2,1	advocacy arguments	3,3,3,3
3	ascorbate	1,1,3,3	other uses	3,3,3,3
4	vitamin e	3,3,3,3	earlier section	3,3,3,3
5	antioxidant	3,3,3,3	vitamin e	3,3,3,3
6	powerful antioxidant	2,3,2,2	antioxidant	3,3,3,3
7	placebo	3,3,3,3	ester-c	3,3,3,3
8	scurvy	3,3,3,3	l-ascorbic acid	1,1,2,1
9	advocacy arguments	3,3,3,3	powerful scavenger	2,2,2,2
10	other uses	3,3,3,3	fewer cateracts	3,3,3,3
...	...	...	...	

baseline frequency ranking, “frq”. Table 63 has the top 10 candidates and their expert judgments for the schemes “frq” and “kpr”.

The figure in appendix 8.39.1 reveals how additional synonyms are found further down the list, but all synonyms of vitamin C (in the sample) are, in fact, present in table 63. However, due to the lack of NP decomposition “ascorbic acid and ascorbate” and “ascorbic acid or ascorbate” are regarded as two different candidates. The complex NPs, of course, contain two different candidates, but these also occur individually and are identified by WWW2REL. However, this is a recall issue, and terminologists using the system will presumably give priority too high precision.

Overall, there are many near-synonyms in the lists of candidates proposed by the system (and a few hypernyms like “antioxidant”[63]). Further down the list, for example, we find candidates like “ascorbate” and “sodium ascorbate”. Neither of these are synonyms of “vitamin C/ascorbic acid”, because they are salts rather than acids. However, a candidate like “ascorbate or ascorbic acid” is truly difficult to assess the correctness of, because it is part right (ascorbic acid) and part wrong (ascorbate). Cases like these

[63]the mysterious candidate “powerful scavenger” is a fragment of the NP “powerful scavenger of free radicals”, which does not provide a synonym for “vitamin C” but describes its function

Table 64: Haloperidol ISA X - top 10 candidates

rank	candidate ("kpr-head-bnc")	judgment	candidate ("kpr-head")
1	antipsychotic	1,1,1,1	antipsychotic
2	typical antipsychotic	1,1,2,1	typical antipsychotic
3	conventional antipsychotic	1,1,2,1	conventional antipsychotic
4	typical antipsychotics	1,1,1,1	typical antipsychotics
5	antipsychotics	1,1,1,1	antipsychotics
6	conventional antipsychotics	1,1,2,1	conventional antipsychotics
7	older antipsychotics	1,1,1,2	older antipsychotics
8	traditional antipsychotics	1,1,2,1	traditional antipsychotics
9	first generation antipsychotics	1,1,1,1	first generation antipsychotics
10	classical antipsychotics	1,1,1,1	classical antipsychotics
...	...	...	...

may, of course, lower the inter-annotator agreement.

5.5.6 Haloperidol ISA X

Terminological definitions based on classical conceptual analysis (see subsection 2.2.1) are formed by finding the immediate hypernym and the delimiting characteristics of the analysandum (i.e. the target concept). Thus in this subsection the relation extraction system attempts to identify hypernyms of the concept represented by the term "haloperidol". In subsection 5.5.7, however, the system attempts to identify the hyponyms of an input term. Although that experiment concerns the extension rather than the intension of a concept, automatically identifying hyponyms may still be a useful tool to a terminologist surveying a new subdomain.

At this point the reader perhaps wonders how the number of correct hypernym candidates for the antipsychotic, "haloperidol", can possibly be a huge 57. The explanation can be found in figure 38 (section 6.4) and in table 64. This table lists the top 10 candidates when ranked by KP range and grouping by head with and without the BNC filter. Both schemes rank the two heads, "antipsychotic" and "antipsychotics", as number one and two, respectively. The reason there are so many correct candidates in the sample, is mainly due to the great diversity of adjectival modifiers like "typical", "classical", "conventional" and so on. It is, of course, also because no lemmatization is performed so that "antipsychotic" and "antipsychotics" are counted as two different heads. Finally, haloperidol, in fact, has a range of characteristic properties and thus multiple hypernyms as illustrated in figure 38.

If desired by the user the linguistic variation can easily be reduced by displaying only the candidate heads and ignoring all NPs grouped under the individual heads as is done in table 65. This table also reveals that there are significant differences between the two schemes, "kpr-head" and "kpr-head-bnc". When the BNC filter is turned on semantically vague heads like "drug(s)", "agent" and "treatment" disappear from the list in favor of "antiemetic" and "butyrophenones".

Figure 34 and table 66 compare the precision of all ranking schemes. As indicated

Table 65: Haloperidol ISA X - top 10 heads

rank	candidate (“kpr-head”)	candidate (“kpr-head-bnc”)
1	antipsychotic	antipsychotic
2	antipsychotics	antipsychotics
3	drugs	antiemetic
4	agents	neuroleptics
5	drug	high-potency
6	treatment	butyrophenones
7	neuroleptics	neuroleptic
8	neuroleptic	butyrophenone
9	medication	phenothiazines
10	antiemetic	medications
...	...	...

Figure 34: Haloperidol ISA X - assorted ranking schemes

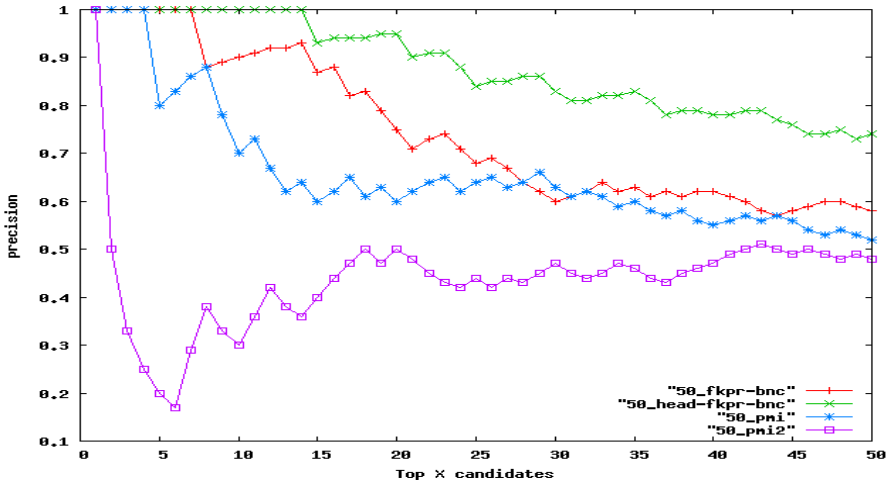
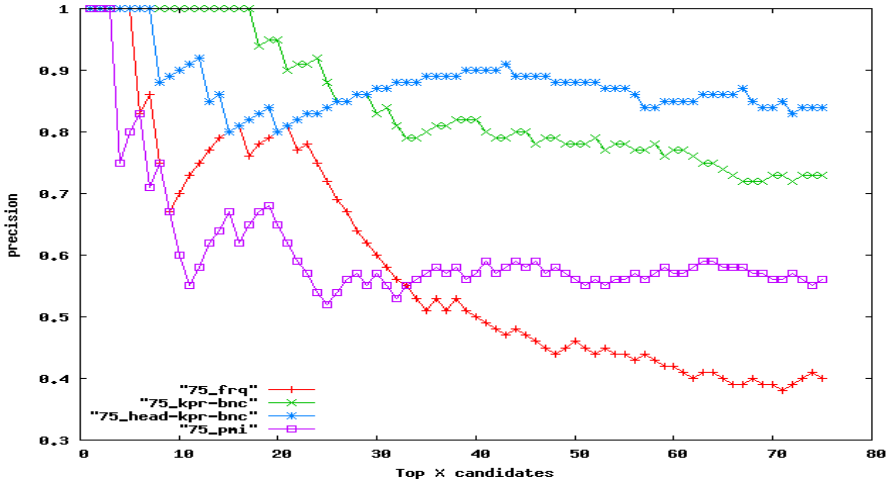


Table 66: Haloperidol ISA X: precision of sample-based schemes

scheme	top 5	top 10	top 25		top5	top10	top25
fkpr	0.80	0.90	0.76	fkpr-head	0.80	0.70	0.76
fkpr-bnc	1.00	0.90	0.68	fkpr-bnc-head	1.00	1.00	0.84
frq	1.00	0.70	0.64	frq-head	1.00	0.80	0.76
kpr	1.00	0.90	0.72	kpr-head	1.00	1.00	0.80
kpr-bnc	1.00	0.90	0.84	kpr-bnc-head	1.00	1.00	0.84
pmi	0.80	0.70	0.64	frq-bnc-head	1.00	1.00	0.80
pmi2	0.20	0.30	0.44				

Figure 35: “X ISA antipsychotic” - assorted ranking schemes



by the scores in the table and the plots (including those in appendix 8.41) performance both in terms of precision, but also recall, is greatly boosted by grouping candidates by their NP head. Furthermore, “(f)kpr-bnc-head” are the two best schemes, whereas “pmi2” performs really poorly. Again, this probably reflects the fact that the hypernyms of “haloperidol” are relatively frequent, and as mentioned before pmi does favor rare events.

Finally, the experiment underscores the difficulty of assessing whether a proposed hypernym is too general or not. When writing analytical definitions the objective is always to identify the closest superordinate concept (genus proximum) and a single distinctive feature. Thus two experts consider the candidates “drugs” and “prescription drugs” too vague, while three experts think the same of the candidate “typical drugs”. While such general hypernyms would clearly constitute a useful bit of knowledge for many non-experts (myself included before writing this thesis), the target user profile must be kept in mind here. For terminologists, who work bottom-up when structuring the knowledge of a domain, a hypernym like “drug” will probably be too general and be of little use when building more specific subontologies level by level bottom-up.

5.5.7 X ISA antipsychotic

Figure 35, reveals that the BNC discounting and head grouping heuristics really make a difference to the precision in the task of identifying antipsychotics in a sample of text snippets from the WWW. The baseline “frq” scheme and the pmi-based scheme, on the other hand, rapidly lose ground. Although precision is not, and could not be expected to be, quite as high as in the “haloperidol ISA X” experiment, it is still remarkably high considering the noisy source of knowledge. Grouping also boosts overall performance in terms of recall/F-score (see appendix 8.40). By the top 100 candidates the F-score of “head-kpr-bnc” exceeds 0.80.

Table 67: “X ISA antipsychotic”: precision of sample-based schemes

scheme	top 10	top 25	top 50		top 10	top 25	top 50
fkpr-bnc	0.80	0.72	0.50	fkpr-bnc-head	0.90	0.84	0.84
fkpr	0.80	0.80	0.58	fkpr-head	0.90	0.80	0.74
frq	0.70	0.72	0.46	frq-head	0.90	0.64	0.64
kpr-bnc	1.00	0.88	0.78	kpr-bnc-head	0.90	0.84	0.88
kpr	0.90	0.76	0.58	kpr-head	0.90	0.84	0.74
pmi	0.60	0.52	0.56	frq-bnc-head	0.90	0.84	0.82
pmi2	0.30	0.48	0.58				

Table 68: “X ISA antipsychotic” - top 10 candidates

rank	candidate (“pmi2”)	judgment	candidate (“kpr-bnc”)	judgment
1	chlorpromazine	1,1,1,1	clozapine	1,1,1,1
2	medications	2,2,2,3	risperidone	1,1,1,1
3	these drugs	2,2,3,3,	olanzapine	1,1,1,1
4	agent	2,2,3,3	haloperidol	1,1,1,1
5	drug	2,2,2,2	quetiapine	1,1,2,1
6	high doses	2,2,3,3	aripiprazole	1,1,1,1
7	mglu23 receptor agonists	3,2,2,2	thioridazine	1,1,1,1
8	agents	2,2,3,3	zyprexa	1,1,1,1
9	perphenazine	1,1,1,1	seroquel	1,1,1,1
10	zeldox	1,1,1,1	risperdal	1,1,1,1
...	...	...	...	

The precision figures in table 67 reveal "head-kpr-bnc" and "kpr-bnc" to be the two best ranking schemes. Finally, table 68 displays the top 10 candidates when ranked by the worst performing "pmi2" and best performing "kpr-bnc" schemes, respectively.

5.5.8 Conclusion

This subsection discusses the overall performance of WWW2REL, first in terms of precision and secondly in terms of F-score. An overall best instance ranking scheme is identified, a default system setting is suggested and a tentative inter-system performance comparison is attempted.

Combining the results of the analyses presented in sections 5.4 and 5.5, it would appear that the two heuristics of head grouping and BNC discounting applied to the "kpr" or "fkpr" ranking scheme constitute the most effective strategy for optimizing the precision of both top 10, top 25 and even more relation instances. In line with the hypothesis from section 5.3 the highest precision rates were achieved in the ISA experiments. ISA relation instances appear to be the easiest to retrieve and rank automatically. As expected the "X induces {vomiting | emesis}" experiments proved difficult, but using "emesis" boosted precision as hypothesized. The causal relations were relatively difficult as well, especially the "selenium" experiment, but as described in subsection 5.2.5 the "selenium" experiment was characterized by significant inter-annotator disagreement and is arguably an anomalous case. Finally, identifying at least one synonym with high precision proved feasible in all experiments (except for "progesterone").

It was hypothesized that the "pmi" schemes would not be worth the effort, but when identifying things which induce emesis or vomiting or synonyms of progesterone "pmi", in fact, proved very useful. Finally, the use of passive KPs (the "pkpr" scheme) did have a positive effect on precision in the "aspirin" and "{drugs} induce emesis" experiments. However, the range of these patterns appears to be too small to make a lasting improvement in experiments with a large search space. One way of improving this ranking scheme would be to attach greater weight to these, presumably, highly reliable patterns than to the regular, active constructions.

Complete F-score plots of all experiments can be seen in the appendices, but table 69 computes the actual system performance as the proportion between the maximum possible F-score and the actual F score achieved by the best ranking scheme in each of the eleven experiments. The maximum possible F-scores are computed relative to a fixed threshold value for the number of candidate instances taken into consideration. Deciding on a threshold value will always be a somewhat arbitrary exercise, but it seems unreasonable that a terminologist using WWW2REL should have to go through more than 50 candidate instances for any relation type. Given a uniform cut-off point of the top 50 instances, the maximum possible F-score in the "aspirin induces X" experiment is $F_{max.pos.}(50) = 2 * \frac{\frac{50}{50} * \frac{50}{148}}{\frac{50}{50} + \frac{50}{148}} \approx 0.51$. In this setting, system performance when selecting the best ranking scheme (fkpr with both heuristics switched on) is thus 81.2% relative to perfect performance.

For synonymy, the maximum possible F-scores drop drastically when raising the threshold from the top 10 to top 50 instances, whereas the exact opposite is the case for

Table 69: Summary chart - overall system performance (F-scores)

experiment	$\frac{F(10)}{F_{max.pos.}(10)}$	$F_{max.pos.}(10)$	$\frac{F(50)}{F_{max.pos.}(50)}$	$F_{max.pos.}(50)$
aspirin induces X	100%	0.13	81.2%	0.51
selenium may_prevent X	60.0%	0.33	38.0%	1.00
X ISA antipsychotic	100%	0.20	86.9%	0.73
haloperidol ISA X	100%	0.30	78.1%	0.94
X induces vomiting	41.4%	0.29	31.6%	0.92
X induces emesis	80.1%	0.57	96.0%	0.67
vitamin c	43.5%	0.67	82.4%	0.18
lactose	100%	0.18	100%	0.05
glucose	24.5%	0.57	100%	0.15
formaldehyde	75.3%	0.57	100%	0.16
progesterone	51.1%	0.33	100%	0.08
OVERALL AVERAGE	70.5%		81.3%	

Table 70: Summary chart - overall best ranking scheme

ave. performance	scheme	ave. performance	scheme
Top 10 instances		Top 50 instances	
69.6%	kpr-bnc-head	65.6%	fkpr-bnc-head
68.9%	kpr-head	64.6%	kpr-bnc-head
65.4%	frq-bnc-head	60.7%	frq-bnc-head
63.9%	fkpr-bnc-head	59.1%	kpr-bnc
58.9%	kpr-bnc	57.2%	kpr-head

ISA and the two causal relations. In the former case, it is easy to achieve 100% system performance (in terms of F-score) by raising the threshold value, but as illustrated by the plots in subsection 5.5.5, this comes at a heavy cost to precision and is probably not desirable in practical terminology work.

To make WWW2REL truly useful to terminologists, a default setting should be provided. Table 70 identifies the overall best ranking scheme in terms of average performance at the two threshold levels across all experiments. It suggests that the default ranking scheme should be "kpr" with both heuristics turned on. As discussed in section 6.5 certain domain characteristics may dictate a change to this default setting, however. Other bits of domain knowledge, such as the perceived size of the search space for the target relation type may also affect e.g. the threshold level.

Comparing the precision and F scores achieved by WWW2REL with those reported for other pattern-based relation extraction systems, e.g. those listed in table 5, section 3.4, is non-trivial at best. For a comparison to be 100% fair, systems should be applied to the exact same data in which instances of the target relation types should be manually annotated by domain experts to provide a common gold standard. Since no such gold standard appears to exist for the relevant relation types in a corpus of Google

Table 71: Inter-system precision comparison

	Espresso	WWW2REL	WWW2REL	Espresso	WWW2REL
relation	ISA	ISA-hypo	ISA-hyper	production	induces
instances	200	50 -> 100	50 -> 100	196	50 -> 100
precision	85%	88% -> 75%	78% -> 55%	72.5%	72% -> 57%

text snippets and WWW2REL is designed to operate on exactly such data, a direct performance comparison is not possible.

Nevertheless, precision ranges obtained by different systems but for the same relation types should be comparable keeping in mind that some data sources (e.g. uncategorized WWW text snippets) pose a greater challenge than others (e.g. corpora of biomedical abstracts) and that precision rates are often reported without reference to the number of instances or the level of recall. However, in the case of the Espresso system [Pantel and Pennacchiotti, 2006], the number of instances is given, and thus table 71 contains a tentative comparison of its precision scores with those of WWW2REL.

Although the text types differ, a Chemistry textbook in the case of Espresso and WWW text snippets in WWW2REL, the two systems perform at comparable precision levels. Precision levels in WWW2REL do deteriorate more rapidly as the number of instances is increased, but this is hardly surprising given the noisy nature of uncategorized text fragments on the Internet.

In conclusion, it would appear that the precision rates achieved are quite satisfactory given the special context of the WWW2REL implementation, namely a noisy text source, the lack of sophisticated NLP techniques, the relative simplicity of the ranking schemes and the domain independence of the system.

6 Recall and portability

Having described and discussed the implementation of WWW2REL and examined its performance from two different perpectives, namely the precision of the various ranking schemes and the precision in the individual experiments, it is now time to investigate other parameters which affect the quality of the system. The first of these is the text snippet sample size which was arbitrarily set to 100 snippets per <term,KP> query. Section 6.1 thus examines the correlation between ranking scheme performance on the snippets sample versus the WWW as a whole. Secondly, the correlation between the BNC frequencies of all instance heads will be compared to the frequencies of these heads on the WWW (i.e. Google hit counts) so as to determine whether using web frequencies rather than BNC frequencies would have made a big difference to the discounting heuristic (section 6.2).

Another important topic is a detailed investigation of the actual performance of *individual* knowledge patterns in order to determine which KPs are the most useful ones (see section 6.3). The issue of recall, and the related issue of portability, is further pursued in the two final sections of the chapter. Specifically, the following questions will be addressed.

1. What is the recall versus the UMLS?
2. What is the proportion of new knowledge retrieved?
3. How domain-independent are the WWW2REL KPs?
4. How domain-independent are the ranking schemes and heuristics?

Section 6.4 attempts to answer the first two questions, but also discusses how measuring system recall versus the UMLS Metathesaurus is not straight-forward. Finally, section 6.5 contains a few tests of the system on another domain (IT) to examine the system portability questions raised above.

6.1 Correlation between snippets sample and WWW

This section will compare the performance of the best sample-based ranking schemes with their WWW-based extensions so as to explore in what way additional data would have affected the performance of these schemes. Of course, if candidates had been extracted from a much larger text sample results might have been different. What is examined in this section, however, is exclusively the correlation between the ranking of the correct candidates when limiting analysis to the existing sample and when using hit counts from the entire WWW.

The expectation is that candidates will be associated with a wider range of KPs and have a higher KP co-occurrence frequency when looking on the entire WWW, but that the sample-based and WWW-based schemes will largely correlate positively with eachother. In other words, the graphs presented in this section are meant to elucidate what the "upper performance bound" would be if we had but time and hit counts enough, so to speak. Is the arbitrary sample size of top 100 text snippets per <term,KP> query inadequate or would additional snippets not have made much of a difference? The reason why the KP range of a particular candidate may be higher when looking on the entire WWW than it is in the snippets sample is that for *some* KPs *some* candidates may *by coincidence* not have been among the top 100 snippets returned by Google, although they do in fact occur with the these KPs on the WWW.

To minimize the time-consuming Google querying process and to keep the graphs as readable as possible, only the overall best ranking scheme, namely "(f)kpr-bnc" will be analyzed and only the experiments with many correct candidates, namely the causal relations and ISA relations, will be examined.

As figure 36 shows, there is a clear and positive correlation between the sample-based and WWW-based performances in all four experiments. When computing KP ranges on the WWW, performance is boosted significantly for "aspirin induces X" in the short run and moderately in the longer run. For the "selenium may_prevent X" the inverse correlation is observed, namely a moderate precision boost in the short run and a more significant one in the longer run. The reason why the positive effect of the WWW is greater for selenium in the longer run is presumably that singletons were allowed in this experiment and, in fact, constituted about half of all candidates. On the WWW these singletons, of course, occur much more often and the "kpr-bnc" scheme is able to perform more accurately based on the extra statistics. The positive effect of

Figure 36: Correlation between WWW and sample-based precision

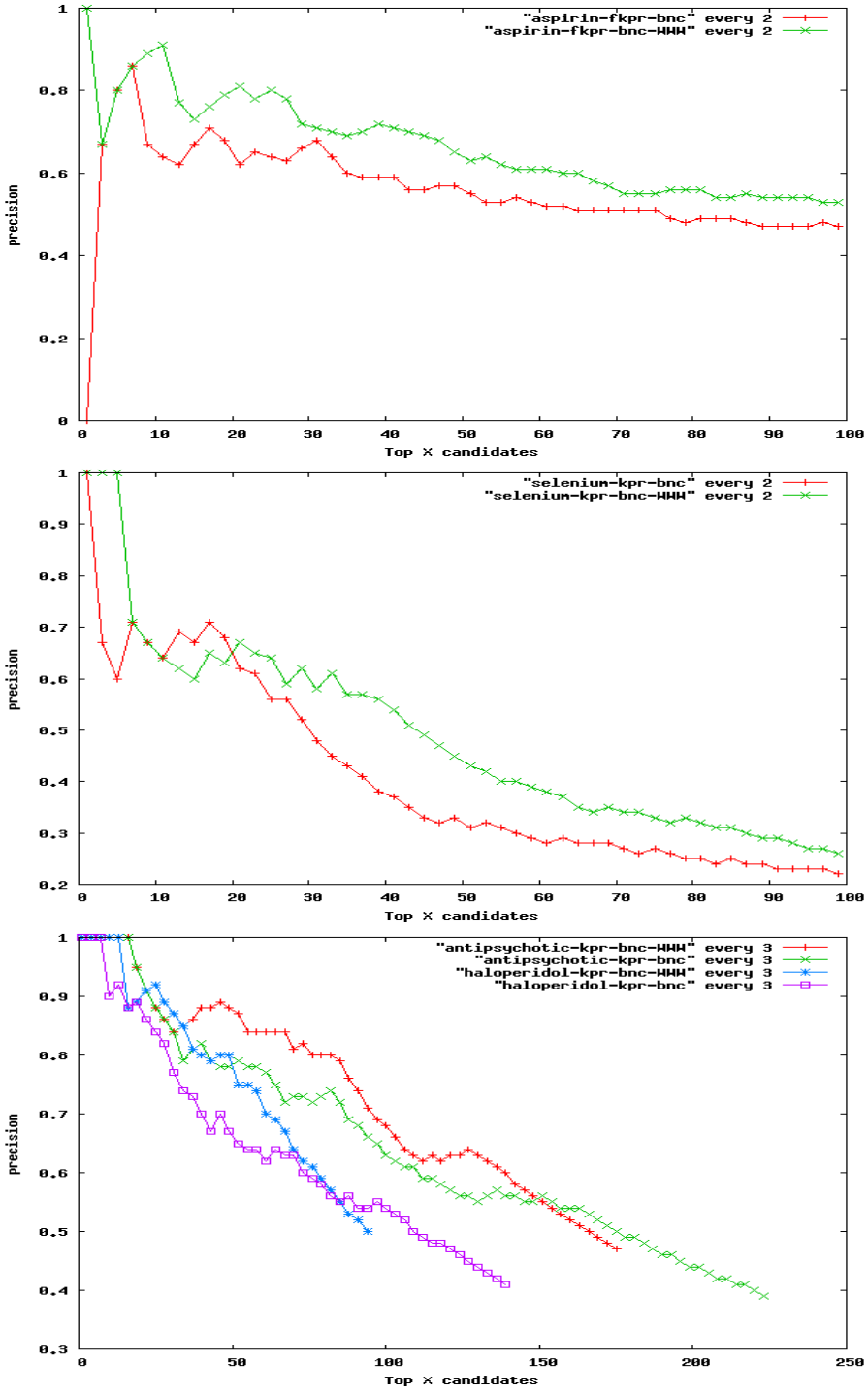
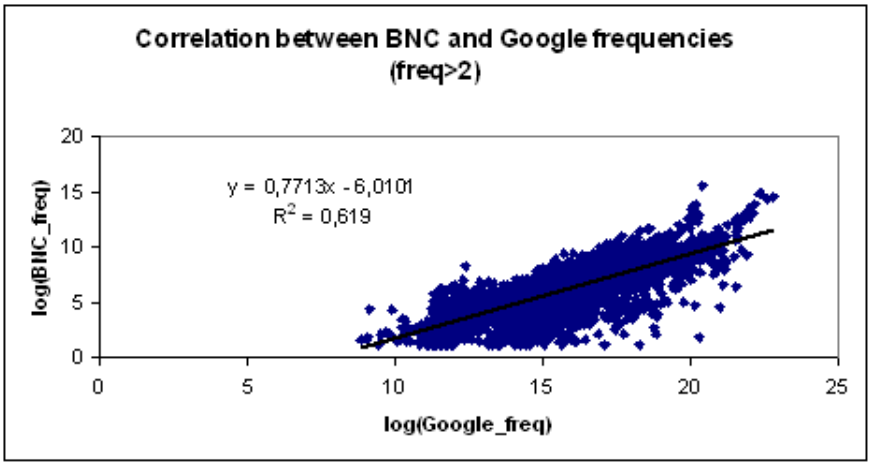


Figure 37: Correlation between BNC and WWW unigram frequencies



using the entire WWW when ranking is also quite conspicuous for the ISA experiment, both when looking for hypernyms of "haloperidol" and hyponyms of "antipsychotics".

In conclusion, there seems to be a high degree of correlation between the sample-based and WWW-based performances of the selected schemes. While the upper performance bound is not dramatically higher than the sample-based performance, extending the sample sizes from 100 snippets per query to, perhaps, the double or triple might still be well advised.

6.2 Correlation between BNC and WWW

The relation instance ranking experiments in section 5.4 indicated that a BNC-based discounting heuristic has a positive impact on system precision. Using the comprehensive ngram statistics recently made public by Google[64] would increase the portability of WWW2REL since the BNC is not freely available. However, instead of exploring whether the Google ngram statistics might have improved the efficiency of the discounting heuristic, this section will examine to what extent the BNC frequencies currently used by the system, in fact, correlate with the corresponding Google hit counts. Thus figure 37 plots the logarithmized BNC frequencies used by WWW2REL against their logarithmized Google hit counts. Not surprisingly, the Google hit counts are much higher than the BNC frequencies, but the plot does indicate a clear correlation between the two. It is thus unlikely that using Google hit counts (or the new Google ngram statistics) would have dramatically changed the effect of the discounting heuristic.

In fact, a possible benefit of using BNC frequencies rather than Google ngram statistics as an indicator of termhood is that all text in the BNC was written before 1994. Since the terminology of many specialized domains, especially the test case of Biomedicine, changes rapidly, many terms would not be found in the BNC but

[64] distributed by the Linguistic Data Consortium (LDC)

surely be found on the WWW. In this way basing the discounting on Google hit counts might penalize correct candidates which would not have been penalized when basing the discounting on the BNC. On the other hand, using Google ngram statistics would allow discounting not just of the instance head but of the complete instance ngram. It is doubtful whether this would boost system precision, but it is an interesting question to be further explored in future work (section 7.2).

6.3 Precision and recall of individual knowledge patterns

Based on the expert judgments, it is now possible to compute the actual precision[65] of the individual knowledge patterns discovered for the four relation types. Actual precision scores are computed by a formula similar to the one displayed in section 3.2. In other words, the precision of a pattern is the number of times this pattern returns a correct relation instance[66] divided by the total number of instances it returns (whether correct or incorrect). Recall, on the other hand, is the number of correct instances returned by the pattern divided by the total number of correct instances which occur in the sample and could possibly be returned by the pattern (see appendix 8.25 for details).

Generally speaking, expert uncertainty or disagreement as well as incorrect knowledge found on the WWW make the performance scores presented in this section somewhat underestimated. While it is not completely "fair" so to speak, these factors will affect negatively the precision scores of the individual knowledge patterns. Finally, by requiring the average correctness level to be 1.50 or below, we consider invalid many of those cases where the pattern *does* instantiate the target relation but the extracted argument is considered too vague or fuzzy (i.e. the judgment "2" has been used). This also "unfairly" reduces the performance scores of the individual KPs, but it is a reasonable requirement for a system which is to be used for the augmentation of *terminological* ontologies, although it may be too strict when applying the system to the expansion of *general* ontologies.

6.3.1 "Induces" patterns

Table 72 lists a few examples of the KPs discovered for the UMLS "induces" relation. The patterns are ranked by F-score, but the table also lists the precision and recall scores of the individual patterns. The complete list can be found in appendix 8.36. Patterns marked by "NA" did not retrieve any instances from the WWW and it was thus impossible to determine their precision or recall levels.

The pattern "overdose can cause" is an example of a domain specific KP for the "induces" relation. It has perfect precision but retrieves only a few instances. The pattern "may aggravate" is another example of a highly reliable, but low recall pattern. Reports on the precision and recall of individual KPs are hard to find in the literature, as researchers normally report on the performance of *systems* applying a host of patterns to solve a specific task. However, the relatively high precision figures for "may cause" are, in fact, corroborated by the findings in [Barrière, 2001, p146] who lists "cause" as

[65] as opposed to the crude measure of precision computed in subsection 4.2.4

[66] defined as having an average correctness score of 1.50 or below

Table 72: Precision and recall of "aspirin <induces> X" patterns (expert judgments)

Pattern	F-score	precision	recall (of 125)
may cause	0.20	0.69	0.11
can cause	0.18	0.58	0.11
can lead to	0.17	0.46	0.11
can induce	0.16	0.78	0.09
induces	0.13	0.55	0.07
may induce	0.12	0.79	0.07
promotes	0.12	0.63	0.07
leads to	0.12	0.37	0.07
to induce	0.11	0.85	0.06
...	...	...	...
overdose can cause	0.04	1.00	0.02
may aggravate	0.04	1.00	0.02

one of a number of noiseless patterns in a manually annotated sample. Also, studying the automatic acquisition of causality markers [Girju and Moldovan, 2002] find that the verb "induce" has a low degree of ambiguity and a high frequency whereas "develop" has a high ambiguity and a high frequency. While these two findings fit with the results presented in appendix 8.36 and tables 72 and 73 ("induce" VPs generally have a high precision), many other markers found by [Girju and Moldovan, 2002] are not found in the present experiment and vice versa, presumably due to the domain specific nature of the term pairs employed during KP dicovery in this thesis.

The two tables 73 and 74 compare which KPs perform the best when using the synonym "emesis" versus "vomiting" and when ignoring how the KPs fare in the "aspirin induces x" experiment. Interestingly, there are far fewer patterns used with "emesis" (22) than with "vomiting" (43), but more significantly "emesis" tends to collocate with patterns containing the verb "induce". There are zero KPs containing this verb among the top 10 best performing patterns for "vomiting", but six among the top 10 best patterns for "emesis". "Vomiting", on the other hand, collocates with stylistically informal features like the relative pronoun"that". This illustrates that features of academic writing tend to occur in clusters and how one can narrow the focus of WWW queries to this text genre by using not just technical synonyms like "emesis", but also technical, or perhaps domain specific, KPs like "induce". Section 6.5 features a small pilot study which attempts to assess the domain specificity of the KPs discussed in the present section.

6.3.2 "May_prevent" patterns

Table 75 lists a few examples of the KPs discovered for the UMLS "may_prevent" relation. Again, patterns are ranked by F-score, but the table also lists the precision and recall scores. The complete list can be found in appendix 8.37.

Generally speaking, the precision (and also F-scores) of "may_prevent" patterns is lower than for "induces" patterns. This presumably reflects the fact that there was

Table 73: Precision and recall of "X <induces> emesis" patterns (expert judgments)

Pattern	F-score	precision	recall (of 25)
induces	0.27	0.88	0.16
causes	0.27	0.79	0.16
to induce	0.23	0.44	0.16
induced	0.21	1.00	0.12
will induce	0.21	0.89	0.12
causing	0.15	1.00	0.08
can induce	0.15	0.82	0.08
may cause	0.14	0.71	0.08
cause	0.14	0.62	0.08
to cause	0.14	0.44	0.08
for inducing	0.13	0.38	0.08
...	...	...	...

Table 74: Precision and recall of "X <induces> vomiting" patterns (expert judgments)

Pattern	F-score	precision	recall (of 59)
overdose include	0.23	0.77	0.14
poisoning include	0.20	0.69	0.12
produces	0.13	0.83	0.07
that can cause	0.13	0.17	0.10
causes	0.12	0.50	0.07
which causes	0.12	0.41	0.07
can also cause	0.12	0.14	0.10
to produce	0.11	0.34	0.07
can cause	0.11	0.17	0.08
does not cause	0.09	0.52	0.05
to cause	0.09	0.40	0.05
...	...	...	...

Table 75: Precision and recall of "may_prevent" patterns (expert judgments)

Pattern	F-score	precision	recall (of 50)
helps prevent	0.16	0.42	0.10
prevents	0.16	0.35	0.10
protects against	0.14	0.24	0.10
decreased	0.14	0.24	0.10
reduced	0.13	0.12	0.14
could reduce	0.12	0.28	0.08
significantly reduces	0.11	0.59	0.06
in preventing	0.11	0.53	0.06
...	...	...	...
cuts	0.08	0.62	0.04
...	...	...	...
to combat	0.04	0.50	0.02

Table 76: Recall and precision per template

template	pattern	recall	precision
a	<hypernym_singular> {KP} <hyponym>	69% (100/145)	48% (100/208)
b	<hypernym_plural> {KP} <hyponym>	48% (70/145)	69% (70/101)
c	<hyponym> {KP} <hypernym_singular>	49% (71/145)	39% (71/183)
d	<hyponym> {KP} <hypernym_plural>	38% (55/145)	47% (55/118)

considerably inter-annotator disagreement in this experiment. Again, some patterns are very reliable, but have low recall (fx. "to combat"). Also, it seems that verbs in the generic, present tense have a higher precision than verbs in the past tense, but this tendency is not clear. Patterns containing the verb "decrease" are also mentioned in a study by [Marshman and L'Homme, 2006] who observe that this causal marker has two non-causal senses out of a total of three senses. In other words, it can be rather noisy as is also indicated by the precision scores for this marker in the table and the appendix.

6.3.3 ISA patterns

Table 76 lists the precision and recall computed for each ISA template. Although the statistics are somewhat meagre with just two input terms, haloperidol and antipsychotic(s), the figures in the table suggest that the most promising template, in terms of recall and for this particular subdomain, is template "a". In other words, constructions where the hypernym occurs in the singular followed by a KP and the hyponym, for example "An antipsychotic <called> haloperidol". Template "d" has the lowest recall, i.e. constructions where the hyponym is followed by a KP and then the hypernym in the plural, for example "risperidone <and other> antipsychotics". In terms of precision, template "b" (for example, "antipsychotics <such as> haloperidol") is far and away the best option with 69%.

Table 77: precision and recall of ISA patterns (per template)

pattern	F	prec	rec	pattern	F	prec	rec
TEMPLATE (a)			(of 100)	TEMPLATE (c)			(of 71)
such as	0.57	0.88	0.42	is an	0.55	0.82	0.41
agents such as	0.35	0.88	0.22	and other	0.50	0.69	0.39
drugs such as	0.31	0.91	0.19	is an effective	0.29	0.57	0.20
efficacy of	0.30	0.55	0.21	as	0.24	0.40	0.17
effect of	0.28	0.54	0.19	is a new	0.22	0.73	0.13
called	0.25	0.75	0.15	is	0.20	0.35	0.14
activity of	0.25	0.56	0.16	as an	0.20	0.27	0.15
effects of	0.25	0.49	0.17	has	0.12	0.38	0.07
action of	0.25	0.49	0.17	an	0.12	0.37	0.07
agent	0.21	0.72	0.12	has an	0.11	0.21	0.07
properties of	0.16	0.44	0.10	another	0.08	0.67	0.04
actions of	0.11	0.30	0.07	exerts its	0.07	0.18	0.04
drug	0.10	0.36	0.06	a new	0.03	0.03	0.03
agents	0.06	0.58	0.03				
drugs	0.06	0.53	0.03				
activity	0.00	0.00	0.00				
TEMPLATE (b)			(of 70)	TEMPLATE (d)			(of 55)
like	0.70	0.89	0.57	and other	0.66	0.73	0.60
such as	0.68	0.87	0.56	or other	0.51	0.70	0.40
including	0.45	0.94	0.30	as	0.26	0.46	0.18
include	0.33	0.69	0.21	with other	0.23	0.57	0.15
e.g.	0.27	0.97	0.16	other	0.16	0.53	0.09
i.e.	0.26	0.84	0.16	see	0.14	0.35	0.09

In table 77 the individual performance of all ISA KPs is listed when restricting analysis to the relation instances retrieved by each of the four search templates. This restriction is enforced because many KPs are specific to one or more, but rarely all four search templates. Thus even if there are 145 correct instances in the snippets sample for the entire ISA experiment[67], the number of correct instances *per template* are used when computing recall and precision in table 77. Merging all four templates when computing KP performance would unfairly have penalized KPs for not retrieving instances they could not possibly retrieve. For example, the pattern "is an" could not have retrieved the instance "antipsychotics", but only "antipsychotic". Anyway, performance scores based on the merged templates are listed in appendix 8.38.

As a first observation, the F-scores in table 77 are significantly higher than for the causal KPs. There are at least two possible explanations why this is the case. First of all, the search space of the non-hierarchical causal relation types is more open than is the case for the ISA relation (see the number of correct instances in table 43), but more importantly the number of different KPs discovered for the causal relation types is much higher than that discovered for the ISA relation and in particular the synonymy relation (see table 33). Thus any single causal KP, no matter how versatile, could not be expected to retrieve a great proportion of all the correct instances in the sample. Secondly, judging the correctness of ISA relation instances appears to be easier than judging the correctness of causal relation instances. This second explanation is corroborated by the fact that inter-annotator agreement is considerably higher in the ISA experiments than in the causal experiments (see table 40), and that the experts are less unsure when judging ISA as opposed to causal relation instances (see table 41).

A second observation is that the "a" template features the two best performing KPs in terms of F-score, namely "like" and "such as". However, many of the KPs for this template, in fact, really belong in template "b", because the hypernym is used as an adjective and is followed by a noun in the plural, for example "antipsychotic agents|drugs <KP> <hyponym>". Also, many KPs for this particular template express both genericity and causality at the same time, for example "<hypernym> effects|actions|properties of <hyponym>". This is presumably a domain dependent phenomenon, and it will be investigated more closely in section 6.5.

Finally, some patterns are extremely reliable, but have relatively low recall (fx. "e.g." and "i.e."). Also, the pattern "is an" reflects the fact that all the six types of central nervous system agents used in the KP discovery phase started with an "a" (see figure 11). This was an unintended effect, but it underscores the importance of chosing one's seed pairs carefully.

6.3.4 Synonymy patterns

Table 78 lists the precision and recall of all KPs discovered for synonymy based on the judgments of the four domain experts and computed over all five experiments.

Generally speaking, the patterns appear to be less domain dependent than was the case for some of the ISA patterns discussed in subsection 6.3.3. However, the bottom three patterns, "severe", "mild" and "acute" are presumably domain dependent.

[67] 57 when "haloperidol" is used as input term and 88 when the input is "antipsychotic(s)"

Table 78: Precision and recall of synonymy patterns (expert judgments)

Pattern	F-score	precision	recall (of 16)
aka	0.60	0.64	0.56
also known as	0.57	0.58	0.56
is also called	0.53	0.66	0.44
called	0.22	0.15	0.38
i.e.	0.20	0.17	0.25
is known as	0.09	0.06	0.25
or	0.06	0.04	0.19
refers to	0.06	0.04	0.12
see	0.05	0.03	0.12
means	0.02	0.01	0.06
was defined as	0.00	0.00	0.00
severe	NA	NA	NA
mild	NA	NA	NA
acute	NA	NA	NA

None of these patterns retrieve any synonyms, but this does not mean that they are completely useless. They appear to be dependent on a particular semantic type of argument, namely physiological phenomena. The reason they do not perform in the five experiments is that none of the input terms are physiological phenomena, but rather chemical substances. That they were discovered in the first place can be explained by the nature of some of the synonymy training pairs, for example "pruritus - itching" (see appendix 8.29).

Secondly, it is observed that three of the synonymy KPs are also ISA KPs, namely the patterns "called", "see" and "i.e.". These three KPs are likely the culprits behind many of the hypernyms returned by the system in some of the synonymy experiments discussed in subsection 5.5.5. As "see", at least, appears to be a relatively low performing pattern, it might be disqualified on these grounds.

Finally, it is interesting that the informal acronym "aka" is the best performing pattern in terms of F-score. This is probably yet another effect of using WWW text snippets rather than a tidy collection of academic papers as a source of knowledge.

6.3.5 Conclusion

In summary, a number of high performing KPs have been identified for each of the four relation types. Analyzing and comparing individual KP performance scores across the four different relation types gives rise to the following observations.

- Due to the greater search space of the causal relations and the greater complexity of the knowledge expressed by these relations, the individual performance of causal KPs tends to be lower than ISA and synonymy KPs.
- Text genre can be targeted by using particular KPs. For example causal KPs including the verb "induce" tend to co-occur with the technical synonym "emesis"

rather than “vomiting”.

- The four different ISA search templates differ significantly in terms of precision and recall. The highest precision is achieved by using the “b” template, the highest recall by using the “a” template (see table 76).
- Some KPs appear to be domain dependent.
- A few KPs may even require certain semantic types as arguments (for example “mild” and “acute”).
- Some KPs can instantiate multiple relation types. For example, there appears to be an overlap between ISA and synonymy.

Interestingly, the tense and modality of patterns for the “may_prevent” and the “induces” relations appear to be different. While, “may_prevent” KPs tend to be non-modal and in the present tense, “induces” KPs tend to be qualified by modal verbs. Thus the top three “may_prevent” KPs are: “helps prevent” (F=0.16), “prevents” (F=0.16) and “protects against” (F=0.14), while the top three “induces” KPs are: “may cause” (F=0.20), “can cause” (F=0.18) and “can lead to” (F=0.17). This could be explained by the fact that drug manufacturers may wish to downplay the likelihood of encountering undesirable side effects of their drugs, for example aspirin, while stressing the certainty of their beneficial effects. However, this observed morphological difference of KPs instantiating the two causal relations is likely to be domain dependent. Finally, it is apparent that the crude precision scores computed using negative term pairs in section 4.2 are unrealistically high when compared to the *actual* precision scores computed on the basis of expert judgments.

While this section discussed the precision and recall of individual KPs, section 6.4 will now examine the overall recall of WWW2REL and the proportion of “new” knowledge discovered by the system.

6.4 Recall versus UMLS and “new” knowledge

> [...] using a gold standard for the evaluation of automatically constructed ontologies is sometimes problematic and may lead to wrong conclusions about the quality of the learned ontology. This is due to the fact that if the learned ontology does not mirror the gold standard, it does not necessarily mean that it is wrong.[Cimiano and Staab, 2005]

Although the above quote discusses the problem of evaluating complete ontologies which have been automatically generated from text collections, one faces the same problem when automatically extracting semantic relation instances from text, as these are in a sense ontology fragments. Just because a particular extracted relation instance is not recorded in the UMLS Metathesaurus it does not mean that this instance is “wrong” or irrelevant at any rate. In this section an evaluation of recall versus relations recorded in the UMLS Metathesaurus will be attempted, even if such an evaluation is likely to provide only a very conservative estimate of system recall compared to recall versus the gold standard generated by the four domain experts.

Table 79: Existing relations in the UMLS Metathesaurus

experiment	#concepts for X	#term variants for X
selenium <may_prevent> X	1	2
aspirin <induces> X	0	3
haloperidol <ISA> X	6	33
X <ISA> antipsychotic	82	213
X <induces> vomiting | emesis	38	82
lactose	NA	0
glucose	NA	2
formaldehyde	NA	6
progesterone	NA	3
vitamin c	NA	3

Table 80: Recall versus UMLS - "aspirin has_physiologic_effect X"

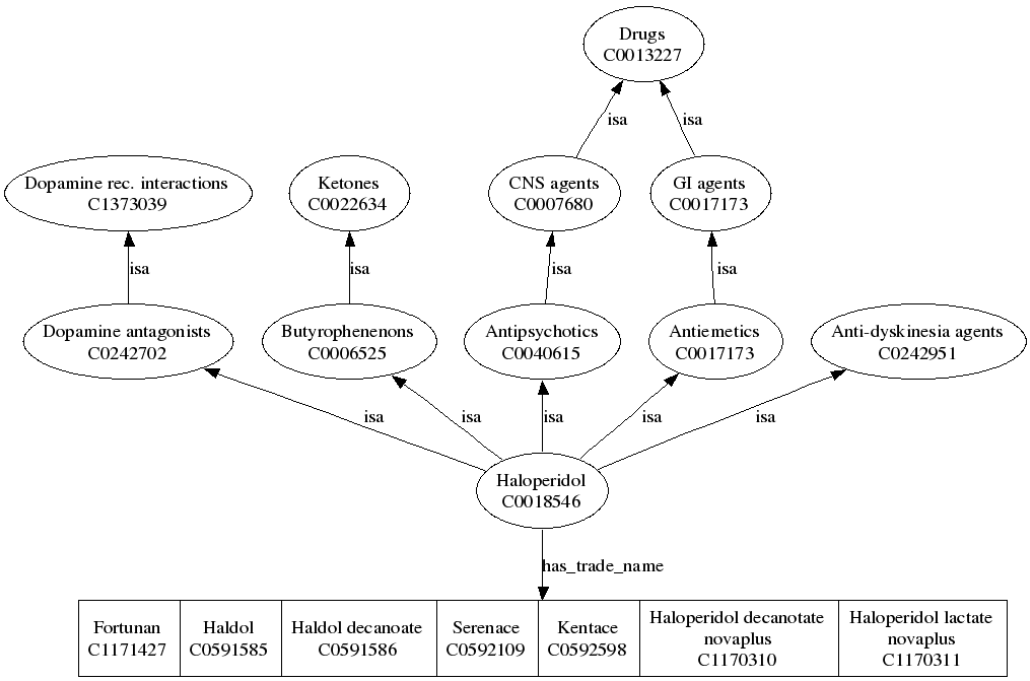
UMLS	WWW2REL near-match	judgment
decreased platelet aggregation	significant platelet dysfunction	1,2,1,1
	expected inhibition of platelet function	1,1,1,2
	equivalent platelet inhibitory effects	1,1,2,1
decreased prostaglandin production	reduced prostaglandin hydroperoxidase activity	2,1,2,1
decreased Thromboxane production	thrombocytopenia	1,1,1,1
	its anti-thrombotic activity	2,1,1,1

Table 79 summarizes the number of concepts already recorded in the UMLS Metathesaurus for each of the experiments on ISA, "induces" and the "may_prevent" relation. For the five synonymy experiments the number of concepts for X is not available ("NA") because synonymy is not a conceptual relation.

The main observation which can be made on the basis of this table is that there is virtually no information in the UMLS on the (side) effects of aspirin or on the beneficial effects of selenium. For "selenium may_prevent X" a single concept is recorded, namely "deficiency diseases", but for "aspirin induces X" we get nothing. However, via an analogous UMLS relation type, named "has_physiologic_effect", we do find three concepts associated with aspirin and each represented by only a single term. None of these are found by the system as *exact* strings, but WWW2REL does find the equivalent expressions listed in table 80. The example is simply meant to illustrate the dangers of computing recall automatically through simple string matching. In this case automatically computing recall versus the UMLS would have given 0%.

For the non-causal experiments UMLS does record a number of relations, and table 82 provides a breakdown of the proportion of "new" versus existing knowledge retrieved in each experiment. However, evaluating recall for the ISA relation introduces new challenges, both when finding hypernyms of "haloperidol", but also when find-

Figure 38: Computing recall against the UMLS - ISA relations



ing examples of antipsychotics. One question is what taxonomical distances to accept when computing recall against the UMLS Metathesaurus? Figure 38 illustrates how there are multiple hypernyms for "haloperidol" and how these hypernyms may also have multiple hypernyms and, in other words, form a complex polyhierarchy. When computing recall for the "haloperidol ISA X" experiment in table 82, only the immediate hypernyms in the Metathesaurus (in this case five concepts) are taken into consideration.

When computing the ratio of new versus existing knowledge for the "X ISA antipsychotic" experiment, it is important to observe that many of the antipsychotic drugs have a range of different trade names, in the case of "haloperidol" no less than seven (see figure 38). Since the trade names are recorded as individual *concepts* rather than term variants, they must be found by querying each of the 82 antipsychotic drug concepts in the Metathesaurus for the "has_trade_name" relation (see appendix 8.26).

Another problem when trying to compute recall against the UMLS is the vast number of "unnatural" variants recorded for each concept. Table 81 provides an example for the concept "alcohol". As illustrated by the table there are no less than 15 UMLS term variants for the two alcohol concepts, most of which would never be found (as exact strings) in natural language text because of the parentheses and syntactic inversion. This will also "artificially" reduce recall if computed automatically. For this reason, the "recall" column in table 82 expresses the proportion of UMLS concepts for which one or more term variants are retrieved by the system. For the five synonyms, "recall"

Table 81: UMLS term variants of “alcohol”

Concept UI	term variant
C0001975	alcohol
“	alcohols
“	alcohols (chemical class)
“	alcohol [chemical class]
“	alcohol in any form
“	alcohol preparation
C0001962	alcohol
“	ethyl alcohol
“	alcohol, ethyl
“	alcohol, dehydrated
“	grain alcohol
“	alcohol, dehydrated preparation
“	alcohol, grain
“	alcohol, ethyl preparation
“	alcohol %

does express the proportion of UMLS term variants retrieved by the system.

Finally, the system's lack of sophisticated NLP may also skew the figures for both the proportion of new knowledge detected and the recall versus UMLS, had they been automatically computed. The three main problems are

1. conjunctions and lists
2. appositions
3. non-restrictive (non-essential) adjectival modifiers

As for the conjunctions the recall for “formaldehyde” in table 82 is, in fact, 57% (4/7) and not 29% as would have been the result of an automatic string matching operation. This is because the instance “formic aldehyde or methyl aldehyde” is not recorded in the UMLS in this exact form, but as the two separate parts of this disjunction. The same is the case with “vitamin C” for which the system returns five different strings judged to be correct, but when morphosyntactic variations like “ascorbic acid and|or ascorbate” are conflated, only three term variants remain, bringing the proportion of new knowledge down to 33% (1/3). Also, in the experiment on finding antipsychotics the degree of “new” knowledge retrieved by the system would be artificially high if computed automatically. Both “risperdal” and “zyprexa”, for example, are recorded as antipsychotics in the UMLS, but the two system candidates “zyprexa and risperdal” and “zyprexa or risperdal” are obviously not.

An example of a correct candidate expressed as an apposition is “muscarinic receptor agonist, xanomeline” in which a hypernym accompanies the antipsychotic drug “xanomeline”. In this case, “xanomeline” is, in fact, *not* registered as an antipsychotic

Table 82: ”New” knowledge retrieved per experiment (manual analysis)

experiment	recall vs. UMLS	A=correct terms	% of A not in UMLS
selenium <may_prevent> X	0% (0/1)	49	100%
aspirin <induces> X	NA	143	100%
X <induces> vomiting	5% (2/38)	61	95%
X <induces> emesis	3% (1/38)	25	96%
haloperidol <ISA> X	80% (4/5)	48	69%
X <ISA> antipsychotic	30% (25/82)	57	39%
lactose	NA	1	100%
glucose	100%	3	33%
formaldehyde	57% (4/7)	5	20%
progesterone	0% (0/7)	2	100%
vitamin c	66% (2/3)	3	33%

Table 83: New relation instances retrieved by WWW2REL (examples)

experiment	terms not recorded in the UMLS Metathesaurus
selenium may_prevent X	prostate cancer risk, lung cancer risk, DNA damage, ...
aspirin induces X	ulcers, stomach irritation, bronchoconstriction, tinnitus, ...
haloperidol ISA X	CNS-depressant drug, less-sedating neuroleptic, ...
X ISA antipsychotic	amisulpride, bretazenil, fluoxetine, iloperidone, zeldox ...
X induces vomiting	meningitis, overeating, stomach virus, salmonella, zoloft, ...
X induces emesis	apomorphine, carboplatin, cisplatin, chemotherapy, tramadol, ...
synonyms of lactose	milk sugar
synonyms of glucose	corn sugar
synonyms of progesterone	progestin, progestogen
synonyms of formaldehyde	metylene oxide
synonyms of vitamin C	ascorbate

in the Metathesaurus, but even if it had been registered the candidate would have been counted as a new term when using simple string matching techniques. Finally, the lack of lemmatization is less of a problem because plural forms are usually included in the sets of term variants of concepts in the Metathesaurus. Non-restrictive adjectival modifiers like "*conventional* antipsychotics" are rare in the case of specific drug names, but very common when looking for hypernyms of the drug, "haloperidol" (see table 42 for examples). These would also artificially boost the degree of "new" knowledge and lower recall if computed automatically.

These cases again stress the potential usefulness of upgrading the system with more comprehensive morphosyntactic analysis. However, it should be noted that an automatic analysis of the three linguistic phenomena listed above is not straight-forward. Conjunctions, for example, often involve elipses which are non-trivial to detect automatically[68]. Also, it may be hard to determine automatically whether an adjectival modifier is restrictive or not. Thus in the absence of an advanced NLP module, both recall figures and the proportion of new knowledge detected by the system (table 82) are based on a manual analysis of system output in which conjunctions, lists and appositions are decomposed, while non-restrictive adjectival modifiers are stripped from the NPs.

Looking at the percentages in table 82 and the examples in table 83 new instances are primarily retrieved for the causal relation types. Indeed, nearly 100% of all causal instances judged to be correct by the experts are not recorded in the UMLS. For the ISA relation the proportion of new knowledge is somewhat lower, especially the coverage of antipsychotic drugs in the UMLS appears fairly comprehensive (only 39% of all correct instances are not recorded). As regards the relatively high proportion of new hypernyms proposed for "haloperidol" (69%), this can be explained by the fact that many of the hypernyms are somewhat more superordinate than is allowed when computing recall. Examples include "medication", "medicine", "drug" and "antagonist"[69].

With the two exceptions of "progesterone" and "lactose", the proportion of new knowledge is considerably lower in the synonymy experiments. One reason is that the number of synonyms for a particular concept is much lower, and presumably also more fixed, than the number of new drugs of a particular category or the number of side effects discovered for a particular drug. Also, as mentioned in subsection 5.5.5 the two new synonyms retrieved for "progesterone" are actually both questionable according to the UMLS Metathesaurus.

Although domain experts might question the usefulness of enriching an ontology like the UMLS with layman expressions like "corn sugar" or "milk sugar", such expressions are certainly terminologically relevant and could be useful to classify and include in an ontology to be used, for instance, when communicating medical texts to the wider public. Also, it must be stressed that the biomedical domain is only meant as a case study and that WWW2REL is envisioned as a domain independent system which might be applied to enrich ontologies in domains featuring less systematic and perhaps more ambiguous nomenclature.

[68] e.g. "persistent stomach upset or {Ø} pain"

[69] Indeed, many of these candidates are borderline cases with average correctness judgments of 1.50.

Finally, it is perhaps surprising that recall versus the UMLS for the "X induces vomiting|emesis" experiment is so low. One may speculate that the search space of these queries is simply too large to expect high recall in a small sample of text snippets. On the other hand lots of correct relation instances are retrieved, although some may not be considered relevant in a strictly biomedical context. The only correct candidates returned by the system and recorded in the UMLS are "ipecac" and "ipecac syrup".

6.4.1 Conclusion

This section discussed and illustrated the problems of automatically assessing the recall of a relation extraction system against a gold standard terminological resource. Even with more sophisticated NLP than is presently implemented computing recall automatically will result in an erratic or, at best, very conservative estimate. The main reason is that linguistic variation will always be greater in natural language than in terminological resources and that mapping term variants onto concepts with high precision typically will require human intervention. The problem of linguistic variation is compounded by using the WWW as a knowledge source rather than academic writing which follows more predictable patterns. For these reasons measurements of recall versus the UMLS were based on a manual analysis of system output. Recall versus the UMLS ranged from 100% in the task of finding synonyms of "glucose" and 80% for finding hypernyms of "haloperidol" to 0% in the case of synonyms of "progesterone".

In the matter of determining the proportion of new knowledge retrieved by the system, an automatic evaluation is obviously not feasible. Based on the expert judgments and a manual analysis of system output, the proportion of new versus recorded knowledge ranged from 100% in the case of the causal relations and synonyms of "lactose" to 20% in the task of finding synonyms of "formaldehyde". In conclusion, even if WWW2REL may have a relatively low recall versus the UMLS (at least in some of the experiments), it does discover lots of new knowledge, and this is essentially the main purpose of the system.

6.5 Domain specificity of relations and KPs

Having thoroughly examined the precision and recall of knowledge patterns instantiating four different relation types, it is now time to briefly investigate the third quality parameter of a KP, namely its *portability* across domains (also mentioned in chapter 1). Assessing a pattern's true portability would require an extensive analysis of its use in multifarious domains, but even if this section only features an analysis of a single other domain, this should at least give some indication of the domain dependence of each KP as well as more global portability issues.

The domain used in this comparison is Information Technology (IT). IT was selected for two reasons. Firstly, in contrast to the domain of Biomedicine, IT is more severely affected by determinologization processes (see the discussion in subsection 2.2.3), and thus in addition to highlighting portability issues, the comparison may also reveal additional challenges related to this phenomenon. Secondly, in the absence of a panel of experts like the four pharmacists the author is able to act as an expert for this domain.

Table 84: “perl ISA X” - top 10 candidates

candidate (“kpr_head”)	judgment	candidate (“kpr_bnc_head”)	judgment
language	2	oopl	1
programming language	1	tools oreilly	3
scripting language	1	improbables	3
interpreted language	1	envt ajaxtk	3
object-oriented language	1	esoterica	3
interpreted programming language	1	lanuage	2
oo language	1	optimizer	3
implementation language	1	envt	3
network-capable high-level language	1	tools amazonuk	3
its macro language	2	scripting	2
...	...	...	...

Six small experiments are carried out to test the domain specificity of the filtered KPs for synonymy, ISA, “induces” and “may_prevent”, respectively. The findings of each experiment are described in the following subsections. Although the experiments do not feature as detailed an analysis of the performance of the various ranking schemes as in section 5.4, the effects of the BNC-based filter on precision will also be discussed.

6.5.1 The ISA relation

“Perl ISA X” Table 84 lists the top 10 candidates returned for the ISA relation when using “perl” as an input term, grouping by NP head and ranking by KP range with (kpr) and without (kpr_bnc) the BNC filter.

In this case the BNC filter has an absolutely disastrous effect on precision, as the NP head, “language” gets heavily penalized and is replaced by incorrect, noisy candidates including spelling mistakes (“lanuage”) and bits of URLs.

When turning the BNC filter off, head grouping proves a useful mechanism not just for precision, but also because it may provide not just one, but several different hypernyms. Thus “programming language” is a more superordinate hypernym than “interpreted language” or “object oriented language”, and just be studying this top 10 a terminologist can build the first part of an entire taxonomy of programming languages.

“X ISA programming language” When looking for hyponyms, head grouping does not improve precision. This is especially true in the case of programming languages which have very dissimilar names. Thus table 85 lists the top 10 candidates returned for the ISA relation when the input term is “programming language(s)” and ranking by KP range with (kpr) and without (kpr_bnc) the BNC filter.

In this case precision for top 10 is 100%, but the BNC filter should not have the credit for this performance. Although it does not reduce precision for top 10, it does *unfairly* penalize the programming language “java”, which occurs 218 times in the BNC, but always in the sense of the Indonesian island rather than the programming language. We know this because Java (the language) did not exist before 1995 and all

Table 85: “X ISA programming language” - top 10 candidates

candidate (“kpr”)	judgment	candidate (“kpr_bnc”)	judgment
java	1	c	1
c	1	perl	1
c++	1	visual basic	1
visual basic	1	c#	1
lisp	1	html	1
html	1	cc++	1
perl	1	javascript	1
c#	1	c and c++	1
prolog	1	c, c++	1
cc++	1	cobol	1
...	...	...	...

text in the BNC predates 1994[70]. This example illustrates that while the BNC filter proved useful in the domain of Biomedicine, it can have dangerous side effects in domains like IT where many terms are coined by extending the meanings of existing lexical units.

The portability of the ISA KPs can be inferred from the frequency numbers in table 86 which are based on both experiments. Generally speaking, the patterns in the lower part of the right column appear to have poor portabillity as they have low recall in the domain of IT. The two biomedical patterns “drugs such as” and “agents such as” which ranked number two and three for their template in terms of both precision and recall (see table 77) are thus, not surprisingly, useless in IT.

6.5.2 The causal relations

This subsection features two small experiments, one for “may_prevent” and one for “induces”, which will assess the portability of KPs discovered in chapter 4 for these two UMLS relations.

“Firewall(s) <may_prevent> X” Table 87 lists the top 10 candidates returned, this time for the “may_prevent” relation, when the input term is “firewall(s)”, grouping is applied and ranking is by KP range with (kpr) and without (kpr_bnc) the BNC filter.

Again the BNC filter does *not* improve precision, in fact, it reduces precision. The effect of turning on the BNC filter is not as disastrous as in the “perl ISA X” experiment, but this is really just a coincidence because all text in the BNC predates 1994, i.e. from the blissful era before nasty things like “spyware” and other “malware” were concocted. The example reveals that there are actually *two* factors which endanger the use of BNC as a filter in the domain of IT.

1. determinologization

[70]This also explains why older languages like “prolog” and “lisp” get (fairly) penalized

Table 86: Portability of ISA KPs

pattern (template)	IT frequency	pattern	IT frequency
such as (b)	214	see	60
such as (a)	205	i.e.	58
and other (d)	199	agent	45
or other	189	drug	31
with other	180	other	27
like	167	agents	21
as (c)	165	is an effective	19
and other (c)	145	properties of	18
as (d)	144	another	18
is	141	drugs	17
is an	132	effect of	7
called	128	activity	4
including	124	effects of	4
as an	97	activity of	3
include	93	action of	2
has	93	exerts its	0
is a new	92	drugs such as	0
a new	91	efficacy of	0
has an	88	agents such as	0
an	77	actions of	0
e.g.	65		

Table 87: “firewall(s) may_prevent X” - top 10 candidates

candidate (“head_kpr”)	judgment	candidate (“head_kpr_bnc”)	judgment
access	1	spyware	1
unauthorized access	1	viruses and spyware	1
unwanted access	1	bandwidth free spyware	2
external access	1	java applet	1
network access	1	applet	1
internet access	1	bandwidth	3
unauthorised access	1	more your bandwidth	3
your access	1	access	1
unauthorized network access	1	unauthorized access	1
inbound access	1	unwanted access	1
...	...	...	...

Table 88: “firewall(s) may_prevent X” - top 10 heads

candidate (“head_kpr”)	candidate (“head_kpr_bnc”)
access	spyware
attacks	applet
traffic	bandwidth
users	access
connections	hackers
hackers	malware
number	attacks
use	traffic
computer	signaling
security	transversal
...	...

2. practice of term formation by semantic extension of existing lexical units

Wheras the use of “spyware” by a considerable number of non-experts is a consequence of the first factor, “languages” are an example of the latter factor. Of course, there are also multiple examples of IT terms created by semantic extension and subsequently affected by determinologization.

Table 88 lists the top 10 heads when grouping by the same two schemes. It reveals that the concept of “bandwidth” had yet to be determinologized in the 1980s and early 1990s. Using a more recent general language corpus as a filter would presumably yield quite different (i.e. worse) results here.

The frequency numbers in table 89 provide information about the portability of the “may_prevent” KPs. Out of the 101 KPs discovered for “may_prevent” only 57 co-occur with “firewall” at least once on the entire WWW (the part indexed by Google). Table 89 lists examples of these domain-dependent patterns, but also of a few of the patterns which appear to be portable.

“Computer virus(es) <induces> X” Table 90 lists the top 10 candidates returned, this time for the “induces” relation, when the input term is “computer virus(s)”, head grouping is activated and ranking is by KP range with (kpr) and without (kpr_bnc) the BNC filter. In this case applying the BNC filter actually helps a little bit, in that “computer system outages” ranks as number one. However, most candidates in both columns of the table are equally vague, although some are correct (e.g. “loss” and “damage”). This example illustrates how using the WWW to extract semantic relations can be difficult if the target domain is undergoing determinologization.

Using “computer virus” as input rather than just “virus” may have contributed to part of the vagueness in the results. In this query “computer” is used as a domain label to disambiguate the otherwise polysemous word “virus” (and avoid hits from biomedical text sources, for example). However, as shown in extensive empirical investigations in [Halskov, 2005b, Halskov, 2005c, Halskov, 2005a] IT experts rarely use domain labels when they communicate with eachother, but these domain labels are typically

Table 89: Portability of “may_prevent” KPs

pattern	IT frequency	pattern	IT frequency
may_prevent	141	had significantly less	0
to prevent	122	significantly reduced	0
to reduce	108	based	0
help prevent	104	can cure	0
preventing	94	to cure	0
prevent	93	does not cure	0
...	...	heals	0
for relieving	0	numbs	0
relieves	0	deters	0
relieve	0	works against	0
in relieving	0	lessens	0
in treating	0	can relieve	0
decreased	0	combats	0
may ease	0	alleviates	0
relieved	0	use in	0
at preventing	0	...	...

Table 90: “computer viruse(s) induce X” - top 10 candidates

candidate (“kpr_head”)	judgment	candidate (“kpr_bnc_head”)	judgment
damage	2	computer system outages	1
serious damage	2	false-positives	3
real damage	2	havoc	2
irreversible damage	2	been causing havoc	2
costly damage	2	major havoc	2
very serious damage	2	damage	2
world wide catastrophic damage	2	serious damage	2
data loss and/or damage	1	real damage	2
extensive damage	2	irreversible damage	2
significant damage	2	costly damage	2
...	...	...	...

Table 91: Portability of "induces" KPs

pattern	IT frequency	pattern	IT frequency
causing	124	can induce	0
can cause	95	and induce	0
can result in	63	induced	0
cause	57	for inducing	0
causes	55	poisoning include	0
caused	49	poisoning	0
to cause	37	provokes	0
associated with	33	will produce	0
could cause	27	produced	0
...	...	in producing	0
to induce	0	may aggravate	0
induces	0	aggravates	0
overdose include	0	may experience	0
which induces	0	may lead to	0
induce	0	can lead to	0
or induce	0	...	...

used as a disambiguation device in non-expert discourse where the domain context is not established from the outset. As we cannot expect non-experts to provide us with a technical account of the effects of viruses on computer systems, the vague arguments returned in this experiment are not really surprising. Extracting (terminologically relevant) semantic relation instances from the WWW thus imply special challenges for the domain of IT, which are less pronounced in Biomedicine.

Finally, table 91 gives an indication as to the portability of the "induces" KPs. Overall, only 42% (30 out of 71) of these patterns occur at all with the term "computer virus(es)", and judging by the zero occurrence patterns in the table, verbs like "induce", "produce" and "aggravate" are domain specific markers of causality and cannot be ported to just any other domain.

6.5.3 Synonymy

Synonyms of "subroutine" Table 92 lists the top 10 candidate heads returned, this time for synonymy when the input term is "subroutine", ranking is by KP range and head grouping is on with/without the BNC filter. With the filter turned off, five correct synonyms are among the top 10 candidate heads, namely "function", "method", "subprogram", "procedure" and "sub". However, when the filter is turned on, no correct candidate heads are returned among top 10.

Finally, table 93 lists all 15 synonymy KPs as ranked by their frequency of occurrence in the "subroutine" experiment. With the exception of the four patterns "acute", "mild", "severe" and "was defined as", all KPs also occur as knowledge probes in the IT domain. This would indicate that synonymy is the most domain-independent relation type of the four relations investigated in this thesis, and that the synonymy patterns

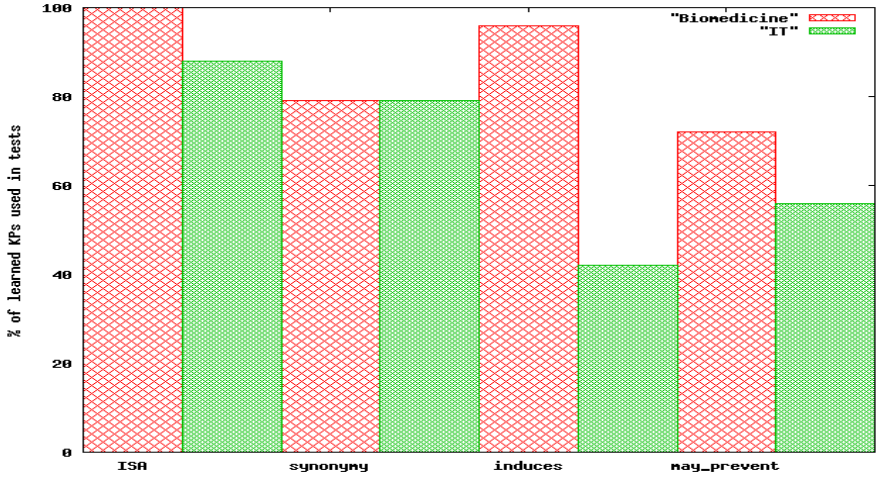
Table 92: Synonyms of “subroutine” - top 10 candidate heads

candidate (“kpr-head-bnc”)	judgment	candidate (“kpr-head”)	judgment
subprogram	2	function	1
coderef	3	method	1
366	3	closure	3
mysub	3	subprogram	2
canceling	3	variable	3
306d	3	procedure	1
autoload	3	c	3
createchildcontrols	3	constructor	3
fdate	3	parameter	3
minimumvalue	3	sub	1
...	...		

Table 93: Portability of synonymy KPs

pattern	IT frequency
or	314
called	63
see	48
i.e.	41
refers to	41
means	38
also called	22
is known as	18
aka	11
also known as	10
is also called	4
acute	0
mild	0
severe	0
was defined as	0

Figure 39: Proportion of KPs making a contribution in tests for two different domains



could readily be ported to other domains.

6.5.4 Conclusion

In conclusion, this section tested both the domain dependence of KPs discovered using biomedical term pairs, but also the portability of the effective BNC heuristic from Biomedicine to the domain of IT.

Firstly, as can be seen by the histograms in figure 39, it was discovered that the KPs for synonymy appeared truly domain-independent. The patterns for the ISA relation appeared relatively portable, while KPs for the two causal relations were the least portable of them all. However, by avoiding certain verbs like “induce”, “aggravate” and “produce”, for example, a wide range of causal markers could also be applied to find similar relation instances in another domain.

As to the usefulness of the BNC filter as part of a ranking scheme in the domain of IT, this cannot be recommended. Term formation by semantic extension of existing lexical units and the effects of determinologization make simple discounting by BNC frequency decidedly counter-productive or at best erratic. For such domains a more viable solution may be to exchange the BNC for a number of specialized reference corpora representing domains for which no formal similarities with the terminology of the target domain are expected.

[Pantel and Pennacchiotti, 2006] observe that the recall of relation extraction systems can be boosted (while maintaning precision) by employing a host of generic KPs but requiring that at least one high precision KP occur with each candidate instance. This is undoubtedly true, but the KP portability experiments presented in this section seem to indicate that many of these high precision KPs tend to be domain specific (“agents such as” and “drugs such as” are two cases in point), thus reducing the portability of the system.

7 Conclusion

In general terms, the main contributions of the thesis can be summarized by the following headings.

1. The development of a methodology to discover and filter knowledge patterns (KPs) for any semantic relation type on the WWW and to use these KPs to find instances of the relations also on the WWW.

   (a) Devising an "iteration range" filter minimizing noise in the KP discovery phase.

   (b) Devising and evaluating a set of schemes and two heuristics (see section 5.3) with which to rank the retrieved relation instances by their reliability.

2. A discussion and an analysis of the problem of measuring the recall of relation extraction systems against existing terminological resources like the UMLS Metathesaurus. Particularly, assessing the proportion of new knowledge found by such systems.

3. An analysis of the domain dependence of filtered KPs representing four different relation types.

4. An analysis of the domain dependence of heuristics which proved useful in the Biomedicine domain.

The following section summarizes key results as grouped by each of the contribution areas.

7.1 Key results

7.1.1 Methodology for pattern-based relation instance extraction

The implementation and evaluation of the WWW2REL ontology extension methodology indeed revealed that simply issuing unrestricted queries to web search engines, retrieving and processing semantically unannotated text snippets was a surprisingly useful way of both discovering KPs for a particular domain (Biomedicine), but also for filtering these KPs and for extracting terminologically relevant relation instances by means of the reduced KP sets. Although using the WWW as a knowledge source does involve more noise in the empirical data, this noise can relatively easily be minimized by measuring the KP range of each candidate instance (the "kpr" scheme) and using this as an indicator of its reliability. In fact, the performance of WWW2REL in terms of precision appears to be on a par with the most comparable relation extraction system (see subsection 5.5.8). It was also noted that many other systems operate on academic papers, often semantically annotated, and employ various non-portable filtering techniques (see table 7 in subsection 3.4.11).

On the negative side, the evaluation revealed that various minor extensions of the NLP implemented in WWW2REL are likely to improve performance. Possible improvements in this direction are listed in section 7.2.

KP discovery The results of the KP discovery experiments (section 4.1) revealed that

- term pairs used to discover KPs may need to be lexically filtered to avoid data sparseness even when querying the entire WWW.
- some relation types (e.g. ISA) may require several search templates because the form of their KPs can be morphosyntactically sensitive.
- forcing a verb in the KPs seems to eliminate much noise, but this may be too restrictive for some relation types (e.g. synonymy and ISA).

KP filtering The results of the KP filtering experiments (section 4.2) include

- the introduction of a novel KP filtering technique dubbed "iteration range filtering". This technique appeared to be effective for relation types like ISA and synonymy for which no linguistic restrictions on the pattern form were enforced.
- the establishment of a set of principles for the selection of negative term pairs as part of an additional KP filter. For example, including negative pairs which instantiate relation types semantically close to the target relation.

Relation instance extraction and ranking Overall, the results presented in chapter 5 showed that relation instances can be extracted with high precision without weighting the individual patterns (as is done in [Pantel and Pennacchiotti, 2006]), but by simply computing the range of filtered KPs with which each instance co-occurs and optionally apply one or two heuristics.

However, instances of non-hierarchical relations, in this case the causal relations "induces" and "may_prevent", proved harder to extract automatically than instances of ISA and synonymy relations. The experts often disagree (see subsection 5.2.5), and sometimes it is not known, or documented, whether a particular drug, for example, is associated with a particular beneficial or unwanted physiological effect.

Although WWW2REL was able to identify correct instances with high precision in four out of five synonymy experiments, these experiments revealed that finding synonyms in the strict terminological sense of the word is somewhat harder than finding terms which are just "semantically related". Finally, when looking for things which induce a certain side effect ("vomiting" in the test case), using a more domain specific synonym for this side effect ("emesis" in the test case) significantly boosted both precision and recall.

Six ranking schemes and two heuristics were devised, thoroughly tested and evaluated. The overall best strategy proved to be a ranking of instances by the range of different KPs with which they occur, i.e. the scheme "kpr", or alternatively, "fkpr" (see section 5.3 for a description). Activating the BNC discounting heuristic, which penalizes instances having a high frequency in a general language corpus, boosted precision and recall even further. Another heuristic of grouping instances by their NP head also had a positive impact on performance, and, in fact, applying both heuristics simultaneously gave the overall best performance (see table 70).

7.1.2 Measuring recall and new knowledge

For all four tested relation types, the implemented system, WWW2REL, was able to extract *new* relation instances not recorded in the most comprehensive terminological resource for the domain of Biomedicine, namely the UMLS Metathesaurus.

A manual analysis of the degree of "new" (unrecorded) knowledge retrieved showed that 100% of the correct causal relation instances ("induces" and "may_prevent") were new. 69% of the correct hypernyms retrieved for "haloperidol" were new, 39% of the correct hyponyms retrieved for "antipsychotic(s)" were new and between 20% and 100% of the correct synonyms retrived were new, or unrecorded in the UMLS.

However, a number of challenges make it difficult to measure *automatically* the degree of "new" knowledge and system recall versus the UMLS (section 6.4).

- unnatural term variants in the gold standard artificially reduce recall.
- non-restrictive adjectival modifiers in system candidates artificially reduce recall, but boost the degree of "new" knowledge found.
- lacking decomposition of conjunctions, lists and appositions artificially reduces recall, but boosts the degree of new knowledge found.

7.1.3 Domain specificity of KPs

As is rarely done in comparable work, the domain portability of the filtered KPs was investigated empirically (section 6.5). It was concluded that, in the case of Biomedicine vs. IT, synonymy KPs were completely portable. ISA KPs were slightly less portable and only between half and two-thirds of the causal KPs were portable. However, the domain-specific KPs did not appear to reduce system precision as high precision was achieved also in the domain of IT for all four relation types.

7.1.4 Domain specificity of instance filtering heuristics

The BNC discounting heuristic seemed to be domain dependent (see the experiments in section 6.5), in that a number of correct instances, for example the programming language "java", were unfairly penalized due to their formal similarity with words referring to non-target domain senses. The head grouping heuristic, on the other hand, appeared to be domain independent and was useful not only in Biomedicine but also in IT.

In conclusion, the benefits of activating the two heuristics depend on

1. the size and homogeneity of the search space
    (a) if large and homogenous (e.g. finding hypernyms or effects given a drug)
        i. activating head grouping will boost performance
    (b) if small or heterogenous (e.g finding synonyms or drugs given an effect)
        i. head grouping is less helpful

2. the target domain and the assumed specificity of the target arguments
    (a) if terms in the target domain are formed by semantic extension of general language lexical items (e.g. IT)
        i. activating BNC discounting can be counter-productive
    (b) if target terms are less specific than the input term (e.g. finding synonyms of "lactose")
        i. activating BNC discounting can be counter-productive
    (c) if target terms are highly specialized (e.g. finding hyponyms of a drug)
        i. activating BNC discounting will boost performance

7.2 Future work

This section outlines different directions into which future work might be directed. Future work may progress along two dimensions, namely an application-oriented one and a basic research-oriented one. The first dimension would involve augmenting the functionality and performance of future versions of WWW2REL, and the second dimension would involve a range of research questions including, for example, a more comprehensive analysis of the nature of knowledge patterns. Many of the following subsections on possible future work were also listed as thesis limitations in section 1.4.

7.2.1 Empirical data and sparseness issues

Firstly, testing to what extent using WWW text snippets complicates the relation extraction task. What sort of performance increase might be gained by applying WWW2REL to a tidy collection of MEDLINE abstracts? Also, how big a problem is data sparseness when using a limited collection of MEDLINE abstracts versus the entire WWW?

Secondly, using a greater range of input terms to explore the effect of variables like term frequency on system performance. The input terms used in the eleven experiments in this thesis were all relatively frequent, and it is not inconceivable that using less frequent input terms might result in data sparseness.

Thirdly, the threshold values of the KP filters described in 4.2.5 were set empirically, but it would be interesting to further investigate these values and evaluate the performance impact of different threshold combinations in different settings. Presumably, the optimal selection of filters and threshold values will be determined not only by the target semantic relation type, but also by the domain specificity, or noisiness, of the text source.

Fourthly, will the discounting heuristic perform even better if the BNC unigram frequencies are exchanged for the ngram statistics recently made public by Google? It is expected that such an exchange will have little effect, because there is a fairly strong correlation between the BNC unigram frequencies and the corresponding Google unigram hit counts. Also, when combining the discounting heuristic with the head grouping heuristic, there is no need to go beyond unigrams. Unless, of course, one aims to build complete taxonomies in a recursive fashion as discussed in e.g. [Gillam, 2004, Gillam et al., 2005].

Finally, one might explore the impact of manipulating Google variables like "daterange" and "site" on system performance. Will daterange restrictions weed out obsolete terms and improve performance? Will restricting searches to authoritative and domain-specific web sites improve performance? It is expected that such restrictions will cause data sparseness, reduce the portability of the system and not improve performance significantly.

7.2.2 Language of analysis

Applying the WWW2REL ontology extension framework to a minority language like Danish to see whether data sparseness issues will arise. It is the expectation that for highly specialized domains like Biomedicine there will not be enough textual content in Danish to reliably discover KPs from WWW text snippets because few biomedical text books, academic papers and other terminological resources are available in Danish due to the ongoing, international anglification.

7.2.3 Domain of analysis

Applying WWW2REL to a range of different domains to further test the domain specificity of the discovered KPs, but also of the ranking schemes and heuristics. While homographs are rarely an issue when restricting analysis to domain-specific text, it is a very real problem when searching for knowledge on the entire WWW. The experiments for the domain of IT in section 6.5 illustrated that using domain tags like "<computer> virus" can be a solution for input terms which have homographs in non-target domains. However, it also revealed that using such tags may lead to the extraction of semantically vague arguments because domain tags are primarily used in communication involving non-experts. In the case of Biomedicine no such input term disambiguation was necessary because terms like "haloperidol", "aspirin" and "selenium" have no homographs in other domains. In this sense, the Biomedicine domain may, in fact, be a relatively benign domain as compared to many other cases where using uncategorized WWW text may require a word sense disambiguation module.

7.2.4 NLP improvements

As mentioned in chapters 4, 5 and 6 a more sophisticated NLP module may improve system performance both in terms of precision and recall. Key areas which could be improved include the following.

- automatic decomposition of conjunctions and lists in relation instances
- more sophisticated treatment of PP attachments in relation instances
- lemmatization of relation instances
- disallowing input term modifiers

Full parsing, however, is not deemed to be worth the effort, or even possible, because of the noisy and fragmented nature of the text snippets making up the system's data source. Also, full parsing is rarely seen in text mining systems.

7.2.5 System integration

Firstly, the individual semantic relation extractors could be combined with each other to boost system performance. For example, KPs for ISA and a causal relation, might make it possible to piece together complete definitions from fragmented text snippets on the WWW. While applications like [Klavans and Muresan, 2000] extract complete definitions from free text such approaches may encounter data sparseness problems which can be avoided by looking for hypernyms (ISA) and delimiting characteristics (e.g. causality) separately. Also, in the case of the "X induces emesis" experiment, X constituted a large set of things some of which were irrelevant in a biomedical context. However, running 1) "X induces emesis" and 2) "X ISA drug" and eliminating all X from 1) which are not found in 2) may filter out these non-biomedical concepts and reduce the search space. Finally, the synonymy extractor could be used to cluster synonymous arguments of relation instances (for example "bleeding" and "hemorrhage") and further reduce the workload of the terminologist.

Secondly, the WWW2REL system could be combined with an application like TerminoWeb[71] [Agbago and Barrière, 2005] developed at the Institute for Information Technology, National Research Council of Canada. Terminoweb is an *interactive* tool developed to help terminologists compile documents which are rich in specialized and explicit knowledge from the WWW and subsequently extract both terms, but also knowledge-rich contexts from these documents. A unique feature of the TerminoWeb system is that documents are ranked by their density of knowledge patterns as well as a number of other relevance parameters. However, Terminoweb lacks a KP discovery module like WWW2REL which would help the user extend the system's inventory of KPs, whether they be domain independent or not.

7.2.6 Knowledge Pattern issues

Automatically discovering, filtering and applying KPs in the task of relation instance extraction revealed a number of pattern-related issues which so far have only been discussed *en passant* in the thesis.

1. data sparseness (subsection 4.1.3)
2. invalid patterns (subsection 4.2.4)
3. discontinuous patterns (subsection 4.2.4)
4. ambiguous patterns (subsections 5.5.5 and 6.3.4)

The following paragraphs will briefly discuss how the two issues of discontinuous and invalid KPs could be explored in future work.

Discontinuous patterns Among the patterns discovered for the "induces" relation were fragments of discontinuous patterns in which the cause is embedded in a possessive construction headed by compounds like "side effect(s)" and followed by a linking

[71] http://termino.iit.nrc.ca

Table 94: Discontinuous knowledge patterns (examples)

Relation type	left	middle	right
induces	side effect(s)? of	<t1> is | are | include <t2>	
may_prevent		<t1> provides <t2>	treatment | therapy | relief
...	...	...	...

Table 95: Inversion of causal relations

Relation type	Added element	Template
may_prevent -> induces	insufficiency	<t1> insufficiency {KP} <t2>
may_prevent -> induces	overdose	<t1> overdose {KP} <t2>
may_prevent -> induces	insufficient|(lack of)	insufficient|(lack of) <t1> {KP} <t2>
may_prevent -> induces	excessive|(too much)	excessive|(too much) <t1> {KP} <t2>
induces -> may_prevent	low doses of	low doses of <t1> {KP} <t2>
may_prevent -> induces	high doses of	high doses of <t1> {KP} <t2>
...	...	...

verb ("is", "are", "include") and the effect. Another example is the semantically vague verb "provide" (discovered in the "may_prevent" snippets) as used in the template in table 94. In this case the "may_prevent" relation is not instantiated as a verb in the middle context, but as a noun in the right context.

In the WWW2REL implementation presented in this thesis no attempt was made at identifying such discontinuous KPs. All patterns were simply extracted from the middle contexts of the term training pairs. Discontinuous sequences of tokens are hard, if not impossible, to identify using traditional n-gram techniques. However, alternative ways of capturing non-contiguous word associations have been proposed and have been referred to as "skipgrams" or "concgrams" [Cheng et al., 2006, p4]. While using "concgrams" might be a way of identifying discontinuous KPs, it is doubtful whether adding them in text mining applications will be worth the extra effort. In spite of their high precision, discontinuous KPs will presumably have a relatively low recall and may also be domain dependent. The group of nouns in the "may_prevent" example in table 94 are unlikely to be useful when extracting "may_prevent" relations from the domain of Information Technology, for example. Nevertheless, it would be interesting to examine the usefulness of discovering and applying discontinuous KPs in a WWW2REL framework.

Invalid patterns The problem of invalid KPs primarily arises in the case of the two causal relation types, "induces" and "may_prevent", which can be contextually inversed and become each other's opposite. For example, the reason KPs like "causes" (see table 28) are discovered in the "may_prevent" training set can be found by examining the left contexts of the term pairs. The main culprit appears to be dosage information given in the left or middle contexts. Although "calcium <may_prevent> osteoporosis", "lack of calcium <induces> osteoporosis". More examples are given in

table 95.

Relational inversion caused by dosage information in the middle context could be avoided by disallowing nouns in the dummy position immediately preceding the KP in the discovery phase (see the search template in subsection 4.1.5). Similarly, inversion through dosage information in the left context could be avoided by disallowing modifiers of <t1> and making sure that <t1> is not a prepositional complement. As in the question of whether or not to disallow input term modifiers (see subsection 5.1.2), the effects of such filters on performance should be empirically tested so as to measure whether the possible increase in precision involves a too costly reduction in recall or is computationally costly. It may also be the case that the phenomenon of contextual inversion of causal relations is specific to the biomedical domain and that implementing filters may have counter-productive results in other domains.

8 Appendices

The appendices contain all source code for WWW2REL, for querying the UMLS Metathesaurus in various ways and for computing measures like Fleiss' kappa. They also contain lists of training term pairs, individual KP precision scores and F-score plots for most of the thesis experiments.

8.1 Conversion of BNC from SGML (SARA) to raw text

```
###unpack from DVD
tar zxvfO texts.tar.gz > BNC.tags
###remove document headers
cat BNC.tag | perl -pe 's/<bncDoc id=.{3}>/\[\1\]/;
s/<tei.*//; s/<.*?>//g;' > BNC.stripped
###one sentence per line
cat BNC.stripped | tr -s '\n' > BNC.stripped2
cat BNC.stripped2 | perl -e 'while(<>){
if(/\[(bnc.*)\]/){print "<\\bncDoc>\n<", $1,
">\n";} else{print "<s>\n", $_, "<\\s>\n";}}'
> BNC.stripped_sent
###move line 1 to last line
cat BNC.stripped_sent | perl -e 'while(<>){print unless
($.== 1);} print "<\\bncDoc>";' > BNC.stripped2
###remove SGML mark-up
cat BNC.stripped2 | perl -e 'use HTML::Entities;
while (<>){print decode_entities($_);}' > BNC.clean
###Remove unorthodox entities
cat BNC.clean | sed -r 's/&percnt;/% /g' | sed -r
's/&.{1,6};//g' > BNC.megaclean
```

8.2 vp2_log_likelihood.pl

```
#!/usr/bin/perl -w
#This script computes the log likelihood ratio as a
#measure of the degree of association between a VP and
#two different corpora
use strict;
scalar(@ARGV) == 5 or die "usage: <ref_freqs> <analysis_
freqs> <ref_size> <analysis_size> <min_freq_of_vp>";
###BNC: 110M, BIOMED: 92M
my %REF_VP = ();
open(REF,"<",$ARGV[0]);
while(<REF>){
   chomp;
   ###FORMAT: [0-9]+ [a-z -]+
   s/^[ ]+//;
   my @line = split;
   my $freq = shift(@line);
   my $vp = join(" ",@line);
   ###SKIP NONLEXICAL TOKENS
   next unless ($vp =~ /[a-zA-Z- ]+/);
   $REF_VP{$vp} = $freq;
}
close(REF);
my($e11,$e12,$e21,$e22,$o11,$o12,$o21,$o22,$c1,$c2);
my $N = $ARGV[2]+$ARGV[3];
open(AC,"<",$ARGV[1]);
while(<AC>){
   chomp;
   s/^[ ]+//;
   my @line = split;

   ###OBSERVED FREQUENCIES
   $o11 = shift(@line); #ANALYSIS CORPUS
   $o12 = $ARGV[3] - $o11; #ANALYSIS CORPUS
   next unless $o11 >= $ARGV[4]; #MIN FREQ OF VP IN AC
   my $ac_vp = join(" ",@line);

   #LOG OF ZERO IS UNDEFINED, VP MUST OCCUR IN BNC
   next unless exists($REF_VP{$ac_vp});
   $o21 = $REF_VP{$ac_vp}; #REF CORPUS
   $o22 = $ARGV[2] - $o21; #REF CORPUS
   $c1 = $o11 + $o21;
   $c2 = $o12 + $o22;

   ###ESTIMATED FREQUENCIES
```

```
   $e11 = $ARGV[3]*($c1/$N);
   $e12 = $ARGV[3]*($c2/$N);
   $e21 = $ARGV[2]*($c1/$N);
   $e22 = $ARGV[2]*($c2/$N);
   my $log_like =
   2*(($o11*log($o11/$e11))+($o12*log($o12/$e12))+
   ($o21*log($o21/$e21))+($o22*log($o22/$e22)));
   if ($o21/$ARGV[2] > $o11/$ARGV[3]){
      ###ASSOCIATION WITH BNC GETS NEGATIVE SCORES
      $log_like *= -1;
   }
   printf("%d\t%d\t%.2f\t%s\n", $o11, $o21, $log_like,
   $ac_vp);
}
close(AC);
```

8.3 vp2log_odds.pl

```
#!/usr/bin/perl -w
use strict;
scalar(@ARGV) == 5 or die "usage: <ref_freqs> <analysis_
freqs> <ref_size> <analysis_size> <min_freq_of_vp>";
###BNC: 110M, BIOMED: 92M
my %REF_VP = ();
open(REF,"<",$ARGV[0]);
while(<REF>){
   chomp;
   ###FORMAT: [0-9]+ [a-z -]+
   s/^[ ]+//;
   my @line = split;
   my $freq = shift(@line);
   my $vp = join(" ",@line);
   $REF_VP{$vp} = $freq;
}
close(REF);
open(AC,"<",$ARGV[1]);
while(<AC>){
   chomp;
   s/^[ ]+//;
   my @line = split;
   my $ac_freq = shift(@line);
   next unless $ac_freq >= $ARGV[4];
   my $ac_vp = join(" ",@line);

   ###MUST OCCUR ONCE IN THE BNC!
   next unless exists($REF_VP{$ac_vp});
```

```
   ###LO(VP) = log((ac_freq*ref_other)/(ref_freq*ac_other))
   my $ref_other = $ARGV[2] - $REF_VP{$ac_vp};
   my $ac_other = $ARGV[3] - $ac_freq;
   my $odds = ($ac_freq*$ref_other)/($REF_VP{$ac_vp}*
   $ac_other);
   printf("%d\t%d\t%.2f\t%s\n", $ac_freq, $REF_VP{$ac_vp},
   log($odds), $ac_vp);
}
close(AC);
```

8.4 umls2random_term_pairs.pl

```
#!/usr/bin/perl
# Last edited by Jakob Halskov
# on August 29 2006
#
use DBI qw(:sql_types);
use strict;
scalar(@ARGV) == 2 or die "usage: <rel_name> <sample_size>";
#Connect to the database
my $host = 'localhost';
my $db = 'UMLS';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
###GET THE TOTAL ROW NUMBER
my $sth1 = $dbh->prepare(qq{SELECT COUNT(*) FROM MRREL WHERE
RELA LIKE ?});
$sth1->execute($ARGV[0]);
my @row_total = $sth1->fetchrow_array();
###COMPUTE sample_size RANDOM OFFSETS
my @offsets = ();
my $i = 0;
while ($i < $ARGV[1]){
   my $offset = int(rand()*$row_total[0]);
   push(@offsets,$offset);
   $i++;
}
###FETCH sample_size RANDOM CUI PAIRS FOR TARGET RELATION
my $sth2 = $dbh->prepare(qq{SELECT CUI1,CUI2 FROM MRREL WHERE
RELA LIKE ? LIMIT ?,1});
###PRINT ALL TERM VARIANTS OF EACH CONCEPT
my $sth3 = $dbh->prepare(qq{SELECT STR,TTY FROM MRCONSO WHERE
CUI LIKE ? AND LAT LIKE 'ENG'});
foreach my $rand (@offsets){
   ###SPECIFY DATATYPES
   $sth2->bind_param(1,$ARGV[0],{TYPE => SQL_VARCHAR});
   $sth2->bind_param(2,$rand,{TYPE => SQL_INTEGER});
```

```
   $sth2->execute();
   my @pair = $sth2->fetchrow_array();
   ###GET ALL STRINGS OOF CONCEPT 1
   $sth3->execute($pair[0]);
   my %hooks = ();
   while (my @terms = $sth3->fetchrow_array){
      ###terms[0]: STR, terms[1]: TTY
      $hooks{lc($terms[0])}++;
   }
   ###GET ALL STRINGS OF CONCEPT 2
   $sth3->execute($pair[1]);
   my %targets = ();
   while (my @terms = $sth3->fetchrow_array){
      $targets{lc($terms[0])}++;
   }
   print $pair[0], ": ";
   foreach my $hook (keys %hooks){
      print $hook, ";";
   }
   print " => ";
   print $pair[1], ": ";
   foreach my $target (keys %targets){
      print $target, ";";
   }
   print "\n\n";
}
$sth1->finish();
$sth2->finish();
$sth3->finish();
$dbh->disconnect();
```

8.5 umls2isa_term_pairs.pl

```
#!/usr/bin/perl
use DBI;
use strict;
scalar(@ARGV) == 2 or die "usage: <CUI>
<DIRECTION:hyponyms|hypernyms>";
###Connect to the database
my $host = 'localhost';
my $db = 'UMLS';
my $user = 'jakob';
my $pass = 'halskov';
my $dbh = DBI->connect("dbi:mysql:$db:$host",
"$user", "$pass");
###FETCH ALL CONCEPT PAIRS FOR THE TARGET DIRECTION
```

```
my $sth0 = "";
if ($ARGV[1] eq "hyponyms"){
   $sth0 = $dbh->prepare("SELECT CUI2,CUI1 FROM MRREL
      WHERE RELA LIKE 'isa' AND CUI1 LIKE ?");
}
elsif ($ARGV[1] eq "hypernyms"){
   $sth0 = $dbh->prepare("SELECT CUI1,CUI2 FROM MRREL
      WHERE RELA LIKE 'isa' AND CUI2 LIKE ?");
}
my $sth1 = $dbh->prepare("SELECT STR FROM MRCONSO
   WHERE CUI LIKE ? AND LAT LIKE 'ENG'");
my %fixed_concept = ();
$sth0->execute($ARGV[0]);
while (my @pair = $sth0->fetchrow_array){
   ###GET TERM VARIANTS OF ARG1 (THE FIXED CONCEPT)
   $sth1->execute($pair[1]);
   while (my @names = $sth1->fetchrow_array){
      $fixed_concept{lc($names[0])}++;
   }
   last;
}
$sth0->execute($ARGV[0]);
my $cui_pairs = $sth0->rows;
while (my @pair = $sth0->fetchrow_array){
   ###GET TERM VARIANTS OF ARG2 (OTHER CONCEPTS)
   $sth1->execute($pair[0]);
   my %other_concept = ();
   while (my @names = $sth1->fetchrow_array){
      $other_concept{lc($names[0])}++;
   }
   foreach my $fixed (keys %fixed_concept){
      foreach my $other (keys %other_concept){
         print $fixed, " <=> ", $other, "\n";
      }
   }
}
$sth0->finish();
$sth1->finish();
$dbh->disconnect();
print STDERR $cui_pairs, " concept pairs\n";
```

8.6 umls2term_pairs.pl

```
#!/usr/bin/perl
use DBI;
use strict;
```

```
###Last modified by Jakob Halskov on Oct 19 2006
#
###This script retrieves term pairs (all variants) from
###UMLS in the target relation, but only those pairs for
###which either argument has at least one "ingredient_of"
###relation
scalar(@ARGV) == 1 or die "usage: <UMLS_REL_NAME>";
###Connect to the database
my $host = 'localhost';
my $db = 'UMLS';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
###FETCH ALL CUI PAIRS FOR TARGET RELATION
my $sth0 = $dbh->prepare("SELECT CUI1,CUI2 FROM MRREL
WHERE RELA LIKE ?");
###FETCH ONLY THOSE PAIRS THAT HAVE ACTIVE INGREDIENTS
my $sth3 = $dbh->prepare("SELECT count(*) FROM MRREL
WHERE RELA LIKE 'ingredient_of' AND CUI1 LIKE ?");
my $sth1 = $dbh->prepare("SELECT CUI2 FROM MRREL WHERE
RELA LIKE 'ingredient_of' AND CUI1 LIKE ?");
###PRINT STRING VALUES
my $sth2 = $dbh->prepare("SELECT STR,TTY FROM MRCONSO
WHERE CUI LIKE ? AND LAT LIKE 'ENG'");
my %ingredients = ();
my %effects = ();
$sth0->execute($ARGV[0]);
while (my @pair = $sth0->fetchrow_array){
   $sth3->execute($pair[1]);
   my @count = $sth3->fetchrow_array;
   ###DOES ARG HAVE ACTIVE INGREDIENT?
   if ($count[0] > 0){
      ###GET INGREDIENT CUI!
      $sth1->execute($pair[1]);
      my @ingr_cui = $sth1->fetchrow_array;
      ###GET INGREDIENT NAMES!
      $sth2->execute($ingr_cui[0]);
      %ingredients = (); ###RESET!
      while (my @ingr_name = $sth2->fetchrow_array){
         $ingredients{lc($ingr_name[0])} = $pair[1];
      }
      $sth2->execute($pair[0]);
      %effects = (); ###RESET!
      while (my @effect = $sth2->fetchrow_array){
         $effects{lc($effect[0])} = $pair[0];
      }
      foreach my $ingr (keys %ingredients){
         foreach my $eff (keys %effects){
```

```
            print $ingr, "\t", $ingredients{$ingr},
            " <=> ", $eff, "\t", $effects{$eff}, "\n";
         }
      }
   }
}
$sth0->finish();
$sth1->finish();
$sth2->finish();
$sth3->finish();
$dbh->disconnect();
```

8.7 google2snippets.pl

```
#!/usr/bin/perl
###Last modified by Jakob Halskov on Oct 19 2006
#
###Based on a list of term pairs (t1 <=> t2) and a
###minimum co-occurrence frequency, this script
###retrieves arg2 number of snippets for each pair
###if the query "t1 * t2" yields more than arg2 hits
use SOAP::Lite;
my $google_key = 'INSERT GOOGLE API KEY HERE';
my $google_wdsl = "/home/jakob/scripts/Google
Search.wsdl";
scalar(@ARGV) == 2 or die "<term_pairs_file>
<min_snippet_no>";
my $min = $ARGV[1];
my $google_search = SOAP::Lite->service("file:
$google_wdsl");
open(PAIRS,"<",$ARGV[0]);
while (<PAIRS>){
   chomp;
   ###FORMAT: t1 => t2 t2
   my @pairs = split(/ <=> /);
   my $filename1 = join("-",@pairs);
   my $filename2 = join("-",reverse @pairs);
   $filename1 =~ tr/[ ]+/_/;
   $filename2 =~ tr/[ ]+/_/;
   my $query = "allintext: \"" . $pairs[0] . " * " .
   $pairs[1] . "\"";
   for (my $offset=0;$offset<$min;$offset+=10){
      my $results = "";
      ###Google search format: key,query,start,
      ###maxResults,filter,restricts,safeSearch,
      ###language_restriction,input_encoding,
```

```
      ###output_encoding
      ###Setting filter to "true" will remove duplicate
      ###results and avoid host crowding
      eval { $results = $google_search->doGoogleSearch(
      $google_key,$query,$offset,10,"true","","false","",
      "latin1","latin1"); };
      if ($results->{estimatedTotalResultsCount} <= $min){
         print STDERR "Less than ", $min, " hits for ",
         $query, "\n";
         last;
      }
      open(OUT,">>active_$filename1");
      if ($offset==0) {
         print STDERR $results->{estimatedTotalResultsCount},
         "\t", $query, "\n";
      }
      foreach my $result (@{$results->{resultElements}}){
         print OUT $result->{snippet}, "\n";
      }
   }
   close(OUT);
   my $query = "allintext: \"" . $pairs[1] . " * " . $pairs[0]
   . "\"";
   for (my $offset=0;$offset<$min;$offset+=10){
      my $results = "";
      eval { $results = $google_search->doGoogleSearch(
      $google_key,$query,$offset,10,"true","","false","",
      "latin1","latin1"); };
      if ($results->{estimatedTotalResultsCount} <= $min){
         print STDERR "Less than ", $min, " hits for ", $query,
         "\n";
         last;
      }
      open(OUT,">>passive_$filename2");
      if ($offset == 0){
         print STDERR $results->{estimatedTotalResultsCount},
         "\t", $query, "\n";
      }
      foreach my $result (@{$results->{resultElements}}){
         print OUT $result->{snippet}, "\n";
      }
   }
   close(OUT);
}
close(PAIRS);
```

8.8 snippets2ten_fold_sets.pl

```
#!/usr/bin/perl -w
###Last modified by Jakob Halskov on Oct 19 2006
#
###Given a series of snippets files as arguments, this script
###generates test and training sets for 10-fold-validation
###Snippet file name format: (active|passive)_t1_t1-t2_t2
use strict;
use HTML::Entities;
scalar(@ARGV) > 0 or die "usage: <snippets_files>";
my %training = ();
my %history = ();
my %last_test_set = ();
my $snip_name = "";
while (scalar(@ARGV) > 0){
   &populate_training_array;
}
###EXTRACT 10 TEST SETS
my $snip_count = keys %training;
my $tenth = int($snip_count/10);
print STDERR "Training set: ", $snip_count, " snippets\n";
print STDERR "Test sets: ", $tenth, " snippets\n";
my $j = 0;
while ($j < 100){
   my $i = 0;
   ###EMPTY LAST_TEST_SET
   %last_test_set = ();
   open(TEST,">>test_$j");
   while($i<$tenth){
      my($snip_name,$snip) = each %training;
      ###HAS THE SNIPPET BEEN USED AS TEST BEFORE?
      if (!exists($history{$snip_name})){
      $history{$snip_name}++;
      print TEST $snip_name, "\n";
      ###SAVE TEST SET IN HASH
      $last_test_set{$snip_name} = $snip;
      ###DELETE FROM TOTAL CORPUS
      delete($training{$snip_name});
      $i++;
   }
}
close(TEST);
###PUT TEST SET BACK
%training = (%training,%last_test_set);
###PRINT CORRESPONDING TRAINING SET (90%)
```

```
open(TRAIN,">>train_$j");
while (my($snip_name,$snip) = each %training){
   ###SNIPPET MUST NOT BE PART OF 10% TEST SET
   if(!exists($last_test_set{$snip_name})){
      print TRAIN $snip, "\n";
      }
   }
   close(TRAIN);
   $j+=10;
}
###SUBROUTINES###
sub populate_training_array {
my $file = shift(@ARGV);
my @pair = split(/\-/,$file);
my ($term1,$term2) = ("","");
if ($pair[0] =~ s/active_//){
   $term1 = $pair[0];
   $term2 = $pair[1];
   $snip_name = $term1 . "-" . $term2;
}
elsif ($pair[0] =~ s/passive_//){
   $term2 = $pair[0];
   $term1 = $pair[1];
   $snip_name = $term2 . "-" . $term1;
}
###LOAD TEXT
open(SNIPPETS,"<$file");
undef $/;
my $snip = <SNIPPETS>;
$/ = "\n";
close(SNIPPETS);
###DECODE ENTITIES
decode_entities($snip);
###REMOVE MARK-UP
$snip =~ s/<[^ ]+>//g;
$snip =~ s/[ ]+/ /g;
###TAG TERMS
$term1 =~ tr/_/ /;
$term2 =~ tr/_/ /;
$snip =~ s/$term1/<term1>/sig;
$snip =~ s/$term2/<term2>/sig;
###NB: HASH KEYS MUST BE UNIQUE
$training{$snip_name} = $snip;
}
```

8.9 learn_knowledge_patterns.sh

```
#!/bin/bash
###Last modified by Jakob Halskov on Oct 19 2006
#
###This bash script POS tags and chunks the training
###snippets. It runs the perl script extract_middle_
###context_VPs.pl to extract candidate knowledge
###patterns and filters these patterns by lowercasing,
###removing punctuation marks and so on. Results are
###two lists of pattern candidates (+/- frequencies)
for i in snippets_*; do tree-tagger-english $i > tag_$i;
done
for i in tag_*; do cat $i | sed -r 's/(.\tSENT\t.)/\1\n/' |
yamcha -m ~/data/wsj_based.model > chunk_$i; done
for i in chunk*; do extract_middle_context_VPs.pl $i active
y | cut -f1 | sed -r 's/^$/_/' | tr '\n' ' ' | tr '_' '\n' |
tr '[:upper:]' '[:lower:]' | tr -d '[,.;%:()-]' | sed -r
's/[ ]+/ /g' | sed -r 's/^[ ]+//' | sed -r 's/[ ]+$//' |
egrep -rv '^[ ]*$' | sort | uniq -c | sort -rn > kp_$i; done
for i in kp_chunk*; do cat $i | sed -r 's/^[ ]+[0-9]+ /\+/'
| sed -r 's/[ ]+/ \+/g' > no_frq_$i; done
```

8.9.1 extract_middle_context_VPs.pl

```
#!/usr/bin/perl -w
###Last modified by Jakob Halskov on Oct 19 2006
#
###This script takes as input, POS tagged and chunked
###snippet files.As output it prints flexible VPs as
###defined by the regular expressions below
use strict;
scalar(@ARGV) == 3 or die "usage: <snippets_file>
<active|passive> <print_dummy:y|n>";
open(SNIPPETS,"<$ARGV[0]");
undef $/;
my $snip = <SNIPPETS>;
$/ = "\n";
close(SNIPPETS);
###TERM PAIRS ARE TAGGED <term1> AND <term2> OR VICE VERSA
my $vp = "(?:[^\n]+\t.-VP\n)+(?:[^\n]+\tB-PP\n)?";
my $np = "(?:[^\n]+\t.-NP\n)*"; ###OPTIONAL
my $dummy = "(?:[^\n]+\n){0,2}"; ###MAX 2
if ($ARGV[2] =~ /y/){
   $vp = "(" . $dummy . $vp . ")";
}
```

```
else {
   $vp = $dummy . "(" . $vp . ")";
}
##ALLOW COMPLEMENTIZERS, PARENTHESES AND SO ON AFTER TERM1
if ($ARGV[1] =~ /active/i){
   while($snip =~ /<term1>[^\n]+\n$vp$np<term2>/sig){
      print $1, "\n";
   }
}
elsif ($ARGV[1] =~ /passive/i){
   while($snip =~ /<term2>[^\n]+\n$vp$np<term1>/sig){
      print $1, "\n";
   }
}
```

8.10 form_queries.pl

```
#!/usr/bin/perl -w
###Last modified by Jakob Halskov on Oct 19 2006
#
###Given a list of term pairs and pattern candidates
###(with frequency numbering), this script produces Google
###queries, which allow for flexibility and active/passive
###alternations
use strict;
scalar(@ARGV) == 4 or die "<term_pairs> <num_patterns>
<active|passive> <flex:y|n>";
my @pairs = ();
open(PAIRS,"<",$ARGV[0]);
while(<PAIRS>){
   chomp;
   ###FORMAT: t1_t1-t2_t2
   tr/_/ /;
   s/^/\+/;
   s/ / \+/g;
   push(@pairs,$_);
}
close(PAIRS);
my @queries = ();
open(PATTERNS,"<",$ARGV[1]);
while(<PATTERNS>){
   chomp;
   ###FORMAT: frq_rank_number\tKP\n
   my @pat = split(/\t/);
   foreach my $pair (@pairs){
      my @terms = split(/-/,$pair);
```

```
      if ($ARGV[3] =~ /y/i && $ARGV[2] =~ /active/i){
         ###ALLOW ONLY RIGHT FLEXIBILITY
         print "\"" . $terms[0] . " " . $pat[1] .
         " * +" . $terms[1] . "\"\t" . $pat[1] . "\t" .
         $pat[0] . "\n";
      }
      elsif ($ARGV[3] =~ /n/i && $ARGV[2] =~ /active/i){
         print "\"" . $terms[0] . " " . $pat[1] . " +" .
         $terms[1] . "\"\t" . $pat[1] . "\t" . $pat[0] . "\n";
      }
      elsif($ARGV[2] eq "passive"){
         print "\"+" . $terms[1] . " " . $pat[1] . " " .
         $terms[0] . "\"\t" . $pat[1] . "\t" . $pat[0] . "\n";
      }
   }
}
close(PATTERNS);
```

8.11 google2frequencies.pl

```
#!/usr/bin/perl
###Last modified by Jakob Halskov on Oct 19 2006
#
###Based on a list of queries (term1-kp-term2), this
###script retrieves query frequencies from Google
use WWW::Mechanize;
use strict;
scalar(@ARGV) == 1 or die "<queries>";
my $mech = WWW::Mechanize->new();
###MASCARADE AS INTERNET EXPLORER
$mech->agent_alias('Windows IE 6');
open(QUERIES,"<",$ARGV[0]);
open(OUT,">>stats_$ARGV[0]");
while (<QUERIES>){
   chomp;
   ###FORMAT: query\tKP\tNUM\n
   my @line = split(/\t/);
   my $query = $line[0];
   my $url = "http://www.google.com/search?q=allintext:
   $query";
   $mech->get($url);
   my $text = $mech->content();
   if ($text =~ /of about <b>([0-9,]+)<\/b> for/sig){
      my $hits = $1;
      $hits =~ tr/,//d;
      print OUT $hits, "\t", $line[1], "\t", $query,
```

```
      "\t", $line[2], "\n";
      print STDERR $hits, "\t", $line[1], "\t", $query,
      "\t", $line[2], "\n";
   }
   else {
      print OUT "0", "\t", $line[1], "\t", $query, "\t",
      $line[2], "\n";
      print STDERR "0", "\t", $line[1], "\t", $query,
      "\t", $line[2], "\n";
   }
}
close(QUERIES);
close(OUT);
```

8.12 normalize_and_compute_pattern_precision.pl

```
#!/usr/bin/perl -w
###Last modified by Jakob Halskov on Oct 19 2006
#
###Input to this script is: two sets of query frequency
###statistics for term1-KP-term2 tuples where the term1-
###term2 pairs instantiate target and non-target relations,
###respectively. It also needs the co-occurrence statistics
###(term1 * term2) of the term pairs to normalize the hit
###counts. Output is precision scores for the KPs involved
###in this test (1 out of 10)
#
use strict;
scalar(@ARGV) == 4 || die "<RIGHT_COOCC> <WRONG_COOCC>
<RIGHT_STATS> <WRONG_STATS>";
my %abs_r = ();
my %abs_a = ();
my %right_co = ();
###LOAD CO-OCCURRENCE STATISTICS OF 4 CORRECT PAIRS
open(RIGHT,"<$ARGV[0]");
while(<RIGHT>){
   chomp;
   my @line = split(/\t/);
   ###FORMAT: FRQ\tt1_t1-t2_t2\n
   $line[1] =~ s/\-/\.\*/;
   $line[1] =~ s/_/ \\+/g; ##REMEMBER 'g' FLAG HERE
   $line[1] =~ s/^/\\+/;
   $right_co{$line[1]} = $line[0];
}
close(RIGHT);
###LOAD CO-OCCURRENCE STATISTICS OF 4 INCORRECT PAIRS
```

```
my %wrong_co = ();
open(WRONG,"<$ARGV[1]");
while(<WRONG>){
   chomp;
   my @line = split(/\t/);
   ###FORMAT: FRQ\tt1_t1-t2_t2\n
   $line[1] =~ s/\-/\.\*/;
   $line[1] =~ s/_/ \\+/g; ##NB: REMEMBER 'g' FLAG HERE
   $line[1] =~ s/^/\\+/;
   $wrong_co{$line[1]} = $line[0];
}
close(WRONG);
###LOAD QUERY FREQUENCY STATISTICS FOR "CORRECT"
###TERM-KP-TERM TUPLES
my %RIGHT = ();
my %ALL = ();
open(RIGHT_STATS,"<$ARGV[2]");
while(<RIGHT_STATS>){
   chomp;
   ###FORMAT: FRQ\tKP\tQUERY\tKP-RANK\n
   my @line = split(/\t/);
   $abs_a{$line[1]} += $line[0];
   $abs_r{$line[1]} += $line[0];
   if ($line[1] =~ /\+&/){
      next;
   }
   foreach my $pair (keys %right_co){
      if($line[2] =~ /$pair/){
         $RIGHT{$line[1]} += ($line[0]/$right_co{$pair});
         $ALL{$line[1]} += ($line[0]/$right_co{$pair});
      }
   }
}
close(RIGHT_STATS);
###LOAD QUERY FREQUENCY STATISTICS FOR "INCORRECT"
###TERM-KP-TERM TUPLES
my %WRONG = ();
open(WRONG_STATS,"<$ARGV[3]");
while(<WRONG_STATS>){
   chomp;
   ###FORMAT: FRQ\tKP\tQUERY\tKP-RANK\n
   my @line = split(/\t/);
   $abs_a{$line[1]} += $line[0];
   if ($line[1] =~ /\+&/){
      next;
   }
```

```
   foreach my $pair (keys %wrong_co){
      if($line[2] =~ /$pair/){
         $WRONG{$line[1]} += ($line[0]/$wrong_co{$pair});
         $ALL{$line[1]} += ($line[0]/$wrong_co{$pair});
      }
   }
}
close(WRONG_STATS);
###PRINT KP PRECISION BASED ON NORMALIZED HIT COUNTS
my $right = 0;
while (my($pat,$rf) = each %ALL){
   if (!exists $RIGHT{$pat}){
      $right = 0;
   }
   else {
      $right = $RIGHT{$pat};
   }
   unless ($rf == 0){
      my $prec = ($right/$rf)*100;
      print $prec, "\t", $abs_r{$pat}, "\t",
      $abs_a{$pat}-$abs_r{$pat}, "\t", $pat, "\n";
   }
}
```

8.12.1 compute_average_precision.pl

```
#!/usr/bin/perl -w
###Last modified by Jakob Halskov on Oct 19 2006
#
###This script simply computes the average precision
###of all KPs across the 10-fold-validation tests
#
use strict;
scalar(@ARGV) == 1 or die "usage: <10_PREC_FILES>";
my %total_prec = ();
my %occurrence = ();
my %total_freq = ();
open(PREC,"<$ARGV[0]");
while (<PREC>){
   chomp;
   ###FORMAT: PREC\tFRQ_R\tFRQ_W\tKP\n
   my @stats = split(/\t/);
   $total_prec{$stats[3]} += $stats[0];
   $occurrence{$stats[3]}++;
   $total_freq{$stats[3]} += $stats[1];
   $total_freq{$stats[3]} += $stats[2];
```

```
}
close(PREC);
while(my($kp,$prec) = each %total_prec){
   printf("%.2f\t%.1f\t%s\n", $prec/$occurrence{$kp},
   $total_freq{$kp}/10, $kp);
}
```

8.13 kp_discovery_power.pl

```
#!/usr/bin/perl -w
###Last modified by Jakob Halskov on Oct 19 2006
#
###For each of the 10 tests, this script takes the
###list of patterns learned during training (36/40)
###and during testing (4/40) as well as two integers
###specifying the topX most frequent training and
###test patterns to compare when computing pattern
###"discovery power"
use strict;
scalar(@ARGV) == 4 || die "<train_pat> <test_pat>
<topX_train> <topX_test>";
my %training_pats = ();
my @test_pats = ();
open(TRAIN,"<$ARGV[0]");
while(<TRAIN>){
   chomp;
   my @line = split(/\t/);
   ###FORMAT: KP_RANK\tKP\n
   if($line[0] <= $ARGV[2]){
      $training_pats{$line[1]}++;
   }
}
close(TRAIN);
my $i = 1;
open(TEST,"<$ARGV[1]");
while(<TEST>){
   if ($i > $ARGV[3]){
   last;
   }
   else {
      chomp;
      push(@test_pats,$_);
      $i++;
   }
}
close(TEST);
```

```
my $hit = 0;
foreach my $test (@test_pats){
   if(exists($training_pats{$test})){
      print STDERR "HIT: ", $test, "\n";
      $hit++;
   }
}
print "RATIO: ", $hit/$ARGV[3], "\n";
```

8.14 umls2synonym_pairs.pl

```
#!/usr/bin/perl
# Last edited by Jakob Halskov
# on August 29 2006
#
use DBI qw(:sql_types);
use strict;
scalar(@ARGV) == 1 or die "usage: <rel_name>";
#Connect to the database
my $host = 'localhost';
my $db = 'UMLS';
my $user = 'jakob';
my $pass = 'halskov';
my $dbh = DBI->connect("dbi:mysql:$db:$host", "$user", "$pass");
##########FETCH CUI PAIRS FOR THE TARGET RELATION
my $sth2 = $dbh->prepare(qq{SELECT CUI1,CUI2 FROM MRREL
WHERE RELA LIKE ?});
##########FETCH TERM VARIANTS FOR TARGET CONCEPTS
my $sth3 = $dbh->prepare(qq{SELECT STR,TTY FROM MRCONSO
WHERE CUI LIKE ? AND LAT LIKE 'ENG'});
##########WE ONLY WANT VARIANTS EXPLICITLY MARKED AS SYNONYMS
my $syn = "(MTH_)?(SY|SS|RSY|RLS|SYGB|USY|ORS|ONS|OBS|OES|
NSY|AS|BSY|ESY|IS)";
$sth2->execute($ARGV[0]);
my %pairs = ();
while (my @cuis = $sth2->fetchrow_array){
    $sth3->execute($cuis[0]);
    my $has_pn = 0;
    my $pn = "";
    while (my @terms = $sth3->fetchrow_array){
        if ($terms[1] =~ /PN/){
            $pn = $terms[0];
            $has_pn = 1;
        }
        elsif ($has_pn == 1 && $terms[1] =~ /$syn/){
            my $pair = lc($pn) . "," . lc($terms[0]);
```

```
            $pairs{$pair}++;
        }
    }
    $has_pn = 0;
    $pn = "";
    $sth3->execute($cuis[1]);
    while (my @terms = $sth3->fetchrow_array){
        if ($terms[1] =~ /PN/){
            $pn = $terms[0];
            $has_pn = 1;
        }
        elsif ($has_pn == 1 && $terms[1] =~ /$syn/){
            my $pair = lc($pn) . ",” . lc($terms[0]);
            $pairs{$pair}++;
        }
    }
}
$sth2->finish();
$sth3->finish();
$dbh->disconnect();
foreach (keys %pairs){
   print $_, "\n”;
}
```

8.15 term_and_kp2snippets.pl

```
#!/usr/bin/perl -w
###Last modified by Jakob Halskov on Oct 23 2006
#
###Given an input term and a list of reliable KPs
###for the target relation this script retrieves
###relation instances (NPs in the term-KP-NP triplet)
use SOAP::Lite;
use DBI;
use strict;
my $google_wdsl = "/home/jakob/scripts/GoogleSearch.wsdl";
my $google_key = 'INSERT DEVELOPER KEY HERE!';
scalar(@ARGV) == 3 or die "<t1> <relation> <term_pos:l|r>";
my $term = $ARGV[0];
my $google_search = SOAP::Lite->service("file:$google_wdsl");
#Connect to the database
my $host = 'localhost';
my $db = 'www2rel';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
###GET THE PATTERNS FOR THE RELATION FROM THE DATABASE
my $sth1 = $dbh->prepare("SELECT id,kp FROM patterns WHERE
```

```
relation LIKE ?");
$sth1->execute($ARGV[1]);
my $t = $term;
$t =~ tr/ /_/;
while(my @line = $sth1->fetchrow_array()){
   ###FORMAT: $line[0]: id, $line[1]: KP
   my $pat = $line[1];
   $pat =~ tr/ /_/;
   my $query = "allintext: ";
   if ($ARGV[2] =~ /l/i){
      $query = "\"" . $term . " " . $line[1] . "\"";
      open(OUT,">>$t-$pat-$line[0]");
   }
   elsif ($ARGV[2] =~ /r/i){
      $query = "\"" . $line[1] . " " . $term . "\"";
      open(OUT,">>$pat-$line[0]-$t");
   }
   print STDERR $query, "\n";
   for (my $offset=0;$offset<100;$offset+=10){
      print STDERR $offset, "\n";
      my $results = $google_search->doGoogleSearch(
      $google_key,$query,$offset,10,"true","","false",
      "","latin1","latin1");
      last unless @{$results->{resultElements}};
      foreach my $result (@{$results->{resultElements}}){
         print OUT $result->{snippet}, "\n";
      }
   }
   close(OUT);
}
$sth1->finish();
$dbh->disconnect();
```

8.16 prepare_corpus.pl

```
#!/usr/bin/perl -w
use strict;
use HTML::Entities;
scalar(@ARGV) > 0 or die "usage: <snippets_files>";
my $file_out = "";
while (scalar(@ARGV) > 0){
   my $file = shift(@ARGV);
   ###FILENAME1: active_term-kp-kpid
   ###FILENAME2: passive_kp-kpid-term
   my @tuple = split(/\-/,$file);
   my($term,$kp,$kp_id) = ("","","");
```

```
   if ($tuple[0] =~ s/active_//){
      $term = $tuple[0];
      $kp = $tuple[1];
      $kp_id = $tuple[2];
   }
   elsif ($tuple[0] =~ s/passive_//){
      $kp = $tuple[0];
      $kp_id = $tuple[1];
      $term = $tuple[2];
   }
   ###LOAD TEXT
   open(SNIPPETS,"<$file");
   undef $/;
   my $snip = <SNIPPETS>;
   $/ = "\n";
   close(SNIPPETS);

   ###DECODE ENTITIES
   decode_entities($snip);

   ###REMOVE MARK-UP
   $snip =~ s/<[^ ]+>//g;
   $snip =~ s/[ ]+/ /g;
   $file_out = $term . "_corpus";
   open(OUT,">>$file_out");

   ###TAG TERMS
   $term =~ tr/_/ /;
   my $regexp = "<kp_" . $kp_id . ">";
   $kp =~ tr/_/ /;
   $snip =~ s/$term/<term>/sig;
   $snip =~ s/$kp/$regexp/sig;
   print OUT $snip;
   close(OUT);
}
`~/TreeTagger/cmd/tree-tagger-english $file_out >
tag_$file_out`;
my $temp = `cat tag_$file_out`;
$temp =~ s/(.\tSENT\t.)/$1\n/sg;
open(OUT,">tag_$file_out");
print OUT $temp;
close(OUT);
system("cat tag_$file_out | yamcha -m
~/bins/wsj_based.model > chunk_$file_out");
```

8.17 extract_relation_instances_store_in_database.pl

```
#!/usr/bin/perl -w
use strict;
use WWW::Mechanize;
use DBI;
scalar(@ARGV) == 4 or die "usage: <chunked_corpus>
<term> <relation> <active|passive>";
#RECORD THE DOWNLOAD DATE
my $dl_date = `date +%Y-%m-%d`;
my $mech = WWW::Mechanize->new();
$mech->agent_alias('Windows IE 6');
my $corpusname = $ARGV[0];
$corpusname =~ tr/[ ]+/_/;
open(SNIPPETS,"<$corpusname");
undef $/;
my $snip = <SNIPPETS>;
$/ = "\n";
close(SNIPPETS);
my $corpus = lc($snip); ###LOWER CASE EVERYTHING
###CONTEXTS ARE TAGGED: <term> <kp_[0-9]+> [...]
my $np = "(?:[^\n]+\t.-np\n)*[^\n]+\t(?:vvg|n[np]s?)
\t[^\n]+\n"; #HEAD MUST BE A NOUN OR GERUND
my $prep = "[^\n]+\tb-pp\n";
my $pp = $prep . $np;
my $dummy = "(?:[^\n]+\n){0,2}";
my $np_mod = "(?:[^\n]+\t.-np\n)*";
###Connect to the database
my $host = 'localhost';
my $db = 'www2rel';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
###Relation instance IDs are auto incremented in database
my $sth1 = $dbh->prepare("INSERT INTO candidates
(head,sample_frq,np,term,relation,dl_date,kp_range)
VALUES(?,?,?,?,?,?,?)");
###np2kp mappings are not auto incremented
my $sth2 = $dbh->prepare("INSERT INTO np2kp (np_id,kp_id,
freq,position) VALUES(?,?,?,?)");
my %candidates_no_pom = ();
my ($regexp_no_pp,$regexp_pp) = ("","");
###
#REMEMBER QUERY FLEXIBILITY
###
if ($ARGV[3] =~ /active/i){
   $regexp_no_pp = "<term>[^\n]+\n$dummy<kp_([0-9]+)>
   [^\n]+\n($np)(?!$prep)";
```

```
   $regexp_pp = "<term>[^\n]+\n$dummy<kp_([0-9]+)>
   [^\n]+\n($np)($prep$np)";
}
elsif ($ARGV[3] =~ /passive/i){
   $regexp_no_pp = "($np)<kp_([0-9]+)>
   [^\n]+\n$np_mod<term>";
   $regexp_pp = "($np)($prep$np)<kp_([0-9]+)>
   [^\n]+\n$np_mod<term>";
}
while($corpus =~ /$regexp_no_pp/sig){
   my($kp_id,$np) = ("","");
   if ($ARGV[3] =~ /active/i){
      ($kp_id,$np) = ($1,$2);
   }
   else {
      ($np,$kp_id) = ($1,$2);
   }
   $np =~ s/[^\n]+\tdt\t[^\n]+\n//g;
   if ($np =~ /\tvvg\t/){
      $np =~ s/([^\n]+)(\tvvg\t)[^\n]+(\t[^\n]+)/
      $1$2$1$3/g;
   }
   open(TMP_simple,">tmp_simple");
   print TMP_simple $np;
   close(TMP_simple);
   $np = `cat tmp_simple | cut -f1 | tr '\n' '_' |
   tr '_' ' '`;
   $np =~ s/[ ]+$//;
   $np =~ s/^[ ]+//;
   ###DELETE ALL NON-ALPHABETICALS
   $np =~ tr/0-9a-zA-Z -//cd;
   my @np = split(/[ ]+/,$np);
   $candidates_no_pom{$np[-1]}->{$np} .= $kp_id . "\t";
}
if ($ARGV[3] =~ /active/i){
   while($corpus =~ /$regexp_pp/sig){
      my($kp_id,$np,$pom) = ($1,$2,$3);
      ###TRANSFORM FX. "BLEEDING IN THE STOMACH" =>
      "STOMACH BLEEDING"
      ###DELETE DETERMINERS AND PREPOSITIONS!
      $pom =~ s/[^\n]+\t(dt|in|to)\t[^\n]+\n//g;
      ###WE DON'T WANT BASE FORM OF GERUND HEADS!
      ###(fx. bleeding -> bleed)
      if ($pom =~ /\tvvg\t/){
         $pom =~ s/([^\n]+)(\tvvg\t)[^\n]+(\t[^\n]+)/
         $1$2$1$3/g;
```

```
      }
      $np =~ s/[^\n]+\tdt\t[^\n]+\n//g;
      if ($np =~ /\tvvg\t/){
         $np =~ s/([^\n]+)(\tvvg\t)[^\n]+(\t[^\n]+)/
         $1$2$1$3/g;
      }
      ###PRINT TO TMP FILE TO USE THE UNIX "CUT" TOOL
      open(TMP_pom,">tmp_pom");
      print TMP_pom $pom;
      close(TMP_pom);
      ###USE WORD-FORMS (COLUMN 1) NOT LEMMAS (COLUMN 3)
      my $np_pom = `cat tmp_pom | cut -f1 | tr '\n' '_'
      | tr '_' ' '`;
      $np_pom =~ s/[ ]+$//;
      $np_pom =~ s/^[ ]+//;
      ###DELETE ALL NON-ALPHABETICALS
      $np_pom =~ tr/0-9a-zA-Z -//cd;
      ###SAME AS ABOVE
      open(TMP_simple,">tmp_simple");
      print TMP_simple $np;
      close(TMP_simple);
      my $np_simple = `cat tmp_simple | cut -f1 |
      tr '\n' '_' | tr '_' ' '`;
      $np_simple =~ s/[ ]+$//;
      $np_simple =~ s/^[ ]+//;
      ###DELETE ALL NON-ALPHABETICALS
      $np_simple =~ tr/0-9a-zA-Z -//cd;
      my @np = split(/[ ]+/,$np_simple);
      ###SKIP PP IF TRANSFORMED NP IS NOT IN
      ###THE HASH %candidates_no_pom!
      my $np_transformed = $np_pom . " " . $np_simple;
      if (exists($candidates_no_pom{$np[-1]}->
         {$np_transformed})){
            $candidates_no_pom{$np[-1]}->{$np_transformed}
            .= $kp_id . "\t";
      }
      else {
         $candidates_no_pom{$np[-1]}->{$np_simple}
         .= $kp_id . "\t";
      }
   }
}
###STORE RELATION INSTANCE IN DATABASE
while(my($head,$np_hash_ref) = each %candidates_no_pom){
   ###VALUE FORMAT: "gi bleeding" => kp_id\tkp_id\t ...
   print STDERR $head, "\n";
```

```
   foreach my $np (keys %{$np_hash_ref}){
      print STDERR "\t", $np, "\n";
      ###GET NP SAMPLE FREQUENCY
      my @kps = split(/\t/,$np_hash_ref->{$np});
      my $np_sample_freq = scalar(@kps);
      my %diff_KPs = ();
      foreach my $kp_id (@kps){
         $diff_KPs{$kp_id}++;
      }
      my $kp_range = scalar(keys %diff_KPs);
      ###STORE NP IN candidates TABLE
      $sth1->execute($head,$np_sample_freq,$np,
      $ARGV[1],$ARGV[2],$dl_date,$kp_range);
      ###GET LAST INSERTED ID
      my $auto_id = $dbh->{'mysql_insertid'};
      ###STORE NP->KP MAPPINGS IN np2kp TABLE
      while(my($kp_id,$f) = each %diff_KPs){
         $sth2->execute($auto_id,$kp_id,$f,$ARGV[3]);
      }
   }
}
$sth1->finish();
$sth2->finish();
$dbh->disconnect();
```

8.18 Fleiss' kappa measure for inter-rater reliability

```
#!/usr/bin/perl -w
use strict;
use DBI;
scalar(@ARGV) == 4 or die "usage: <term> <relation>
<frq_gt_X> <strict|lax?>";
#Connect to the database
my $host = 'localhost';
my $db = 'www2rel';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
my ($sth0,$sth1) = ("","");
if ($ARGV[1] eq "isa"){
    $sth1 = $dbh->prepare("SELECT judgment FROM
    candidates WHERE term LIKE ? AND relation LIKE
    'isa_%' AND judgment IS NOT NULL GROUP BY np");
}
else {
    $sth1 = $dbh->prepare("SELECT judgment FROM
    candidates WHERE term LIKE ? AND relation LIKE ?
    AND sample_frq > ? AND judgment IS NOT NULL");
```

```
}
my @row_agr = ();
my @col_sums = ();
my $counter = 0;
if ($ARGV[1] eq "isa"){
    $sth1->execute($ARGV[0]);
}
else {
    $sth1->execute($ARGV[0],$ARGV[1],$ARGV[2]);
}
my $no_of_instances = $sth1->rows();
#NB: THERE ARE FOUR JUDGES
my $matrix_sum = $no_of_instances*4;
while(my @judgments = $sth1->fetchrow_array){
    my @ratings = split(/,/,$judgments[0]);
    #INSERT "DON'T KNOW" IF JUDGMENT IS MISSING
    while(scalar(@ratings) < 4){
        push(@ratings,"2");
    }

    ###COMPUTE COLUMN AND ROW SUMS
    my $mean_sum = 0;
    my @cell_sum = (0,0,0);
    foreach (@ratings){
        if ($ARGV[3] eq "lax"){
            if (/[14]/){
                $col_sums[0]++;
                $cell_sum[0]++;
            }
            elsif (/[23]/){
                $col_sums[1]++;
                $cell_sum[1]++;
            }
        }
        elsif ($ARGV[3] eq "strict"){
            if (/[14]/){
                $col_sums[0]++;
                $cell_sum[0]++;
            }
            elsif (/2/){
                $col_sums[1]++;
                $cell_sum[1]++;
            }
            elsif (/3/){
                $col_sums[2]++;
                $cell_sum[2]++;
```

```
            }
        }
    }
    foreach (@cell_sum){
        $mean_sum += (($_*$_)-$_);
    }
    $row_agr[$counter] = $mean_sum/(4*(4-1));
    $counter++;
}
$sth1->finish();
$dbh->disconnect();
my $pi_sum = 0;
foreach (@row_agr){
    $pi_sum += $_;
}
my $prob_mean = ($pi_sum*4*(4-1))/
($no_of_instances*4*(4-1));
my $prob_obs = 0;
foreach (@col_sums){
    $prob_obs += (($_/$matrix_sum)*($_/$matrix_sum));
}
my $kappa = ($prob_mean-$prob_obs)/(1-$prob_obs);
print "Kappa for ", $ARGV[0], " ", $ARGV[1], ": ",
$kappa, "\n";
print $no_of_instances, " instances in experiment\n";
```

8.19 compute_PRF.pl

```
#!/usr/bin/perl -w
use strict;
use DBI qw(:sql_types);
scalar(@ARGV) == 6 or die "usage: <term> <relation>
<algo:frq|kpr|pkpr|fkpr|pkpr_bnc|fkpr_bnc|kpr_bnc>
<count_4_as_1:y|n> <only_drugs:y|n> <frq_gt_X>";
#Connect to the database
my $host = 'localhost';
my $db = 'www2rel';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
###LOAD ALL CANDIDATES INTO HASH AND ADD AVE. JUDGMENT
my %rated_candidates = ();
my $sth1 = $dbh->prepare("SELECT np,judgment FROM
candidates WHERE term LIKE ? AND relation LIKE ? AND
sample_frq > ? AND judgment IS NOT NULL");
my $missing = 0;
$sth1->execute($ARGV[0],$ARGV[1],$ARGV[5]);
while(my @candidate = $sth1->fetchrow_array){
```

```
    my @ratings = split(/,/,$candidate[1]);
    ###INSERT "DON'T KNOW" IF JUDGMENT IS MISSING
    while(scalar(@ratings) < 4){
        push(@ratings,"2");
        $missing++;
    }
    ###STORE BY AVERAGE RATING!
    my $sum_rating = 0;
    foreach (@ratings){
        if ($ARGV[3] eq "y" && $_ =~ /4/){
            $sum_rating += 1;
        }
        else {
            $sum_rating += $_;
        }
    }
    $rated_candidates{$candidate[0]} = sprintf("%.2f",
    ($sum_rating/4));
}
$sth1->finish();
###LOAD ALL CANDIDATES IN A PARTICULAR ORDER
my $sth2 = "";
my %BNC = ();
if ($ARGV[2] eq "frq"){
    $sth2 = $dbh->prepare("SELECT np FROM candidates
    WHERE term LIKE ? AND relation LIKE ? AND sample_frq
    > ? AND judgment IS NOT NULL ORDER BY sample_frq
    DESC,kp_range DESC");
}
elsif ($ARGV[2] eq "kpr"){
    $sth2 = $dbh->prepare("SELECT np FROM candidates
    WHERE term LIKE ? AND relation LIKE ? AND sample_frq
    > ? AND judgment IS NOT NULL ORDER BY kp_range DESC,
    sample_frq DESC");
}
elsif ($ARGV[2] eq "fkpr"){
    $sth2 = $dbh->prepare("SELECT np FROM candidates
    WHERE term like ? AND relation like ? AND sample_frq > ?
    AND judgment IS NOT NULL ORDER BY sample_frq*kp_range
    DESC");
}
elsif ($ARGV[2] eq "pkpr"){
    $sth2 = $dbh->prepare("SELECT np,kp_range+kp_range_passive
    FROM candidates WHERE term LIKE ? AND relation LIKE ? AND
    sample_frq > ? AND judgment IS NOT NULL ORDER BY
    kp_range+kp_range_passive DESC");
```

```
}
elsif ($ARGV[2] =~ /.*_bnc/i){
    $sth2 = $dbh->prepare("SELECT np,sample_frq,kp_range,
    kp_range_passive FROM candidates WHERE term LIKE ? AND
    relation LIKE ? AND sample_frq > ? AND judgment IS NOT
    NULL");
    open(BNC,"<frq_BNC");
    while (<BNC>){
        chomp;
        my @line = split(/[ ]+/);
        $BNC{$line[0]} = $line[1];
    }
    close(BNC);
}
$sth2->execute($ARGV[0],$ARGV[1],$ARGV[5]);
my @system_candidates_ordered = ();
my %system_candidates = ();
while(my @candidate = $sth2->fetchrow_array){
    if ($ARGV[2] eq "fkpr_bnc"){
        ###FORMAT:id,sample_frq,kpr
        if (exists($BNC{$candidate[0]})){
            $system_candidates{$candidate[0]} =
            (($candidate[1]*$candidate[2])/
            log($BNC{$candidate[0]}));
        }
        else{
            $system_candidates{$candidate[0]} =
            $candidate[1]*$candidate[2];
        }
    }
    elsif ($ARGV[2] eq "kpr_bnc"){
        ###FORMAT:id,sample_frq,kpr
        if (exists($BNC{$candidate[0]})){
            $system_candidates{$candidate[0]} =
            ($candidate[2]/log($BNC{$candidate[0]}));
        }
        else{
            $system_candidates{$candidate[0]} =
            $candidate[2];
        }
    }
    elsif ($ARGV[2] eq "pkpr_bnc"){
        ###FORMAT:id,sample_frq,kpr,pkpr
        if (exists($BNC{$candidate[0]}) &&
           defined($candidate[3])){
           $system_candidates{$candidate[0]} =
```

```
            ($candidate[2]+$candidate[3])/
            log($BNC{$candidate[0]});
        }
        else{
            $system_candidates{$candidate[0]} =
            $candidate[2];
        }
    }
    else {
        push(@system_candidates_ordered,$candidate[0]);
    }
}
$sth2->finish();
$dbh->disconnect();
###ORDER SYSTEM CANDIDATES IF REQUIRED
if ($ARGV[2] =~ /.*_bnc/i){
     foreach (sort { $system_candidates{$b} <=>
        $system_candidates{$a} } keys %system_candidates){
        push(@system_candidates_ordered,$_);
     }
 }
###COMPUTE MAX RECALL OF CANDIDATES
my $all_correct = 0;
if ($ARGV[4] eq "y"){
    for (my $i=0;$i<scalar(@system_candidates_ordered);$i++){
        if ($rated_candidates{$system_candidates_ordered[$i]}
           == 4){
           $all_correct++;
        }
    }
}
else {
    for (my $i=0;$i<scalar(@system_candidates_ordered);$i++){
        if ($rated_candidates{$system_candidates_ordered[$i]}
           < 1.75){
           $all_correct++;
        }
    }
}
if (-e "precision"){
    `rm precision`;
}
if (-e "recall"){
    `rm recall`;
}
if (-e "f-score"){
```

```
    `rm f-score`;
}
open(PRECISION,">>precision");
open(RECALL,">>recall");
open(F,">>f-score");
my $correct = 0;
if ($ARGV[4] eq "y"){
    for (my $i=0;$i<scalar(@system_candidates_ordered);$i++){
        if ($rated_candidates{$system_candidates_ordered[$i]}
           == 4){
           $correct++;
        }
        my $precision = $correct/($i+1);
        my $recall = $correct/$all_correct;
        my $F = 0;
        if (($precision + $recall) != 0){
            $F = 2*$precision*$recall/($precision+$recall);
        }
        printf PRECISION ("%d\t%.2f\n", $i+1,$precision);
        printf RECALL ("%d\t%.2f\n", $i+1,$recall);
        printf F ("%d\t%.2f\n", $i+1,$F);
    }
}
else {
    for (my $i=0;$i<scalar(@system_candidates_ordered);$i++){
        if ($rated_candidates{$system_candidates_ordered[$i]}
           < 1.75){
           $correct++;
        }
        my $precision = $correct/($i+1);
        my $recall = $correct/$all_correct;
        my $F = 0;
        if (($precision + $recall) != 0){
            $F = 2*$precision*$recall/($precision+$recall);
        }
        printf PRECISION ("%d\t%.2f\n", $i+1,$precision);
        printf RECALL ("%d\t%.2f\n", $i+1,$recall);
        printf F ("%d\t%.2f\n", $i+1,$F);
    }
}
close(PRECISION);
close(RECALL);
close(F);
```

8.20 compute_PRF_head_grouping.pl

```
#!/usr/bin/perl -w
use strict;
use DBI qw(:sql_types);
scalar(@ARGV) == 5 or die "usage: <term> <relation>
<algo:frq|kpr|fkpr|fkpr_bnc|kpr_bnc|pkpr_bnc>
<4_as_1:y|n> <only_drugs:y|n>";
if ($ARGV[1] eq "isa"){
    $ARGV[1] .= "%";
}
#Connect to the database
my $host = 'localhost';
my $db = 'www2rel';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
###LOAD ALL CANDIDATES INTO HASH AND ADD AVE. JUDGMENT
my %rated_nps = ();
my $sth2 = $dbh->prepare("SELECT np,judgment FROM
candidates WHERE term LIKE ? AND relation LIKE ? AND
judgment IS NOT NULL");
my $missing = 0;
$sth2->execute($ARGV[0],$ARGV[1]);
while(my @candidate = $sth2->fetchrow_array){
    my @ratings = split(/,/,$candidate[1]);
    ###INSERT "DON'T KNOW" IF JUDGMENT IS MISSING
    while(scalar(@ratings) < 4){
        push(@ratings,"2");
        $missing++;
    }
    ###STORE BY AVERAGE RATING!
    my $sum_rating = 0;
    foreach (@ratings){
        if ($ARGV[3] eq "y" && $_ =~ /4/){
            $sum_rating++;
        }
        else {
            $sum_rating += $_;
        }
    }
    $rated_nps{$candidate[0]} = sprintf("%.2f",
    ($sum_rating/4));
}
$sth2->finish();
###COMPUTE HEAD FRQ AND KPR
my %head_kpr = ();
my %head_frq = ();
```

```
my %head_fkpr = ();
my $sth1 = $dbh->prepare("SELECT head, kp_id FROM
candidates,np2kp WHERE term LIKE ? AND relation LIKE ?
AND freq > 0 AND id=np2kp.np_id AND judgment IS NOT NULL");
$sth1->execute($ARGV[0],$ARGV[1]);
while(my @head = $sth1->fetchrow_array()){
    $head_kpr{$head[0]} .= $head[1] . ",";
}
$sth1->finish;
while (my($head,$kps) = each %head_kpr){
    my @list = split(/,/,$head_kpr{$head});
    my %diff = map { $_ => 1 } @list;
    my $kpr = keys %diff;
    $head_kpr{$head} = $kpr;
}
my $sth1a = $dbh->prepare("SELECT head,sample_frq FROM
candidates WHERE term LIKE ? AND relation LIKE ? AND
judgment IS NOT NULL GROUP BY np");
$sth1a->execute($ARGV[0],$ARGV[1]);
while(my @head = $sth1a->fetchrow_array()){
    $head_frq{$head[0]} += $head[1];
}
$sth1a->finish;
while (my($head,$frq) = each %head_frq){
    $head_fkpr{$head} = ($frq*$head_kpr{$head});
}
my @system_candidates_ordered = ();
my %system_candidates = ();
if ($ARGV[2] eq "frq"){
    foreach (sort { $head_frq{$b} <=> $head_frq{$a} }
       keys %head_frq){
       push(@system_candidates_ordered,$_);
    }
}
elsif ($ARGV[2] eq "kpr"){
    foreach (sort { $head_kpr{$b} <=> $head_kpr{$a} }
       keys %head_kpr){
       push(@system_candidates_ordered,$_);
    }
}
elsif ($ARGV[2] eq "fkpr"){
    foreach (sort { $head_fkpr{$b} <=> $head_fkpr{$a} }
       keys %head_fkpr){
       push(@system_candidates_ordered,$_);
    }
}
```

```
elsif ($ARGV[2] =~ /.*_bnc/i){
    my %BNC = ();
    open(BNC,"<frq_BNC");
    while (<BNC>){
        chomp;
        my @line = split(/[ ]+/);
        $BNC{$line[0]} = $line[1];
    }
    close(BNC);
    if ($ARGV[2] eq "frq_bnc"){
        foreach (keys %head_frq){
            if (exists($BNC{$_})){
                $system_candidates{$_} = $head_frq{$_}/
                log($BNC{$_});
            }
            else {
                $system_candidates{$_} = $head_frq{$_};
            }
        }
    }
    elsif ($ARGV[2] eq "kpr_bnc"){
        foreach (keys %head_kpr){
            if (exists($BNC{$_})){
                $system_candidates{$_} = $head_kpr{$_}/
                log($BNC{$_});
            }
            else {
                $system_candidates{$_} = $head_kpr{$_};
            }
        }
    }
    elsif ($ARGV[2] eq "fkpr_bnc"){
        foreach (keys %head_fkpr){
            if (exists($BNC{$_})){
                $system_candidates{$_} = $head_fkpr{$_}/
                log($BNC{$_});
            }
            else {
                $system_candidates{$_} = $head_fkpr{$_};
            }
        }
    }
    foreach (sort { $system_candidates{$b} <=>
    $system_candidates{$a} } keys %system_candidates){
       push(@system_candidates_ordered,$_);
    }
```

```
}
###COMPUTE MAX RECALL
my $all_correct = 0;
if ($ARGV[4] eq "y"){
    while (my($np,$judg) = each %rated_nps){
        if ($judg == 4){
            $all_correct++;
        }
    }
}
else {
    while (my($np,$judg) = each %rated_nps){
        if ($judg < 1.75){
            $all_correct++;
        }
    }
}
if (-e "precision"){
    `rm precision`;
}
if (-e "recall"){
    `rm recall`;
}
if (-e "f-score"){
    `rm f-score`;
}
open(PRECISION,">>precision");
open(RECALL,">>recall");
open(F,">>f-score");
my $sth3 = "";
my($current,$correct) = (0,0);
for (my $i=0;$i<scalar(@system_candidates_ordered);$i++){
    ###GET ALL NPS FOR THIS HEAD + STATISTICS
    if ($ARGV[2] =~ /^frq.*/i){
        $sth3 = $dbh->prepare("SELECT np,sum(sample_frq)
        FROM candidates WHERE term LIKE ? AND head LIKE ?
        AND judgment IS NOT NULL GROUP BY np ORDER BY
        sum(sample_frq) DESC");
    }
    elsif ($ARGV[2] =~ /^kpr.*/i){
        $sth3 = $dbh->prepare("SELECT np,sum(kp_range)
        FROM candidates WHERE term LIKE ? AND head LIKE ?
        AND judgment IS NOT NULL GROUP BY np ORDER BY
        sum(kp_range) DESC");
    }
    elsif ($ARGV[2] =~ /^fkpr.*/i){
```

```
            $sth3 = $dbh->prepare("SELECT np,sum(sample_frq)*
            sum(kp_range) AS score FROM candidates WHERE term
            LIKE ? AND head LIKE ? AND judgment IS NOT NULL
            GROUP BY np ORDER BY score DESC");
        }
        $sth3->execute($ARGV[0],$system_candidates_ordered[$i]);
        while (my @nps = $sth3->fetchrow_array){
            if ($ARGV[4] eq "y" && $rated_nps{$nps[0]} == 4){
                $correct++;
            }
            elsif ($ARGV[4] eq "n" && $rated_nps{$nps[0]} < 1.75){
                $correct++;
            }
            $current++;
            my $precision = $correct/$current;
            my $recall = $correct/$all_correct;
            my $F = 0;
            if (($precision + $recall) != 0){
                $F = 2*$precision*$recall/($precision+$recall);
            }
            printf PRECISION ("%d\t%.2f\n", $current,$precision);
            printf RECALL ("%d\t%.2f\n", $current,$recall);
            printf F ("%d\t%.2f\n", $current,$F);
            if ($current < 50){
                print $system_candidates_ordered[$i], "\t",
                $nps[0], "\t", $nps[1], "\n";
            }
        }
    }
    $sth3->finish();
    $dbh->disconnect();
```

8.21 load_www_freqs_for_pmi_into_mysql.pl

```
#!/usr/bin/perl -w
use strict;
use DBI;
use WWW::Mechanize;
scalar(@ARGV) == 4 or die "<term> <relation> <pos:l|r|b>
<frq_gt_x>";
#RECORD THE DOWNLOAD DATE
my $dl_date = `date +%Y-%m-%d`;
#Connect to the database
my $host = 'localhost';
my $db = 'www2rel';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
```

```
###GET KPS FOR TARGET RELATION AND THEIR GOOGLE FREQS
my $sth0 = $dbh->prepare("SELECT kp,id FROM patterns
where relation like ?");
###GET ANNOTATED INSTANCES FOR TARGET RELATION
my $sth1 = $dbh->prepare("SELECT id,np FROM candidates
WHERE relation LIKE ? AND term LIKE ? AND sample_frq > ?
AND judgment IS NOT NULL");
###INSERT WWW-ONLY INSTANCES
my $sth2a = $dbh->prepare("INSERT INTO np2kp
(np_id,kp_id,freq,position,observed,
expected,dl_date,only_www) VALUES(?,?,?,?,?,?,?,?)");
###UPDATE EXISTING INSTANCES WITH WWW STATS
my $sth2b = $dbh->prepare("UPDATE np2kp SET observed=?,
expected=?,dl_date=?,only_www=? WHERE np_id=? AND kp_id=?
AND position like ?");
my $sth3 = $dbh->prepare("SELECT position FROM np2kp WHERE
np_id=? AND kp_id=?");
my $mech = WWW::Mechanize->new();
###MASCARADE AS BG
$mech->agent_alias('Windows IE 6');
###################MAIN#######################
my %kps = ();
###GET KPS ONLY ONCE!###
my $kp_number = $sth0->execute($ARGV[1]);
while (my @kp = $sth0->fetchrow_array){
   ###$kp[0]: kp, $kp[1]: id
   $kps{$kp[0]} = $kp[1];
}
$sth0->finish();
my $filename = $ARGV[0] . "_log_file.txt";
open(LOG,">>$filename");
$sth1->execute($ARGV[1],$ARGV[0],$ARGV[3]);
my $instance_number = $sth1->rows();
if ($ARGV[2] eq "l"){
   &get_google_stats("l");
}
elsif ($ARGV[2] eq "r"){
   &get_google_stats("r");
}
elsif ($ARGV[2] eq "b"){
   &get_google_stats("l");
   &get_google_stats("r");
}
$sth1->finish();
$sth2a->finish();
$sth2b->finish();
```

```
$sth3->finish();
$dbh->disconnect();
close(LOG);
###SUBROUTINES###
sub get_google_stats {
   my $expected_query = "";
   my $observed_query = "";
   my $i = 1;
   while(my @instance = $sth1->fetchrow_array){
      ###$instance[0]: id, $instance[1]: np
      print STDERR $i, " of ", $instance_number, " ",
      $_[0], "\n";
      ###$_[0]: l|r
      if ($_[0] =~ /l/i){
         $expected_query = "\"" . $ARGV[0] . " * " .
         $instance[1] . "\"";
      }
      elsif ($_[0] =~ /r/i){
         $expected_query = "\"" . $instance[1] . " * "
         . $ARGV[0] . "\"";
      }
      my $url = "http://www.google.com/search?q=allintext:
      $expected_query";
      my $expected = 0;
      $mech->get($url);
      my $text = $mech->content();
      if ($text =~ /of about <b>([0-9,]+)<\/b> for/sig){
         $expected = $1;
         $expected =~ tr/,//d;
      }
      foreach my $kp (keys %kps){
         if ($_[0] =~ /l/i){
         $observed_query = "\"" . $ARGV[0] . " " . $kp .
         " " . $instance[1] . "\"";
         }
         elsif ($_[0] =~ /r/i){
            $observed_query = "\"" . $instance[1] . " "
            . $kp . " " . $ARGV[0] . "\"";
         }
         $url = "http://www.google.com/search?q=allintext:
         $observed_query";
         my $observed = 0;
         $mech->get($url);
         $text = $mech->content();
         if ($text =~ /of about <b>([0-9,]+)<\/b> for/sig){
            $observed = $1;
```

```
            $observed =~ tr/,//d;
         }
         #CHECK IF KP AND POSITION HAS BEEN USED OR NOT
         $sth3->execute($instance[0],$kps{$kp});
         my $rows = $sth3->rows();
         my @pos = $sth3->fetchrow_array();
         ###STORE FREQS IN DATABASE!
         if ($rows == 0){
            ###INSERT FREQS
            $sth2a->execute($instance[0],$kps{$kp},0,
            $_[0],$observed,$expected,$dl_date,1);
            print STDERR "Inserting, 0 rows\n";
         }
         elsif ($rows == 1 && $pos[0] !~ /$_[0]/i){
            ###INSERT FREQS
            $sth2a->execute($instance[0],$kps{$kp},1,
            $_[0],$observed,$expected,$dl_date,1);
            print STDERR "Inserting, 1 row\n";
         }
         else{
            ###UPDATE FREQS
            $sth2b->execute($observed,$expected,
            $dl_date,0,$instance[0],$kps{$kp},$_[0]);
            print STDERR "Updating, ", $rows, " rows\n";
         }
         ###PRINT DETAILS TO USER!
         print STDERR $observed, "\t", $observed_query,
         "\n";
         print STDERR $expected, "\t", $expected_query,
         "\n";
         print LOG $observed_query, ",", $observed, ",",
         $expected, ",", $kps{$kp}, "\n";
      }
      $i++;
   }
}
```

8.22 compute_sample_pmi_PRF.pl

```
#!/usr/bin/perl -w
use strict;
use DBI qw(:sql_types);
scalar(@ARGV) == 6 or die "usage: <term> <relation>
<position:l|r|b> <algo:pmi|pmi2> <count_4_as_1:y|n>
<only_drugs:y|n>";
#Connect to the database
```

```
my $host = 'localhost';
my $db = 'www2rel';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
###LOAD ALL CANDIDATES INTO HASH AND ADD AVE. JUDGMENT
my %rated_candidates = ();
my $sth1 = $dbh->prepare("SELECT id,judgment FROM
candidates WHERE term like ? AND relation like ? AND
judgment IS NOT NULL GROUP BY np");
my $missing = 0;
$sth1->execute($ARGV[0],$ARGV[1]);
while(my @candidate = $sth1->fetchrow_array){
   my @ratings = split(/,/,$candidate[1]);
   ###INSERT "DON'T KNOW" IF JUDGMENT IS MISSING!
   while(scalar(@ratings) < 4){
      push(@ratings,"2");
      $missing++;
   }
   ###STORE BY AVERAGE RATING!
   my $sum_rating = 0;
   foreach (@ratings){
      if ($ARGV[4] eq "y" && $_ =~ /4/){
         $sum_rating += 1;
      }
      else {
         $sum_rating += $_;
      }
   }
   $rated_candidates{$candidate[0]} = sprintf("%.2f",
   ($sum_rating/4));
}
$sth1->finish();
###GET KPs FOR RELATION!
my $sth2 = $dbh->prepare("SELECT id FROM patterns WHERE
relation LIKE ?");
my %kps = ();
$sth2->execute($ARGV[1]);
while (my @kp = $sth2->fetchrow_array){
   $kps{$kp[0]} = 0;
}
$sth2->finish();
###GET KP EXPECTED FREQS
my $sth3 = $dbh->prepare("SELECT sum(freq) FROM np2kp
WHERE kp_id = ? AND only_www = 0");
foreach (keys %kps){
   $sth3->execute($_);
   my @kp_freq = $sth3->fetchrow_array();
```

```
   $kps{$_} = $kp_freq[0];
}
$sth3->finish();
###GET OBSERVED FREQS AND NP EXPECTED FREQS
my $sth4 = "";
if ($ARGV[2] eq "b"){
   $sth4 = $dbh->prepare("SELECT freq,sample_frq FROM
   candidates,np2kp WHERE np_id=? AND id=np_id AND kp_id=?
   AND only_www like '0'");
}
else {
   $sth4 = $dbh->prepare("SELECT freq,sample_frq FROM
   candidates,np2kp WHERE np_id=? AND id=np_id AND
   kp_id=? AND position like ? AND only_www like '0'");
}
my %candidate_pmi = ();
my @pmis = ();
###LOOP THROUGH CANDIDATES AND COMPUTE PMI!
while (my($np_id,$rating) = each %rated_candidates){
   while (my($kp_id,$kp_frq) = each %kps){
      if ($ARGV[2] eq "b"){
         $sth4->execute($np_id,$kp_id);
      }
      else {
         $sth4->execute($np_id,$kp_id,$ARGV[2]);
      }
      while (my @stats = $sth4->fetchrow_array){
         if ($sth4->rows() == 0 || $kp_frq == 0){
            next;
         }
         #$stats[0]: observed, $stats[1]: expected1
         my $instance_pmi = 0;
         if ($ARGV[3] eq "pmi"){
            $instance_pmi = log($stats[0]/($stats[1]*
            $kp_frq));
         }
         elsif ($ARGV[3] eq "pmi2"){
            $instance_pmi = log(($stats[0]*$stats[0])/
            ($stats[1]*$kp_frq));
         }
         print STDERR $instance_pmi, "\t", $np_id,
         "\t", $kp_id, "\n";
         $candidate_pmi{$np_id} += $instance_pmi;
         push(@pmis,$instance_pmi);
      }
   }
```

```
}
$sth4->finish();
my @pmi_max = sort {$b <=> $a} @pmis;
while ($pmi_max[0] == 0){
   shift(@pmi_max);
}
print STDERR $pmi_max[0], " max pmi!\n\n";
my %temp = ();
my $sth5 = $dbh->prepare("SELECT kp_range FROM
candidates WHERE id=?");
###ORDER CANDIDATES BY PMI
while (my($np_id,$pmi_sum) = each %candidate_pmi){
   $sth5->execute($np_id);
   my @kpr = $sth5->fetchrow_array();
   my $pmi_score = ($pmi_sum/$pmi_max[0])/$kpr[0];
   $temp{$np_id} = $pmi_score;
}
my @system_candidates_ordered = sort { $temp{$b} <=>
$temp{$a} } keys %temp;
$sth5->finish();
$dbh->disconnect();
###COMPUTE MAX RECALL
my $all_correct = 0;
foreach (keys %rated_candidates){
   if ($ARGV[5] eq "y"){
      if ($rated_candidates{$_} == 4){
         $all_correct++;
      }
   }
else {
   if ($rated_candidates{$_} < 1.75){
      $all_correct++;
      }
   }
}
print $all_correct, " correct candidates\n";
if (-e "precision"){
   `rm precision`;
}
if (-e "recall"){
   `rm recall`;
}
if (-e "f-score"){
   `rm f-score`;
}
open(PRECISION,">>precision");
```

```
open(RECALL,">>recall");
open(F,">>f-score");
my $correct = 0;
if ($ARGV[5] eq "y"){
   for (my $i=0;$i<scalar(@system_candidates_ordered);
      $i++){
      if ($rated_candidates{
         $system_candidates_ordered[$i]} == 4){
         $correct++;
      }
      my $precision = $correct/($i+1);
      my $recall = $correct/$all_correct;
      my $F = 0;
      if (($precision + $recall) != 0){
         $F = 2*$precision*$recall/($precision+$recall);
      }
      printf PRECISION ("%d\t%.2f\n", $i+1,$precision);
      printf RECALL ("%d\t%.2f\n", $i+1,$recall);
      printf F ("%d\t%.2f\n", $i+1,$F);
   }
}
elsif($ARGV[5] eq "n"){
   for (my $i=0;$i<scalar(@system_candidates_ordered);
      $i++){
      if ($rated_candidates{$system_candidates_ordered[$i]}
         < 1.75){
         $correct++;
      }
      my $precision = $correct/($i+1);
      my $recall = $correct/$all_correct;
      my $F = 0;
      if (($precision + $recall) != 0){
         $F = 2*$precision*$recall/($precision+$recall);
      }
      printf PRECISION ("%d\t%.2f\n", $i+1,$precision);
      printf RECALL ("%d\t%.2f\n", $i+1,$recall);
      printf F ("%d\t%.2f\n", $i+1,$F);
   }
}
close(PRECISION);
close(RECALL);
close(F);
```

8.23 compute_pmi_PRF.pl

```
#!/usr/bin/perl -w
```

```
use strict;
use DBI qw(:sql_types);
scalar(@ARGV) == 6 or die "usage: <term> <relation>
<algo:pmi|pmi2|pmi3> <frq_gt_X> <count_4_as_1:y|n>
<only_drugs:y|n>";
#Connect to the database
my $host = 'localhost';
my $db = 'www2rel';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
###LOAD CANDIDATES INTO HASH, ADD AVE. JUDGMENT
my %rated_candidates = ();
my $sth1 = $dbh->prepare("SELECT np,judgment FROM
candidates WHERE term LIKE ? AND relation LIKE ?
AND sample_frq > ? AND judgment IS NOT NULL");
my $missing = 0;
$sth1->execute($ARGV[0],$ARGV[1],$ARGV[3]);
while(my @candidate = $sth1->fetchrow_array){
   my @ratings = split(/,/,$candidate[1]);
   ###INSERT "DON'T KNOW" IF JUDGMENT IS MISSING
   while(scalar(@ratings) < 4){
      push(@ratings,"2");
      $missing++;
   }
   ###STORE BY AVERAGE RATING!
   my $sum_rating = 0;
   foreach (@ratings){
      if ($ARGV[4] eq "y" && $_ =~ /4/){
         $sum_rating += 1;
      }
      else {
         $sum_rating += $_;
      }
   }
   $rated_candidates{$candidate[0]} = sprintf("%.2f",
   ($sum_rating/4));
}
$sth1->finish();
###LOAD ALL CANDIDATES TO COMPUTE PMI
my $sth2 = $dbh->prepare("SELECT np,observed,expected,
patterns.google_freq,kp_id,position FROM candidates,
np2kp,patterns WHERE candidates.id=np2kp.np_id AND
np2kp.kp_id=patterns.id AND term LIKE ? AND candidates.
relation LIKE ? AND sample_frq > ? AND judgment IS
NOT NULL");
$sth2->execute($ARGV[0],$ARGV[1],$ARGV[3]);
my %candidate_pmi = ();
```

```
my %kp_count = ();
my @pmis = ();
while(my @cand = $sth2->fetchrow_array){
   if ($cand[1] > 0 && $cand[2] > 0){
      my $instance_pmi = 0;
      if ($ARGV[2] eq "pmi"){
         $instance_pmi = log($cand[1]/($cand[2]*$cand[3]));
      }
      elsif ($ARGV[2] eq "pmi2"){
         $instance_pmi = log(($cand[1]*$cand[1])/
         ($cand[2]*$cand[3]));
      }
      elsif ($ARGV[2] eq "pmi3"){
         $instance_pmi = log(($cand[1]*$cand[1]*$cand[1])/
         ($cand[2]*$cand[3]));
      }
      $candidate_pmi{$cand[0]} += $instance_pmi;
      my $kp = $cand[4] . "_" . $cand[5];
      $kp_count{$kp}++;
      push(@pmis,$instance_pmi);
   }
}
$sth2->finish();
$dbh->disconnect();
my @pmi_max = sort {$b <=> $a} @pmis;
my $kp_no = scalar(keys %kp_count);
my %temp = ();
###ORDER CANDIDATES BY PMI
while (my($np,$pmi_sum) = each %candidate_pmi){
   my $pmi_score = ($pmi_sum/$pmi_max[0])/$kp_no;
   $temp{$np} = $pmi_score;
}
my @system_candidates_ordered = sort { $temp{$b} <=>
$temp{$a} } keys %temp;
###COMPUTE MAX RECALL
my $all_correct = 0;
foreach (keys %rated_candidates){
   if ($ARGV[5] eq "y"){
      if ($rated_candidates{$_} == 4){
         $all_correct++;
      }
   }
   else {
      if ($rated_candidates{$_} < 1.75){
         $all_correct++;
      }
```

```
   }
}
print $all_correct, " correct candidates\n";
if (-e "precision"){
`rm precision`;
}
if (-e "recall"){
`rm recall`;
}
if (-e "f-score"){
`rm f-score`;
}
open(PRECISION,">>precision");
open(RECALL,">>recall");
open(F,">>f-score");
my $correct = 0;
if ($ARGV[5] eq "y"){
   for (my $i=0;$i<scalar(@system_candidates_ordered);$i++){
      if ($rated_candidates{$system_candidates_ordered[$i]}
      == 4){
         $correct++;
   }
   my $precision = $correct/($i+1);
   my $recall = $correct/$all_correct;
   my $F = 0;
   if (($precision + $recall) != 0){
      $F = 2*$precision*$recall/($precision+$recall);
   }
   printf PRECISION ("%d\t%.2f\n", $i+1,$precision);
   printf RECALL ("%d\t%.2f\n", $i+1,$recall);
   pri ntf F ("%d\t%.2f\n", $i+1,$F);
   }
}
elsif($ARGV[5] eq "n"){
   for (my $i=0;$i<scalar(@system_candidates_ordered);$i++){
      if ($rated_candidates{$system_candidates_ordered[$i]}
         < 1.75){
         $correct++;
   }
   my $precision = $correct/($i+1);
   my $recall = $correct/$all_correct;
   my $F = 0;
   if (($precision + $recall) != 0){
      $F = 2*$precision*$recall/($precision+$recall);
   }
   printf PRECISION ("%d\t%.2f\n", $i+1,$precision);
```

```
   printf RECALL ("%d\t%.2f\n", $i+1,$recall);
   printf F ("%d\t%.2f\n", $i+1,$F);
   }
}
close(PRECISION);
close(RECALL);
close(F);
```

8.24 load_passive_kpr_from_www_into_mysql.pl

```
#!/usr/bin/perl -w
use strict;
use DBI qw(:sql_types);
use WWW::Mechanize;
scalar(@ARGV) == 4 or die "usage: <term> <relation>
<frq_gt_X> <term_pos:l|r>";
#Connect to the database
my $host = 'localhost';
my $db = 'www2rel';
my $dbh = DBI->connect("dbi:mysql:$db:$host");
###LOAD THE PASSIVE KPs
my %pats = ();
my $sth2 = $dbh->prepare("SELECT kp,id FROM patterns
WHERE relation like 'induced_by'");
$sth2->execute();
while(my @kp = $sth2->fetchrow_array){
   print STDERR $kp[0], "\n";
   $pats{$kp[0]} = $kp[1];
}
$sth2->finish();
my $mech = WWW::Mechanize->new();
$mech->agent_alias('Windows IE 6');
###GET THE CANDIDATES
my $sth1 = $dbh->prepare("SELECT np,id FROM candidates
WHERE term like ? AND relation like ? AND sample_frq > ?
AND judgment IS NOT NULL");
###UPDATE DB WITH PASSIVE KPR
my $sth0 = $dbh->prepare("UPDATE candidates SET
kp_range_passive=?,passive_www_frq=? WHERE id=?");
###REMEMBER NP-KP MAPPINGS FOR PRECISION CALCULATIONS
my $sth3 = $dbh->prepare("INSERT INTO np2kp (np_id,kp_id,
freq,position) VALUES (?,?,?,?)");
$sth1->execute($ARGV[0],$ARGV[1],$ARGV[2]);
while(my @np = $sth1->fetchrow_array){
   my $passive_www_kpr = 0;
   my $passive_www_frq = 0;
```

```
   foreach my $kp (keys %pats){
      ###FORM GOOGLE QUERY
      my $query = "";
      if ($ARGV[3] eq "r"){
         $query = "\"" . $np[0] . " " . $kp . " " .
         $ARGV[0] . "\"";
      }
      elsif ($ARGV[3] eq "l"){
         $query = "\"" . $ARGV[0] . " " . $kp . " "
         . $np[0] . "\"";
      }
      print STDERR $query, "\n";
      my $url = "http://www.google.com/search?q=allintext:
      $query";
      $mech->get($url);
      my $text = $mech->content();
      my $hits = 0;
      if ($text =~ /of about <b>([0-9,]+)<\/b> for/sig){
         $hits = $1;
         $hits =~ tr/,//d;
         $passive_www_frq += $hits;
         $passive_www_kpr++;
      }
      $sth3->execute($np[1],$pats{$kp},$hits,$ARGV[3]);
      print STDERR "\t", $hits, "\n";
   }
   $sth0->execute($passive_www_kpr,$passive_www_frq,$np[1]);
   print STDERR "------------\n", $np[0], "\t", $passive_www_kpr,
   "\t", $passive_www_frq, "\n";
}
$sth0->finish();
$sth1->finish();
$sth3->finish();
$dbh->disconnect();
```

8.25 measure_real_PRF_of_all_KPs.pl

```
#!/usr/bin/perl -w
use strict;
use DBI qw(:sql_types);
scalar(@ARGV) == 6 or die "usage: <term> <relation>
<position> <AVE_FOR_REL?:y|n> <threshold: fx. 1.50>
<count_4_as_1:y|n>";
#Connect to the database
my $host = 'localhost';
my $db = 'www2rel';
```

```
my $dbh = DBI->connect("dbi:mysql:$db:$host");
my %all_correct = ("aspirin" => 148,
                   "selenium" => 50,
                   "vomiting" => 59,
                   "emesis" => 25,
                   "formaldehyde" => 4,
                   "vitamin c" => 5,
                   "lactose" => 1,
                   "glucose" => 4,
                   "progesterone" => 2,
                   "antipsychotic" => 88,
                   "haloperidol" => 57,
                   "both_isa" => 145,
                   "isa_hyper_sing_hypo" => 100,
                   "isa_hyper_plur_hypo" => 70,
                   "isa_hypo_hyper_sing" => 71,
                   "isa_hypo_hyper_plur" => 55,
                   "all_synonyms" => 16,
                   "all_induces" => 232,
                   );
my $sth1 = "";
my $sth2 = "";
my $sth3 = "";
my %positives = ();
my %negatives = ();
if ($ARGV[3] eq "y"){
    if ($ARGV[1] eq "isa"){
        $sth1 = $dbh->prepare("SELECT judgment,id
        FROM candidates WHERE relation like 'isa_%'
        AND judgment IS NOT NULL");
        $sth1->execute();
    }
    else {
        $sth1 = $dbh->prepare("SELECT judgment,id FROM
        candidates WHERE relation like ? AND judgment
        IS NOT NULL");
        $sth1->execute($ARGV[1]);
    }
    $sth2 = $dbh->prepare("SELECT kp_id,freq FROM
    np2kp WHERE np_id=? AND only_www LIKE '0'");
}
elsif ($ARGV[3] eq "n"){
    $sth1 = $dbh->prepare("SELECT judgment,id FROM
    candidates WHERE term like ? AND relation like ?
    AND judgment IS NOT NULL");
    if ($ARGV[1] =~ /isa.*/i){
```

```
        $sth2 = $dbh->prepare("SELECT kp_id,freq FROM np2kp
        WHERE np_id=? AND only_www LIKE '0'");
    }
    else {
        $sth2 = $dbh->prepare("SELECT kp_id,freq FROM
        np2kp WHERE np_id=? AND only_www LIKE '0' AND
        position like ?");
    }
    $sth1->execute($ARGV[0],$ARGV[1]);
}
my %recall = ();
my %precision_pos = ();
my %precision_neg = ();
###COMPUTE AVE. JUDGMENTS FOR ALL RELEVANT NPs
while(my @candidate = $sth1->fetchrow_array){
    my @ratings = split(/,/,$candidate[0]);
    my $total = 0;
    foreach (@ratings){
       if ($ARGV[5] eq "y" && $_ =~ /4/){
          $total++;
       }
       else {
          $total += $_;
       }
    }
    my $ave = $total/4;
    ###RECORD WITH WHICH KPS THIS NP OCCURS
    if ($ARGV[3] eq "y" || $ARGV[1] =~ /isa.*/i){
        $sth2->execute($candidate[1]);
    }
    else {
        $sth2->execute($candidate[1],$ARGV[2]);
    }
    while (my @kps = $sth2->fetchrow_array){
        if ($ave <= $ARGV[4]){
            #ASSOCIATE POS. CANDIDATES WITH THEIR KPS
            $recall{$kps[0]} .= $candidate[1] . "\t";
            $precision_pos{$kps[0]} += $kps[1];
        }
        else {
            $precision_neg{$kps[0]} += $kps[1];
        }
    }
}
$sth1->finish();
$sth2->finish();
```

```
if ($ARGV[0] eq "both_isa" && $ARGV[1] eq "isa"
    && $ARGV[3] eq "y"){
    $sth3 = $dbh->prepare("SELECT id,kp FROM patterns
    WHERE relation like 'isa_%'");
    $sth3->execute();
}
else {
    $sth3 = $dbh->prepare("SELECT id,kp FROM patterns
    WHERE relation like ?");
    $sth3->execute($ARGV[1]);
}
###LOOP THROUGH ALL KP FOR CANDIDATES AND COMPUTE P,R,F
while (my @kps = $sth3->fetchrow_array){
    ###FORMAT: kp_id, kp
    if (exists($recall{$kps[0]})){
        $recall{$kps[0]} =~ s/\t$//;
        my @tmp = split(/\t/,$recall{$kps[0]});
        my %correct = map { $_ => 1 } @tmp;
        my $correct_returned = scalar(keys %correct);
        #COMPUTE PREC., RECALL AND F-SCORE FOR THIS KP
        my $rec = $correct_returned/$all_correct{$ARGV[0]};
        my $prec = 0;
        if (!exists($precision_neg{$kps[0]})){
            $prec = 1;
        }
        else {
            $prec = $precision_pos{$kps[0]}/($precision_pos
            {$kps[0]}+$precision_neg{$kps[0]});
        }
        my $F = 2*$prec*$rec/($prec+$rec);
        printf("%.2f\t%.2f\t%.2f\t%s_%d\n", $F, $prec,
        $rec, $kps[1], $kps[0]);
    }
    else {
        if (!exists($precision_neg{$kps[0]})){
            print "NA\tNA\tNA\t", $kps[1], "_",
            $kps[0], "\n";
        }
        else {
            print "0\t0\t0\t", $kps[1], "_",
            $kps[0], "\n";
        }
    }
}
$sth3->finish();
$dbh->disconnect();
```

8.26 UMLS2term_pairs.pl

```
#!/usr/bin/perl
# Last edited by Jakob Halskov
# on June 16 2006
#
use DBI;
use strict;
scalar(@ARGV) == 4 or die "Usage: <REL: fx. isa>
<CONCEPT: fx. C0040615> <up|down> <active_ingredient:y|n>";
#Connect to the database
my $host = 'localhost';
my $db = 'UMLS';
my $user = 'jakob';
my $pass = 'halskov';
my $dbh = DBI->connect("dbi:mysql:$db:$host", "$user", "$pass");
###FETCH THE CUIS FOR THE GIVEN RELATION
my $sth1 = "";
if ($ARGV[2] eq "up"){
   $sth1 = $dbh->prepare("SELECT CUI1 FROM MRREL WHERE
   RELA LIKE ? AND CUI2 LIKE ? GROUP BY CUI1");
}
elsif ($ARGV[2] eq "down"){
   $sth1 = $dbh->prepare("SELECT CUI2 FROM MRREL WHERE
   RELA LIKE ? AND CUI1 LIKE ? GROUP BY CUI2");
}
###
##########FETCH THE TRADE NAMES OF CONCEPT IF ANY
my $sth2 = $dbh->prepare("SELECT CUI1 FROM MRREL WHERE
RELA LIKE 'has_tradename' AND CUI2 LIKE ? GROUP BY CUI1");
##########FETCH THE ACTIVE INGREDIENTS IF NEEDED
my $sth2b = $dbh->prepare("SELECT CUI1 FROM MRREL WHERE
RELA LIKE 'has_ingredient' AND CUI2 LIKE ? GROUP BY CUI1");
##########FETCH THE TERM VARIANTS FOR EACH CUI PAIR
my $sth3 = $dbh->prepare("SELECT STR FROM MRCONSO WHERE
CUI LIKE ?");
$sth1->execute($ARGV[0],$ARGV[1]);
while (my @cui = $sth1->fetchrow_array){
    $sth3->execute($cui[0]);
    while (my @vars = $sth3->fetchrow_array){
        print $cui[0], "_", lc($vars[0]), "\n";
    }
    ###PRINT ALSO ACTIVE INGREDIENTS?
    if ($ARGV[3] eq "y"){
        $sth2b->execute($cui[0]);
        if ($sth2b->rows() > 0){
```

```
            while (my @ingr = $sth2b->fetchrow_array){
                $sth3->execute($ingr[0]);
                while (my @names = $sth3->fetchrow_array){
                    print $cui[0], "_", lc($names[0]), "\n";
                }
            }
        }
    }
    ###ANY TRADE NAMES??
    $sth2->execute($cui[0]);
    if ($sth2->rows() > 0){
        while (my @names = $sth2->fetchrow_array){
            $sth3->execute($names[0]);
            while (my @vars = $sth3->fetchrow_array){
                print $cui[0], "_", lc($vars[0]), "\n";
            }
        }
    }
}
print STDERR $sth1->rows(), " concepts!\n";
$sth1->finish();
$sth2->finish();
$sth2b->finish();
$sth3->finish();
$dbh->disconnect();
```

8.27 Induces training pairs

term1-term2
candida_albicans-allergy
alcohol-unconsciousness
plague-headache
alcohol-vomiting
sodium-unconsciousness
phenylephrine-mydriasis
alcohol-flushing
alcohol-diarrhea
co2-shaking
sodium-allergy
co2-respiratory_acidosis
niacinamide-flushing
ipecac_syrup-vomiting
lactose-diarrhea
carbon_dioxide-death
chorionic_gonadotropin-pregnancy
aspergillus-allergy
vitamin_c-diarrhea
calcium-diarrhea
thiopental-unconsciousness
sodium-vomiting
oxygen-retinopathy_of_prematurity
alcohol-amnesia
calcium-hyperoxaluria
co2-death
nicotinic_acid-flushing
methylene_blue-methemoglobinemia
atropine-mydriasis
niacin-flushing
ipecac-vomiting
niaspan-flushing
propofol-unconsciousness
glucose-allergy
carbon_dioxide-suffocation
candida-allergy
carbon_dioxide-respiratory_acidosis
nitrous_oxide-nausea
aspirin-diarrhea
syrup_of_ipecac-vomiting
scopolamine-amnesia

8.28 May_prevent training pairs

term1-term2
calcium-magnesium_deficiency
prilocaine-pain
metoclopramide-nausea
psyllium-constipation
calcium-plaque
pegfilgrastim-neutropenia
flunisolide-asthma
propofol-pain
ondansetron-vomiting
bupivacaine-pain
amantadine-flu
oseltamivir-flu
ether-pain
t4-hypothyroidism
mineral_oil-constipation
melatonin-jet_lag
rabies_vaccines-rabies
enoxaparin-thromboembolism
fluoride-plaque
thimerosal-flu
ketamine-pain
estrogen-postmenopausal_osteoporosis
insulin-hyperglycemia
mefloquine-malaria
cimetidine-duodenal_ulcer
zanamivir-flu
fluoride-tooth_decay
ganciclovir-cytomegalovirus_infection
disulfiram-alcoholism
levonorgestrel-pregnancy
granisetron-nausea
warfarin-stroke
echinacea-cold
botulinum_toxin-migraine
heparin-pulmonary_embolism
calcium-osteoporosis
ibuprofen-pain
beclomethasone-asthma
adrenalin-pain
dexamethasone-vomiting

8.29 Synonymy training pairs

term1 <=> term2
spontaneous abortion <=> miscarriage
diarrhea <=> loose stools
dyspnea <=> breathlessness
dyspnea <=> difficulty breathing
fever <=> increased body temperature
fever <=> hyperthermia
fever <=> pyrexia
fever <=> temperature increase
immediate hypersensitivity <=> allergy
hypotension <=> decreased blood pressure
hypotension <=> low blood pressure
kidney failure <=> renal failure
miosis <=> pupillary constriction
mydriasis <=> dilation of the pupil
pregnancy <=> gestation
retinopathy of prematurity <=> retrolental fibroplasia
RLF <=> retrolental fibroplasia
vomiting <=> emesis
myocardial infarction <=> heart attack
pruritus <=> itching

8.30 ISA training pairs

8.30.1 Plural hypernym - hyponym

term pair
analgesics;gabapentin
analgesics;ketamine
analgesics;nonopioid
antipyretics;aspirin
antipyretics;paracetamol
anesthetics;cyclopropane
anticonvulsant drugs;carbamazepine
anticonvulsant drugs;phenytoin
anticonvulsants;carbamazepine
anticonvulsants;diphenylhydantoin
anticonvulsants;divalproex
anticonvulsants;ethosuximide
anticonvulsants;etiracetam
anticonvulsants;gabapentin

anticonvulsants;lamotrigine
anticonvulsants;levetiracetam
anticonvulsants;primidone
anticonvulsants;topiramate
anticonvulsants;valproic acid
anticonvulsants;zonisamide
antidepressants;lamotrigine
antidepressants;moclobemide
antidepressants;sibutramine
antidepressants;tricyclic antidepressant
antidepressive agents;sertraline
anticonvulsants;clonazepam
antiparkinson agents;levodopa
antiparkinson agents;apomorphine
antiparkinsonian agents;levodopa
antidepressants;phenelzine
gastrointestinal agents;cisapride
gastrointestinal agents;infliximab
gastrointestinal agents;octreotide
gastrointestinal agents;antacids
gastrointestinal drugs;antacids
gastrointestinal agents;lactulose
cardiovascular agents;diuretics
cardiovascular drugs;diuretics
cardiovascular agents;calcium channel blockers
cardiovascular agents;diltiazem

8.30.2 Singular hypernym - hyponym

term pair
analgesic;adenosine
analgesic;dalargin
analgesic;flurbiprofen
analgesic;gabapentin
analgesic;ketamine
analgesic;nonopioid
anticonvulsant;diazepam
anticonvulsant;ethosuximide

anticonvulsant;felbamate
anticonvulsant;gabapentin
anticonvulsant;lorazepam
anticonvulsant;mephenytoin
anticonvulsant;melatonin
anticonvulsant;phenobarbital
anticonvulsant;propofol
anticonvulsant;topiramate
anticonvulsant;valproate
anticonvulsant;vigabatrin
antidepressant;duloxetine
antidepressant;lamotrigine
antidepressant;moclobemide
antidepressant;rolipram
antidepressant;sertraline
antidepressant;tricyclic antidepressant
antipyretic;paracetamol
analgesic-clonidine
antidepressant-phenelzine
antidepressant-reboxetine
anticonvulsant-phenobarbitone
anesthetic-cyclopropane
vitamin-tocopherol
vitamin-cholecalciferol
vitamin-retinol
vitamin-ascorbic acid
vitamin-phytonadione
vitamin-menaquinone
vitamin-calcitriol
vitamin-meadione
vitamin-bioflavonoid
vitamin-calcifediol

8.30.3 Hyponym - plural hypernym

term pair
clonazepam;anticonvulsants

gabapentin;anticonvulsants
lamotrigine;anticonvulsants
moclobemide;antidepressants
phenobarbital;anticonvulsants
phenytoin;anticonvulsants
reboxetine;antidepressants
tranylcypromine;antidepressants
valproic acid;anticonvulsants
lamotrigine;antidepressants
sertraline;antidepressants
duloxetine;antidepressants
phenelzine;antidepressants
milnacipran;antidepressants
valproate;anticonvulsants
carbamazepine;anticonvulsants
opioid analgesics;analgesics
nonopioid analgesics;analgesics
ketamine;analgesics
paracetamol;antipyretics
cyclopropane;anesthetics
lithium carbonate;antidepressants
tricyclic antidepressant;antidepressants
aspirin;antipyretics
gabapentin;analgesics
calcium channel blockers;cardiovascular agents
bioflavonoids;vitamins
phenytoin;anticonvulsant drugs
topiramate;anticonvulsants
sibutramine;antidepressants
clonidine;analgesics
vitamin e;vitamins
c vitamin;vitamins
a vitamin;vitamins
tocopherol;vitamins
cholecalciferol;vitamins
d vitamin;vitamins
vitamin k 3;vitamins

ascorbic acid;vitamins
vitamin d;vitamins
vitamin e;vitamins

8.30.4 Hyponym - singular hypernym

term pair
acetaminophen;antipyretic
carbamazepine;anticonvulsant
clonazepam;anticonvulsant
clonidine;analgesic
d 23129;anticonvulsant
diazepam;anticonvulsant
divalproex;anticonvulsant
felbamate;anticonvulsant
gabapentin;analgesic
ibuprofen;antipyretic
ketamine;analgesic
lamotrigine;analgesic
milnacipran;antidepressant
moclobemide;antidepressant
nonopioid;analgesic
oxcarbazepine;anticonvulsant
paracetamol;antipyretic
phenytoin;anticonvulsant
pregabalin;anticonvulsant
sertraline;antidepressant
sibutramine;antidepressant
topiramate;anticonvulsant
trimethadione;anticonvulsant
valproate;anticonvulsant
vigabatrin;anticonvulsant
gabapentin;anticonvulsant
phenobarbital;anticonvulsant
duloxetine;antidepressant
melatonin;anticonvulsant
primidone;anticonvulsant
cisapride;gastrointestinal agent

cisapride;gastrointestinal drug
motilin;gastrointestinal agent
ascorbic acid;vitamin
tocopherol;vitamin
retinol;vitamin
cholecalciferol;vitamin
menadine;vitamin
tocotrienol;vitamin
phytomenadione;vitamin

8.31 Unfiltered "may_prevent" patterns precision scores

Ave. precision	Ave. frequency	KP
100.00	444.1	+to +prevent
100.00	326.3	+for +relieving
100.00	307.9	+for +preventing
100.00	281.1	+helps +prevent
100.00	260.6	+relieves
100.00	185.2	+in +preventing
100.00	133.3	+cuts
100.00	127.8	+prevents
100.00	114.0	+relieve
100.00	59.5	+to +relieve
100.00	57.5	+to +reduce
100.00	37.0	+for +treating
100.00	29.2	+decreases
100.00	28.0	+help +prevent
100.00	23.7	+in +treating
100.00	22.9	+prevent
100.00	22.2	+reduced
100.00	22.1	+to +treat
100.00	20.2	+reduce
100.00	18.4	+to +control
100.00	15.4	+to +combat
100.00	13.1	+prevented
100.00	12.9	+containing
100.00	12.2	+based
100.00	11.2	+decreased
100.00	9.1	+may +ease
100.00	8.1	+improves
100.00	7.3	+can +prevent

Ave. precision	Ave. frequency	KP
100.00	6.8	+will +prevent
100.00	6.3	+to +alleviate
100.00	5.7	+affects
100.00	5.4	+significantly +reduces, +relieved, +can +relieve
100.00	5.3	+helps
100.00	5.1	+combats, +alleviates
100.00	5.0	+protects +against
100.00	4.0	+use +in
100.00	3.9	+used +for
100.00	3.8	+preventing, +at +preventing
100.00	3.7	+diminishes
100.00	3.3	+that +prevents
100.00	3.2	+had +significantly +less
100.00	3.0	+inhibits
100.00	2.7	+eliminates
100.00	2.6	+significantly +reduced, +has
100.00	2.5	+attenuates
100.00	2.3	+treated, +lowers
100.00	2.1	+in +relieving, +in +fighting, +group +had
100.00	2.0	+fighting
100.00	1.9	+provides
100.00	1.8	+provided, +in +decreasing
100.00	1.7	+group +experienced
100.00	1.6	+makes
100.00	1.4	+could +reduce
100.00	1.2	+will +relieve, +minimizes, +may +prevent
100.00	1.0	+would +decrease, +developed
100.00	1.0	+remedy, +was, +group +reported
100.00	0.9	+lessens, +helps +relieve
100.00	0.7	+reverses, +heals
100.00	0.6	+to +kill, +to +cure, +deters, +following
100.00	0.5	+numbs, significantly +decreased, +cut
100.00	0.5	+does +not +cure
100.00	0.4	+inhibited, to +suppress, +ease, +caused
100.00	0.3	+eases, +can +cure, +also +relieves, +eased
100.00	0.2	+works +against, +therapy +prevents, +provide
100.00	0.2	+may +decrease, +after +developing, +showing
100.00	0.2	+for +decreasing, +can, +alleviated
60.92	373.5	+reduces
53.15	34.9	+can +reduce
26.11	113.8	+may +reduce
22.21	195.0	+causes

Ave. precision	Ave. frequency	KP
17.5	24.9	+deficiency
11.87	4.3	+produces
11.45	131.0	+induced
11.45	22.9	+cause
10.24	153.7	+can +cause
8.31	364.2	+is
6.30	11.1	+does +not +cause
5.38	9.2	+causing
3.50	5.0	+can +also +cause
3.17	27.7	+are
0.97	88.8	+include
0.67	9.3	+were
0.57	137.7	+had
0.55	11.1	+will+cause
0.00	46.8	+contracted
0.00	6.3	+based +on
0.00	4.0	+reported
0.00	1.8	+will +have, +make

8.32 Unfiltered "induces" patterns precision scores

Ave. precision	Ave. frequency	KP
100.00	163.2	+may +cause
100.00	149.8	+to +induce
100.00	85.8	+produce
100.00	33.5	+induces
100.00	20.7	+to +cause
100.00	15.9	+cause
100.00	15.8	+overdose +include
100.00	14.7	+does +not +cause
100.00	11.6	+which +induces
100.00	7.7	+produces
100.00	7.2	+induce
100.00	7.1	+can +lead +to
100.00	6.4	+to +produce
100.00	6.1	+or +induce
100.00	5.6	+which +causes
100.00	5.3	+will +cause
100.00	5.0	+will +induce

100.00	4.7	+causing
100.00	4.5	+does +not +produce
100.00	4.3	+poisoning +include
100.00	3.3	+leading +to
100.00	3.0	+may +lead +to
100.00	2.8	+produced
100.00	2.7	+can +produce
100.00	2.3	+can +induce
100.00	2.1	+can +also +cause
100.00	1.9	+poisoning
100.00	1.6	+promotes, +can +result +in
100.00	1.5	+without +causing, +may +experience
100.00	1.3	+that +can +cause
100.00	1.1	+leads +to, +for +inducing
100.00	1.0	+may +induce
100.00	0.9	+which +promotes, +resulted +in, +and +induce
100.00	0.8	+characterized +by, +can +itself +cause
100.00	0.8	+provokes, +it +causes
100.00	0.6	+may +aggravate, +consumption +include, +retention
100.00	0.5	+caused
100.00	0.4	+commonly +causes, +when +you +have
100.00	0.4	+because +inducing
100.00	0.3	+usually +results +in, +was +used +for
100.00	0.3	+resulting +in, +will +produce, +intake +is
100.00	0.3	+in +producing, +has +been, +associated +with, +aggravates
100.00	0.2	+overdose +can +cause, +tension, +see
100.00	0.2	+maintains, +in +maintaining, +could +cause
100.00	0.2	+and +develop, +if +inducing, +has, +did +not +cause
99.71	1514.0	+induced
87.37	56.5	+causes
78.55	81.6	+can +cause
46.89	104.2	+is
43.08	68.3	+include
32.10	38.4	+associated
31.19	66.8	+related
29.87	11.1	+are
28.06	20.8	+to +prevent
15.08	40.8	+to +reduce
10.97	6.1	+experienced
5.31	282.1	+prevents
0.64	48.7	+to +treat
0.00	982.8	+relieves

0.00	139.0	+had
0.00	96.3	+reduces
0.00	78.3	+in +treating
0.00	41.1	+can +help +minimize
0.00	5.9	+relieved
0.00	5.4	+develop
0.00	2.4	+reduce

8.32.1 "Induced_by" patterns precision scores

Ave. precision	Ave. frequency	KP
100.00	53.6	+induced +by
100.00	23.7	+caused +by
66.51	12.6	+associated +with
100.00	10.6	+produced +by
100.00	4.6	+related +to
100.00	3.4	+was +induced +by
100.00	3.1	+by +administering
100.00	2.2	+resulting +from
100.00	2.2	+was +induced +with
100.00	1.3	+when +taken +with
100.00	0.6	+occurs +with
100.00	0.4	+occurs +when
100.00	0.4	+may +result +from

8.33 Unfiltered "ISA" patterns precision scores

8.33.1 hyper_plur_hypo

iteration range	precision	freq	pattern	accepted
10	99.98	107.9	+e.g	yes
10	99.98	107.9	+eg	yes
10	99.88	1289.4	+such +as	yes
10	99.16	74.8	+including	yes
10	9.77	1717.1	+on	no
10	89.60	56.7	+like	yes
10	69.43	13.4	+i.e	yes
10	69.02	37.4	+include	yes
10	5.91	19.2	+but	no
10	40.52	134.4	+are	no
10	30.28	753.2	+as	no
10	28.04	659.8	+to	no

iteration range	precision	freq	pattern	accepted
10	23.44	32939.3	+or	no
10	23.16	124827.4	+and	no
10	19.63	906.2	+with	no
9	9.33	1170.6	+the	no
9	77.15	12.8	+ie	yes
9	22.17	11.2	+analgesics	no
9	0.00	9.7	+both	no
9	0.00	860.6	+a	no
9	0.00	7.2	+focus +on	no
9	0.00	4.5	+group	no
9	0.00	4.5	+developed	no
9	0.00	31.5	+corticosteroids	no
9	0.00	2.7	+recently	no
9	0.00	2.7	+including +the	no
9	0.00	20.1	+first	no
9	0.00	19.8	+were	no
9	0.00	196.2	+not	no
9	0.00	1.8	+since +this	no
9	0.00	1.8	+and +include	no
9	0.00	153.0	+known +as	no
7	100.00	17.7	+carbamazepine	no
7	100.00	13.6	+phenobarbital	no
6	100.00	9.3	+especially	no
6	100.00	41.8	+phenytoin	no
6	100.00	35.8	+phenytoin +and	no
5	100.00	6.8	+particularly	no
4	100.00	9.9	+other +than	no
4	100.00	8.1	+valproate +and	no
4	100.00	7.1	+carbamazepine +and	no
4	100.00	5.3	+valproate	no
4	100.00	4.3	+except	no
4	100.00	2.6	+mainly	no
4	100.00	16.3	+gabapentin +and	no
4	100.00	12.8	+phenobarbital +and	no
3	100.00	8.9	+e.g. +phenytoin	no
3	100.00	7.3	+lamotrigine	no
3	100.00	4.8	+gabapentin	no
3	100.00	3.1	+valproic +acid	no
3	100.00	2.2	+phenobarbitone	no
3	100.00	0.9	+diazepam	no
2	100.00	4.6	+barbiturates	no
2	100.00	3.7	+valproic +acid +and	no
2	100.00	2.2	+including +phenytoin	no

iteration range	precision	freq	pattern	accepted
2	100.00	1.8	+topiramate +and	no
2	100.00	1.6	+diphenylhydantoin +and	no
2	100.00	1.5	+for +example	no
2	100.00	1.5	+divalproex +sodium	no
2	100.00	1.4	+sodium +valproate +and	no
2	100.00	1.3	+specifically	no
2	100.00	1.1	+in +particular	no
2	100.00	1.0	+decreased +serum	no
2	100.00	0.8	+including +carbamazepine	no
2	100.00	0.5	+only	no
1	100.00	1.4	+although	no
1	100.00	1.0	+lamotrigine +and	no
1	100.00	1.0	+gabapentin +or	no
1	100.00	1.0	+felbamate	no
1	100.00	0.9	+vigabatrin	no
1	100.00	0.6	+benzodiazepines	no
1	100.00	0.4	+divalproex +and	no
1	100.00	0.4	+antibiotics	no
1	100.00	0.3	+triazines	no
1	100.00	0.3	+such +valproate	no
1	100.00	0.3	+oxcarbazepine	no
1	100.00	0.3	+morphine	no
1	100.00	0.3	+ethosuximide	no
1	100.00	0.3	+divalproex	no
1	100.00	0.2	+topiramate	no
1	100.00	0.2	+patients +using	no
1	100.00	0.2	+initially	no
1	100.00	0.2	+fluoxetine +and	no
1	100.00	0.2	+fluoxetine	no
1	100.00	0.2	+clonazepam	no
1	100.00	0.2	+and +benzodiazepines	no
1	100.00	0.2	+along +with	no

8.33.2 hyper_sing_hypo

iteration range	precision	freq	pattern	accepted
10	99.94	338.6	+drug	yes
10	88.59	28.7	+activity	yes
10	79.52	82.3	+such +as	yes
10	7.34	604633.7	+and	no
10	46.35	13675.3	+e	no
10	42.63	5361.6	+d	no
10	5.18	8.2	+especially	no

10	50.52	276.9	+is	yes
10	47.77	57.4	+like	no
10	47.21	60.8	+than	no
10	4.62	12.0	+b	no
10	40.80	43.1	+e.g	no
10	39.80	13.7	+treatment +with	no
10	39.29	81.0	+medication	no
10	35.33	2642.6	+to	no
10	3.30	279.5	+medicine	no
10	29.83	100.4	+treatment	no
10	19.53	34.8	+except	no
10	18.97	25.3	+plus	no
10	16.67	30.3	+i.e	no
10	100.00	293.1	+effects +of	yes
10	100.00	190.9	+effect +of	yes
10	100.00	101.5	+activity +of	yes
10	0.82	390908.0	+or	no
10	0.00	2.0	+namely	no
9	9.94	694.6	+as	no
9	7.43	26.8	+ie	no
9	6.03	106.8	+including	no
9	45.33	39.5	+eg	no
9	32.74	15.5	+mechanisms +of	no
9	19.09	42.6	+includes	no
9	16.91	3114.6	+with	no
9	11.49	2076.1	+a	no
9	10.81	1144.7	+with +a	no
9	100.00	98.3	+efficacy +of	yes
9	100.00	90.8	+action +of	yes
9	100.00	41.4	+drugs	yes
9	100.00	30.1	+agent	yes
9	0.59	3882.4	+for	no
9	0.05	8153.2	+in	no
9	0.00	6.3	+is +called	no
9	0.00	59.4	+but	no
9	0.00	5.4	+parathyroid	no
9	0.00	49.5	+i	no
9	0.00	44.1	+versus	no
9	0.00	42.3	+human	no
9	0.00	3.6	+r	no
9	0.00	31.5	+over	no
9	0.00	2.7	+related +to	no
9	0.00	2.7	+family	no

9	0.00	235.8	+by	no
9	0.00	19.8	+s	no
9	0.00	16.2	+bipolar	no
9	0.00	15.3	+w	no
9	0.00	13.5	+either	no
9	0.00	10.8	+depression	no
8	100.00	23.6	+actions +of	yes
7	100.00	8.3	+agents	yes
7	100.00	7.6	+agents +such +as	yes
7	100.00	37.6	+called	yes
7	100.00	28.3	+drugs +such +as	yes
7	100.00	23.7	+properties +of	yes
6	100.00	7.9	+compound	no
6	100.00	6.2	+effects	no
6	100.00	1975.2	+k	no
5	100.00	6.7	+property +of	no
5	100.00	3.1	+therapy	no
5	100.00	3.0	+drugs, +including	no
5	100.00	3.0	+drugs +including	no
5	100.00	24.4	+response +to	no
4	100.00	4.6	+mechanism +of	no
4	100.00	3.4	+effectiveness +of	no
4	100.00	34.6	+therapy +with	no
4	100.00	2.4	+properties	no
4	100.00	1.9	+effect +with	no
4	100.00	15.8	+potency +of	no
4	100.00	10.5	+medications	no
3	100.00	9.0	+effect	no
3	100.00	69.5	+c	no
3	100.00	2.8	+drug +called	no
3	100.00	2.5	+medications +including	no
3	100.00	2.3	+drugs +(eg	no
3	100.00	2.1	+collection	no
3	100.00	21.3	+d +or	no
3	100.00	1.5	+effect +when	no
3	100.00	1.5	+compounds	no
3	100.00	1.1	+agents, +including	no
3	100.00	10.7	+doses +of	no
2	100.00	5.8	+dose +of	no
2	100.00	4.7	+effect +of +intrathecal	no
2	100.00	3.6	+c +und	no
2	100.00	2.8	+barbiturate	no
2	100.00	2.5	+action	no

2	100.00	1.5	+efficacy +for	no
2	100.00	1.5	+effects +of +intrathecal	no
2	100.00	10.5	+d +called	no
2	100.00	0.6	+mix	no
2	100.00	0.6	+medications +include	no
1	100.00	5.6	+premix	no
1	100.00	2.5	+d +such +as	no
1	100.00	2.0	+agents +include	no
1	100.00	1.2	+effect +for	no
1	100.00	1.1	+ki	no
1	100.00	1.1	+k +called	no
1	100.00	1.0	+activities +of	no
1	100.00	0.8	+potencies +of	no
1	100.00	0.7	+regimen +containing	no
1	100.00	0.7	+efficacy	no
1	100.00	0.7	+d +metabolites	no
1	100.00	0.7	+activity +for	no
1	100.00	0.6	+sodium	no
1	100.00	0.6	+order	no
1	100.00	0.6	+levels	no
1	100.00	0.5	+drugs +carbamazepine	no
1	100.00	0.4	+regimens +with	no
1	100.00	0.4	+d +see	no
1	100.00	0.4	+d +analog	no
1	100.00	0.4	+although	no
1	100.00	0.4	+activity +as	no
1	100.00	0.3	+phenytoin	no
1	100.00	0.3	+k +analogue	no
1	100.00	0.3	+group	no
1	100.00	0.3	+c +buy	no
1	100.00	0.3	+also +called	no
1	100.00	0.2	+therapies	no
1	100.00	0.2	+techniques	no
1	100.00	0.2	+regimen	no
1	100.00	0.2	+muscle +relaxant	no
1	100.00	0.2	+medications +carbamazepine +and	no
1	100.00	0.2	+fluoxetine	no
1	100.00	0.2	+effect +than	no
1	100.00	0.2	+effects +of +intravenous	no
1	100.00	0.2	+effects +for	no
1	100.00	0.2	+doses	no
1	100.00	0.2	+deficiencies	no
1	100.00	0.2	+actions	no

8.33.3 hypo_hyper_plur

iteration range	precision	freq	pattern	accepted
10	99.96	184.2	+an	yes
10	89.57	53.4	+has	yes
10	79.83	22.6	+is +a +new	yes
10	76.72	585.7	+is	yes
10	69.65	89.7	+a +new	yes
10	63.94	938.8	+as	yes
10	59.99	10.1	+has +an	yes
10	5.87	608786.3	+and	no
10	57.58	73.1	+another	yes
10	55.96	71.3	+and +other	yes
10	39.99	12.8	+provides	no
10	38.68	25.0	+s	no
10	37.19	687.2	+on +the	no
10	36.57	199.8	+are	no
10	36.56	977.1	+natural	no
10	31.69	162.7	+on	no
10	3.14	396.7	+this	no
10	23.74	134.2	+or +other	no
10	21.36	23777.9	+the	no
10	19.86	32.7	+the +only	no
10	18.93	24.4	+had	no
10	10.00	4337.0	+for	no
10	100.00	2334.6	+is +an	yes
10	100.00	22.2	+is +an +effective	yes
9	9.72	2052.6	+a	no
9	7.67	6676.1	+at	no
9	6.94	3.8	+and +the +other	no
9	5.23	13.0	+in +terms +of	no
9	43.09	15.7	+or +another	no
9	3.84	46.7	+in +our	no
9	33.79	40.8	+ie	no
9	33.32	5.7	+has +no	no
9	32.02	41.0	+i.e	no
9	30.71	1128.4	+and +the	no
9	20.36	45.6	+plus	no
9	19.19	8426.9	+in	no
9	17.18	4.9	+however	no
9	11.74	26.6	+a +common	no
9	11.11	23.3	+or +any +other	no
9	11.11	2.0	+tablet	no
9	11.10	3.8	+tablets	no

9	11.08	10.9	+has +both	no
9	10.57	5.6	+two	no
9	10.52	47.7	+like	no
9	1.00	27.4	+as +the +first	no
9	100.00	48.8	+exerts +its	yes
9	100.00	114.1	+as +an	yes
9	0.00	8.0	+can +lead +to	no
9	0.00	73.2	+its	no
9	0.00	7.2	+the +common	no
9	0.00	7.2	+d	no
9	0.00	5.4	+for +chronic	no
9	0.00	4.5	+showed	no
9	0.00	40.5	+fever	no
9	0.00	233.1	+including +the	no
9	0.00	18683.8	+my	no
9	0.00	144.0	+causes	no
8	25.21	18.6	+have	no
6	100.00	6.6	+a +potent	no
5	100.00	8.8	+a +novel	no
5	100.00	5.0	+sodium	no
5	100.00	17.8	+is +a +novel	no
5	100.00	17.2	+tocopherol	no
4	100.00	9.2	+which +is	no
4	100.00	5.7	+induced	no
4	100.00	1.8	+used +as +an	no
4	100.00	1270.5	+source +of	no
3	100.00	6.0	+increases	no
3	100.00	54.4	+und	no
3	100.00	3.9	+have +similar	no
3	100.00	32.2	+has +demonstrated	no
3	100.00	3.1	+pro	no
3	100.00	2.9	+also +called	no
3	100.00	27.3	+enhances +the	no
3	100.00	2.3	+serum	no
3	100.00	2.3	+deficiency	no
3	100.00	23.4	+synthetic	no
3	100.00	1.7	+metabolism	no
3	100.00	1.6	+containing	no
3	100.00	1.5	+was +the +first	no
3	100.00	1.5	+a +standard	no
3	100.00	1.3	+other	no
3	100.00	0.8	+which +has	no
2	100.00	5.7	+ascorbyl +palmitate	no

2	100.00	5.2	+concentrations +and	no
2	100.00	3.6	+analgesic	no
2	100.00	2.8	+exhibits	no
2	100.00	21.1	+retinol	no
2	100.00	1.9	+analgesic +and	no
2	100.00	1.6	+concentrate	no
2	100.00	1.5	+aka	no
2	100.00	1.2	+and +tocopherol	no
2	100.00	1.1	+to +another	no
2	100.00	0.9	+has +potent	no
2	100.00	0.7	+a +known	no
2	100.00	0.6	+synonyms	no
2	100.00	0.6	+and +carbamazepine	no
2	100.00	0.4	+is +the +preferred	no
1	100.00	8.8	+acetate	no
1	100.00	5.8	+phosphate	no
1	100.00	4.4	+equivalents	no
1	100.00	4.3	+new	no
1	100.00	3.3	+an +established	no
1	100.00	2.3	+carotenoids	no
1	100.00	1.6	+and +novel	no
1	100.00	1.4	+is +an +efficacious	no
1	100.00	1.3	+phenytoin	no
1	100.00	1.2	+are +effective	no
1	100.00	1.0	+is +an +analgesic	no
1	100.00	1.0	+improved +the	no
1	100.00	1.0	+as +sole	no
1	100.00	0.7	+that +had	no
1	100.00	0.7	+hydrochloride	no
1	100.00	0.7	+folate	no
1	100.00	0.7	+displays	no
1	100.00	0.6	+riboflavin	no
1	100.00	0.5	+used +for	no
1	100.00	0.5	+to +increase	no
1	100.00	0.5	+produced +an	no
1	100.00	0.5	+narcotic	no
1	100.00	0.5	+as +a +potential	no
1	100.00	0.5	+and +a +tricyclic	no
1	100.00	0.4	+or +any +appropriate	no
1	100.00	0.4	+high +potency	no
1	100.00	0.4	+had +comparable	no
1	100.00	0.4	+antioxidants	no
1	100.00	0.4	+also +has	no

1	100.00	0.3	+tricyclic	no
1	100.00	0.3	+to +an	no
1	100.00	0.3	+may +decrease	no
1	100.00	0.3	+fat +soluble	no
1	100.00	0.3	+exhibited +an	no
1	100.00	0.3	+exerts	no
1	100.00	0.3	+deficient	no
1	100.00	0.3	+at +an	no
1	100.00	0.2	+to +other	no
1	100.00	0.2	+to +have	no
1	100.00	0.2	+rich	no
1	100.00	0.2	+produced	no
1	100.00	0.2	+phytonadione	no
1	100.00	0.2	+has +analgesic	no
1	100.00	0.2	+daily	no
1	100.00	0.2	+a +major	no

8.34 Unfiltered synonymy patterns precision scores

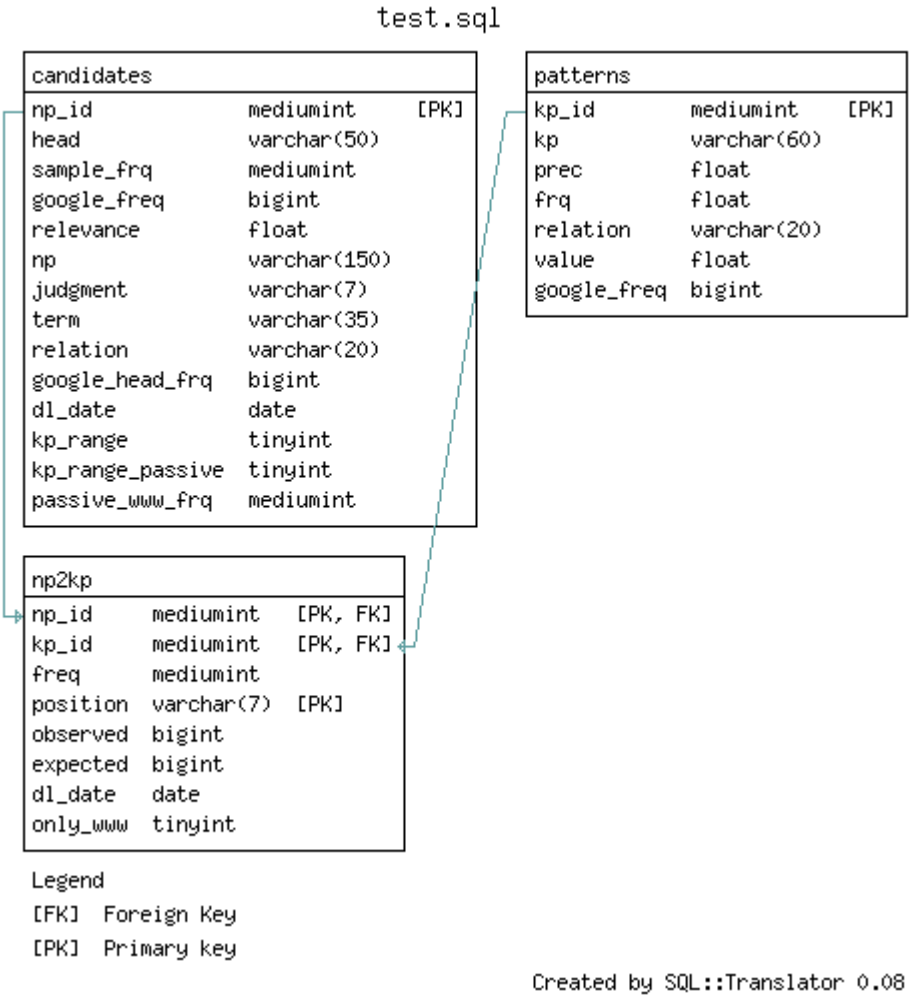
iteration range	precision	freq	pattern	accepted
9	100.00	67.2	+see	yes
9	100.00	45.1	+also +known +as	yes
9	100.00	23.8	+ie	yes
8	100.00	72.9	+means	yes
8	100.00	187.7	+also +called	yes
7	100.00	968.6	+acute	yes
6	100.00	20.2	+called	yes
6	100.00	15.8	+aka	yes
5	100.00	6.5	+is +also +called	yes
4	100.00	4.6	+mild	yes
4	100.00	3.4	+is +known +as	yes
4	100.00	2.9	+refers +to	yes
4	100.00	19.1	+severe	yes
4	100.00	1.7	+was +defined +as	yes
3	100.00	6.0	+this	no
3	100.00	4.5	+nausea	no
3	100.00	3.8	+from	no
3	100.00	3.3	+defined +as	no
3	100.00	1.9	+associated +with	no
3	100.00	17.7	+during	no
3	100.00	12.8	+recurrent	no
3	100.00	0.9	+in +patients +with	no
2	100.00	8.7	+extreme	no

2	100.00	7.0	+is +an	no
2	100.00	6.8	+early	no
2	100.00	6.1	+affects	no
2	100.00	5.2	+postural	no
2	100.00	48.2	+pregnancy	no
2	100.00	4.7	+high	no
2	100.00	45.1	+abnormally	no
2	100.00	3.2	+unusually	no
2	100.00	2.8	+sudden	no
2	100.00	2.8	+or +acute	no
2	100.00	2.7	+caused +by	no
2	100.00	2.3	+excessive	no
2	100.00	22.0	+multiple	no
2	100.00	2.1	+includes	no
2	100.00	2.1	+hypertension	no
2	100.00	1.6	+an	no
2	100.00	15.6	+ectopic	no
2	100.00	1.1	+unlike	no
2	100.00	1.1	+rather +than	no
2	100.00	1.0	+gravidity	no
2	100.00	0.9	+or +psychogenic	no
2	100.00	0.7	+vulvar	no
2	100.00	0.7	+and +recurrent	no
2	100.00	0.6	+uf	no
1	100.00	6.3	+throughout	no
1	100.00	5.6	+end +stage	no
1	100.00	4.5	+resulting +from +normal	no
1	100.00	4.4	+called +a	no
1	100.00	2.4	+headache	no
1	100.00	2.2	+normal	no
1	100.00	1.8	+is +characterized +by +frequent	no
1	100.00	1.4	+patients +with	no
1	100.00	1.2	+nor	no
1	100.00	1.2	+generalized	no
1	100.00	1.1	+type +i	no
1	100.00	1.1	+stillbirth	no
1	100.00	1.1	+first	no
1	100.00	1.0	+or +postural	no
1	100.00	1.0	+coronary	no
1	100.00	0.9	+whereas	no
1	100.00	0.9	+was +measured +with +the +baseline	no
1	100.00	0.8	+terminated	no
1	100.00	0.8	+is +called +an +acute	no

1	100.00	0.7	+prolonged	no
1	100.00	0.7	+now +called	no
1	100.00	0.6	+to +prolong	no
1	100.00	0.6	+testing	no
1	100.00	0.6	+rash	no
1	100.00	0.6	+multifetal	no
1	100.00	0.6	+component	no
1	100.00	0.6	+attacks	no
1	100.00	0.5	+stroke	no
1	100.00	0.5	+rop	no
1	100.00	0.5	+nos	no
1	100.00	0.5	+exercise	no
1	100.00	0.4	+medicine +this +medicine +helps +lower +an	no
1	100.00	0.4	+ischemic +heart +disease	no
1	100.00	0.3	+threatened	no
1	100.00	0.3	+since	no
1	100.00	0.3	+may +differ +from	no
1	100.00	0.3	+is +acute +and +severe +with +symptoms +of	no
1	100.00	0.3	+intractable	no
1	100.00	0.3	+in +pregnancy	no
1	100.00	0.2	+without +actual	no
1	100.00	0.2	+with +exertion	no
1	100.00	0.2	+which +means	no
1	100.00	0.2	+tummy	no
1	100.00	0.2	+temperature	no
1	100.00	0.2	+rlf	no
1	100.00	0.2	+often +called	no
1	100.00	0.2	+natural	no
1	100.00	0.2	+may +prolong	no
1	100.00	0.2	+massive	no
1	100.00	0.2	+is +acute	no
1	100.00	0.2	+in +rats	no
1	100.00	0.2	+any	no
10	99.13	6120.3	+or	yes
10	31.23	672.6	+the	no
10	26.98	19.6	+known +as	no
10	22.51	372.1	+is	no
10	20.72	6.6	+vs	no
10	19.89	6011.5	+and	no
9	19.68	5.5	+however	no
9	14.25	408.7	+of	no
9	13.10	8.9	+due +to	no
10	10.92	240.2	+including	no

9	10.60	602.0	+in	no
10	9.76	17.5	+if	no
10	8.03	47.1	+skin	no
10	6.80	42.3	+chronic	no
9	6.43	28.9	+with +normal	no
9	6.12	235.1	+of +the	no
9	5.74	15.3	+after	no
10	5.12	98.8	+like	no
10	4.01	144.3	+her	no
9	3.73	85.2	+induced	no
10	1.60	44.5	+eg	no
9	1.03	25.4	+some	no
9	0.46	121.9	+had	no
9	0.03	969.4	+without	no
9	0.00	95.4	+include	no
9	0.00	9.0	+rt	no
9	0.00	83.1	+no	no
9	0.00	6.9	+yellow	no
9	0.00	3.6	+two	no
9	0.00	3.6	+other +than	no
9	0.00	3.6	+ans	no
9	0.00	35.1	+but +no	no
9	0.00	3.3	+watery	no
9	0.00	2.7	+especially	no
9	0.00	252.9	+prevents	no

8.35 Database schema



8.36 "induces" KP performance (manual evaluation)

F	precision	recall	pattern_id
0.20	0.69	0.11	may cause_1
0.18	0.58	0.11	can cause_71
0.17	0.46	0.11	can lead to_12
0.16	0.78	0.09	can induce_25
0.13	0.55	0.07	induces_4
0.12	0.79	0.07	may induce_35

0.12	0.63	0.07	promotes_28
0.12	0.37	0.07	leads to_33
0.11	0.85	0.06	to induce_2
0.11	0.64	0.06	can also cause_26
0.11	0.45	0.06	causes_70
0.11	0.30	0.07	cause_6
0.10	0.60	0.05	to cause_5
0.10	0.57	0.05	may lead to_22
0.10	0.50	0.05	could cause_67
0.09	0.68	0.05	can result in_29
0.08	0.36	0.05	causing_18
0.07	0.34	0.04	will cause_16
0.07	0.21	0.04	caused_46
0.06	0.79	0.03	does not cause_8
0.06	0.61	0.03	induce_11
0.06	0.27	0.03	can produce_24
0.06	0.26	0.03	produces_10
0.05	0.77	0.03	induced_69
0.05	0.62	0.03	provokes_39
0.05	0.62	0.03	produce_3
0.05	0.30	0.03	associated with_57
0.04	1.00	0.02	overdose can cause_61
0.04	1.00	0.02	may aggravate_44
0.04	1.00	0.02	commonly causes_48
0.04	0.67	0.02	when you have_47
0.04	0.56	0.02	which causes_15
0.04	0.50	0.02	in producing_55
0.04	0.38	0.02	was used for_51
0.04	0.38	0.02	and develop_68
0.04	0.35	0.02	that can cause_32
0.04	0.32	0.02	may experience_31
0.04	0.26	0.02	leading to_21
0.04	0.20	0.02	to produce_13
0.03	0.83	0.01	does not produce_19
0.03	0.50	0.01	overdose include_7
0.03	0.40	0.01	maintains_62
0.03	0.31	0.01	has_65
0.02	0.08	0.01	resulted in_37
0.01	1.00	0.01	without causing_30
0.01	1.00	0.01	which promotes_36
0.01	1.00	0.01	consumption include_45
0.01	0.50	0.01	which induces_9
0.01	0.50	0.01	for inducing_34

0.01	0.45	0.01	aggravates_58
0.01	0.33	0.01	will induce_17
0.01	0.15	0.01	in maintaining_63
0.01	0.12	0.01	produced_23
0.01	0.08	0.01	did not cause_66
NA	NA	NA	usually results in_52
NA	NA	NA	tension ·_59
NA	NA	NA	see_60
NA	NA	NA	retention_43
NA	NA	NA	or induce_14
NA	NA	NA	if inducing_64
NA	NA	NA	can itself cause_42
NA	NA	NA	because inducing_49
NA	NA	NA	and induce_38
0	0	0	will produce_50
0	0	0	resulting in_53
0	0	0	poisoning include_20
0	0	0	poisoning_27
0	0	0	it causes_40
0	0	0	intake is_54
0	0	0	has been_56
0	0	0	characterized by_41

8.37 "may_prevent" KP performance (manual evaluation)

F	precision	recall	pattern_id
0.16	0.42	0.10	helps prevent_75
0.16	0.35	0.10	prevents_79
0.14	0.24	0.10	protects against_109
0.14	0.24	0.10	decreased_96
0.13	0.12	0.14	reduced_88
0.12	0.28	0.08	could reduce_133
0.11	0.59	0.06	significantly reduces_103
0.11	0.53	0.06	in preventing_77
0.11	0.50	0.06	may decrease_166
0.10	0.30	0.06	prevent_87
0.10	0.12	0.08	reduce_90
0.10	0.11	0.10	inhibits_117
0.10	0.10	0.10	significantly reduced_119
0.09	0.19	0.06	can prevent_99
0.09	0.17	0.06	decreases_84
0.08	0.62	0.04	help prevent_85
0.08	0.62	0.04	cuts_78

0.08	0.12	0.06	prevented_93
0.07	0.49	0.04	to reduce_82
0.07	0.19	0.04	may prevent_136
0.07	0.10	0.06	provide_165
0.05	0.07	0.04	lowers_123
0.05	0.06	0.04	can reduce_172
0.05	0.04	0.06	cut_153
0.04	0.50	0.02	would decrease_137
0.04	0.50	0.02	to combat_92
0.04	0.50	0.02	attenuates_121
0.04	0.44	0.02	for preventing_74
0.04	0.36	0.02	developed_141
0.04	0.33	0.02	preventing_112
0.04	0.33	0.02	alleviates_108
0.04	0.25	0.02	works against_162
0.04	0.22	0.02	reduces_171
0.04	0.20	0.02	significantly decreased_150
0.04	0.20	0.02	provided_129
0.04	0.20	0.02	combats_107
0.03	0.08	0.02	inhibited_155
0.03	0.07	0.02	helps_106
0.03	0.04	0.02	to prevent_72
0.02	0.03	0.02	in treating_86
NA	NA	NA	will relieve_134
NA	NA	NA	to suppress_154
NA	NA	NA	to relieve_81
NA	NA	NA	to alleviate_101
NA	NA	NA	therapy prevents_163
NA	NA	NA	remedy_139
NA	NA	NA	relieved_104
NA	NA	NA	relieve_80
NA	NA	NA	numbs_151
NA	NA	NA	minimizes_135
NA	NA	NA	may ease_97
NA	NA	NA	lessens_142
NA	NA	NA	in relieving_124
NA	NA	NA	in fighting_125
NA	NA	NA	heals_145
NA	NA	NA	had significantly less_116
NA	NA	NA	group reported_140
NA	NA	NA	for relieving_73
NA	NA	NA	for decreasing_167
NA	NA	NA	fighting_127

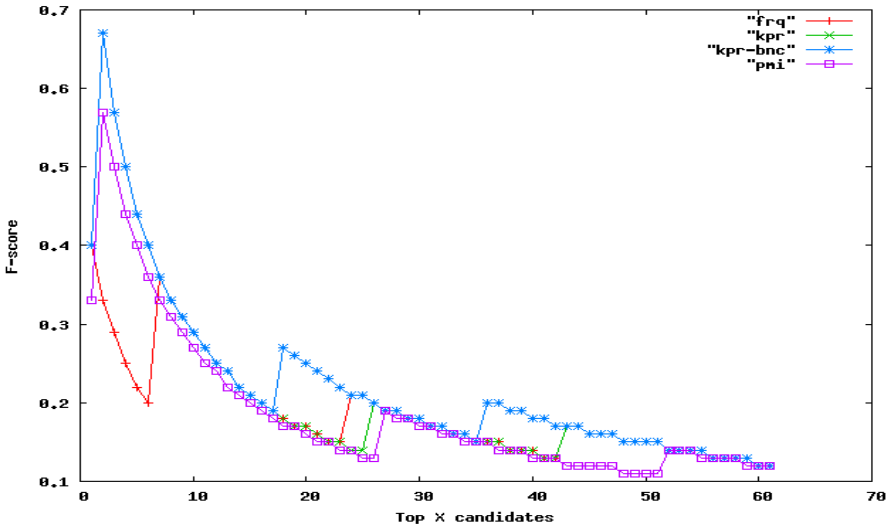
NA	NA	NA	eases_158
NA	NA	NA	ease_156
NA	NA	NA	does not cure_152
NA	NA	NA	deters_149
NA	NA	NA	can relieve_105
NA	NA	NA	at preventing_113
NA	NA	NA	also relieves_161
NA	NA	NA	after developing_170
0	0	0	will prevent_100
0	0	0	was_138
0	0	0	use in_110
0	0	0	used for_111
0	0	0	treated_122
0	0	0	to treat_89
0	0	0	to kill_146
0	0	0	to cure_147
0	0	0	to control_91
0	0	0	that prevents_115
0	0	0	showing_164
0	0	0	reverses_144
0	0	0	relieves_76
0	0	0	provides_128
0	0	0	makes_132
0	0	0	in decreasing_130
0	0	0	improves_98
0	0	0	helps relieve_143
0	0	0	has_120
0	0	0	group had_126
0	0	0	group experienced_131
0	0	0	for treating_83
0	0	0	following_148
0	0	0	eliminates_118
0	0	0	eased_159
0	0	0	diminishes_114
0	0	0	containing_94
0	0	0	caused_157
0	0	0	can cure_160
0	0	0	can_168
0	0	0	based_95
0	0	0	alleviated_169
0	0	0	affects_102

8.38 ISA KP performance (manual evaluation)

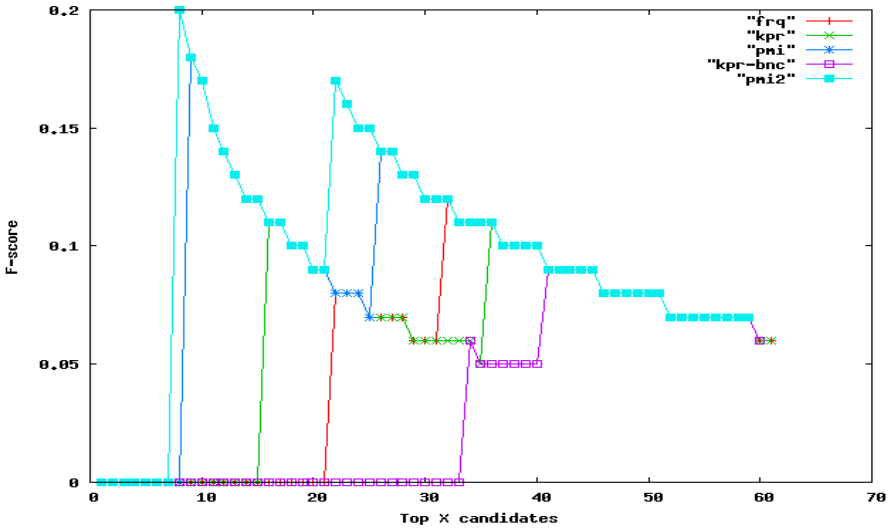
F	precision	recall	pattern_id
0.44	0.88	0.29	such as_341
0.42	0.89	0.28	like_323
0.41	0.87	0.27	such as_321
0.35	0.73	0.23	and other_342
0.32	0.82	0.20	is an_350
0.30	0.69	0.19	and other_360
0.26	0.88	0.15	agents such as_332
0.25	0.94	0.14	including_322
0.25	0.70	0.15	or other_343
0.23	0.91	0.13	drugs such as_334
0.23	0.55	0.14	efficacy of_326
0.21	0.54	0.13	effect of_337
0.19	0.49	0.12	effects of_336
0.19	0.49	0.12	action of_327
0.18	0.75	0.10	called_333
0.18	0.69	0.10	include_325
0.18	0.56	0.11	activity of_338
0.17	0.57	0.10	is an effective_351
0.15	0.72	0.08	agent_329
0.14	0.97	0.08	e.g._320
0.14	0.84	0.08	i.e._324
0.14	0.40	0.08	as_357
0.12	0.46	0.07	as_347
0.12	0.44	0.07	properties of_335
0.12	0.35	0.07	is_355
0.12	0.27	0.08	as an_349
0.11	0.73	0.06	is a new_354
0.10	0.57	0.06	with other_345
0.08	0.30	0.05	actions of_330
0.07	0.36	0.04	drug_339
0.06	0.53	0.03	other_344
0.06	0.38	0.03	has_353
0.06	0.37	0.03	an_352
0.06	0.35	0.03	see_346
0.06	0.21	0.03	has an_358
0.04	0.67	0.02	another_359
0.04	0.58	0.02	agents_331
0.04	0.53	0.02	drugs_328
0.04	0.18	0.02	exerts its_348
0.02	0.03	0.01	a new_356
0.00	0.00	0.00	activity_340

8.39 Synonymy ranking schemes

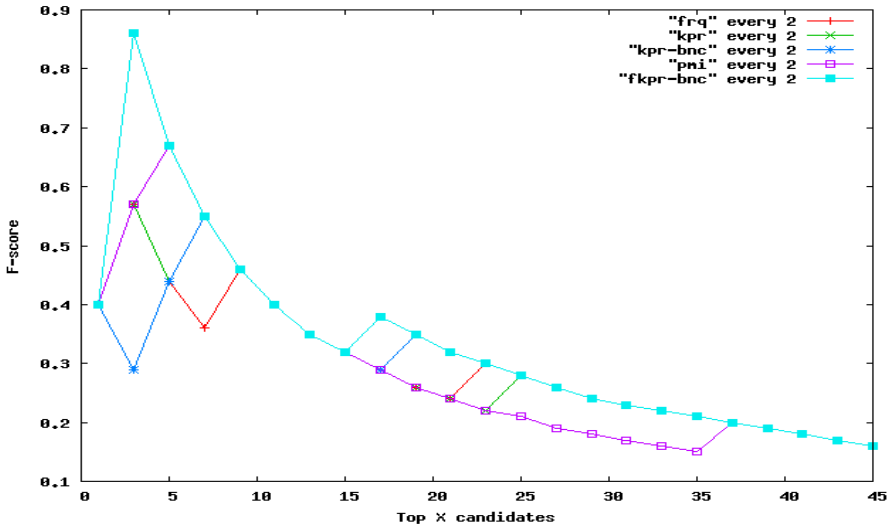
8.39.1 Vitamin C - F-scores



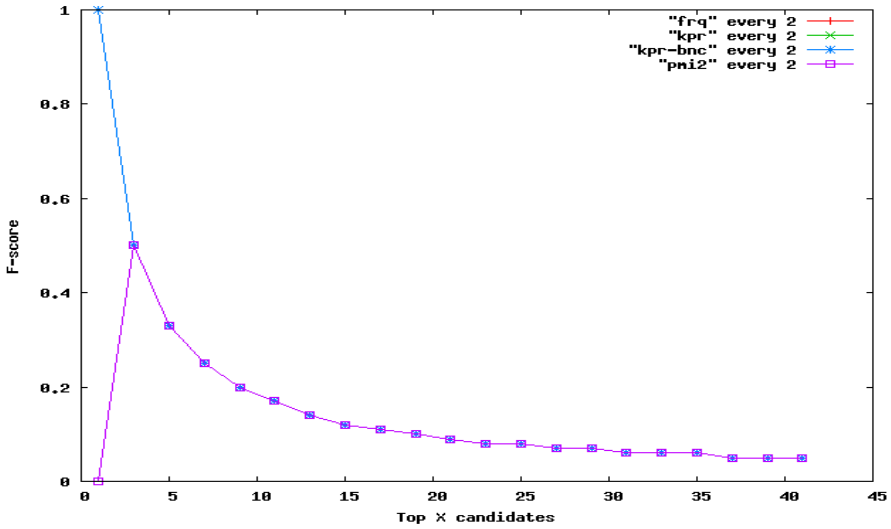
8.39.2 Progesterone - F-scores



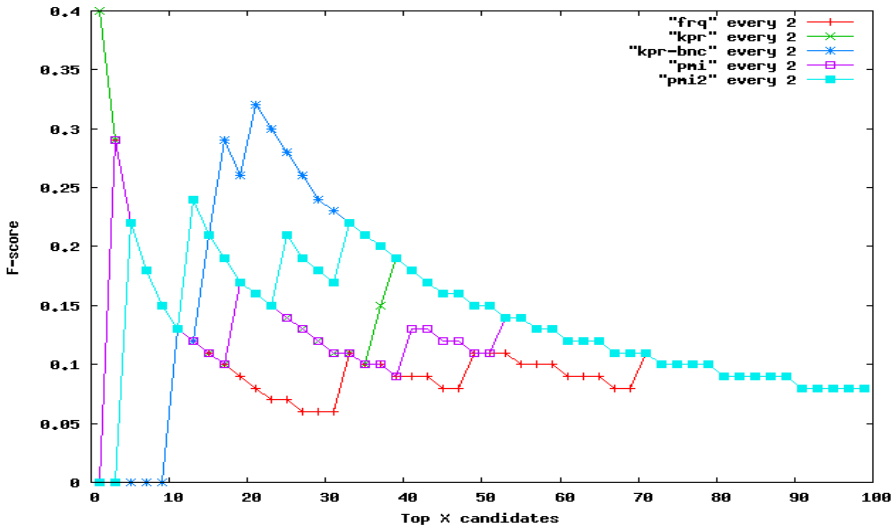
8.39.3 Formaldehyde - F-scores



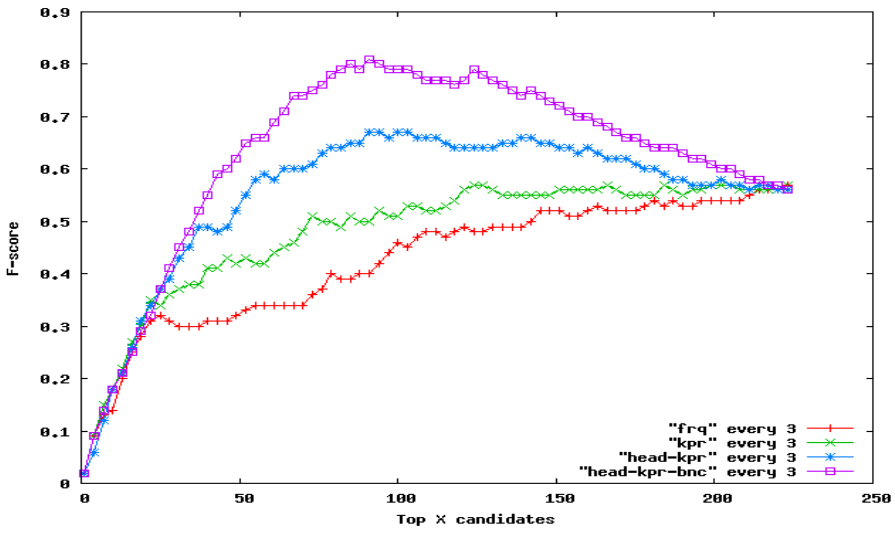
8.39.4 Lactose - F-scores



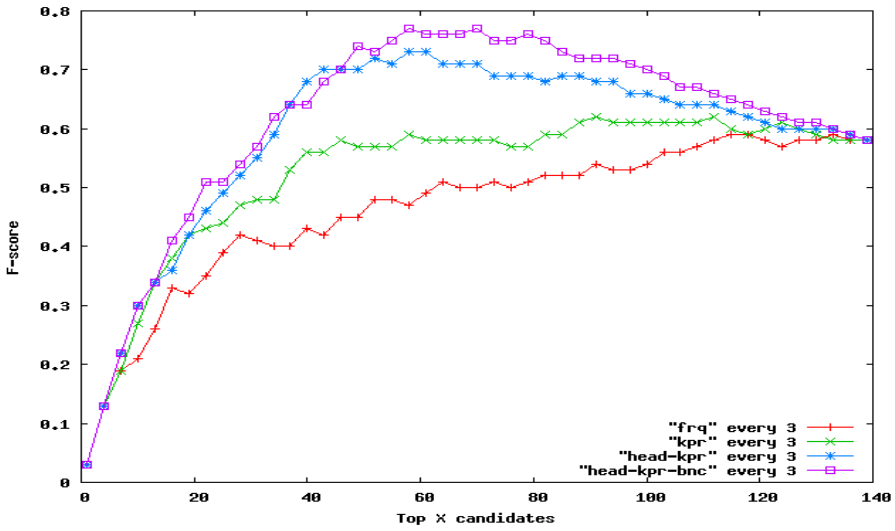
8.39.5 Glucose - F-scores



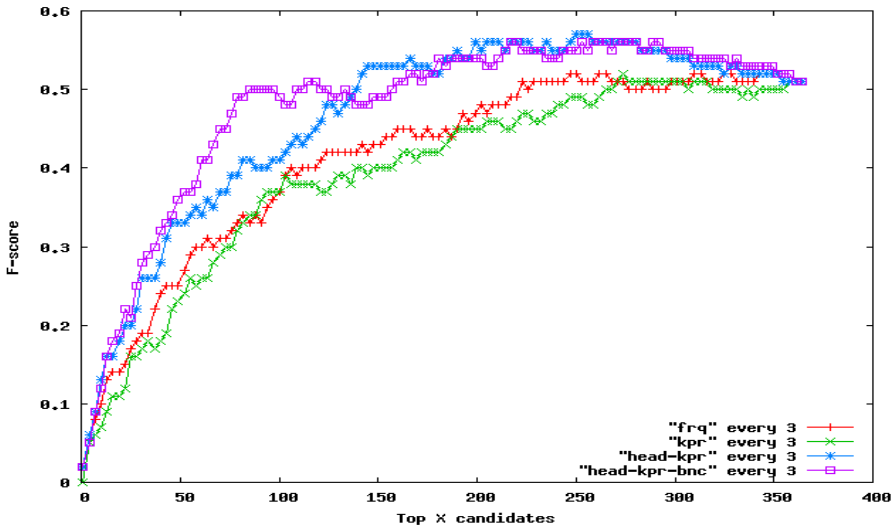
8.40 X ISA antipsychotic - F-scores



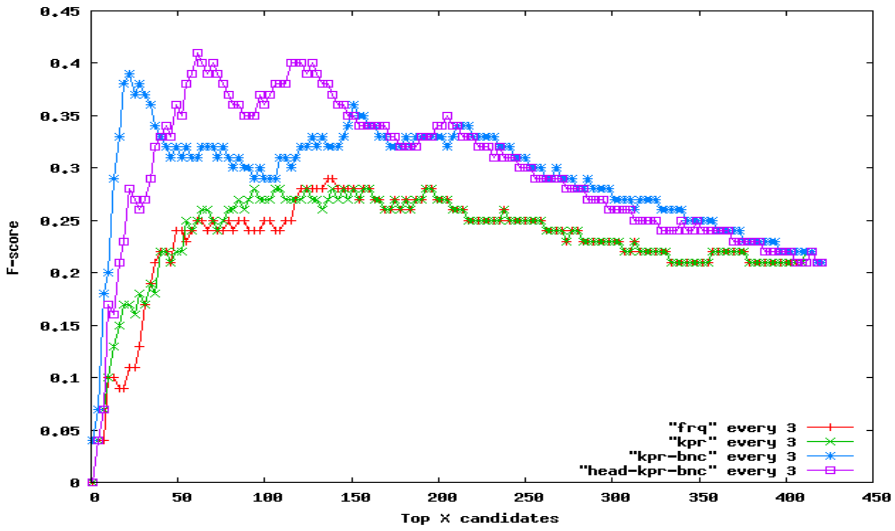
8.41 haloperidol ISA X - F-scores



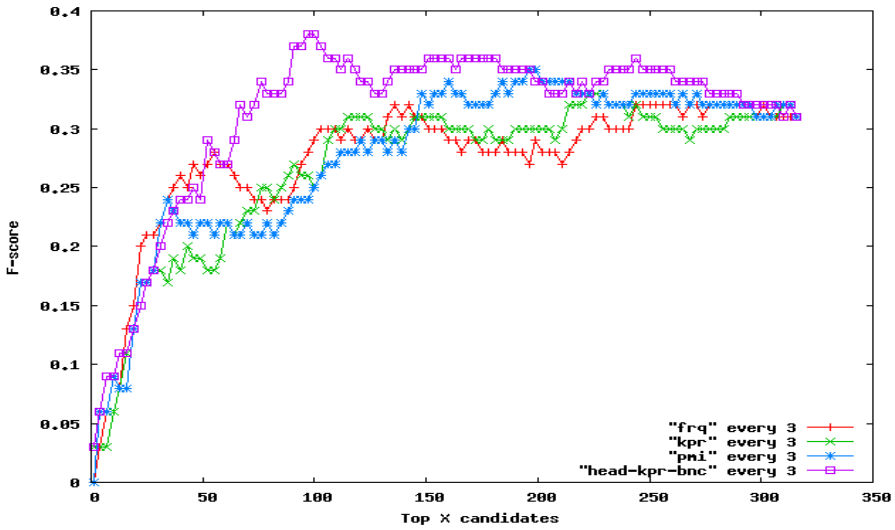
8.42 aspirin induces X - F-scores



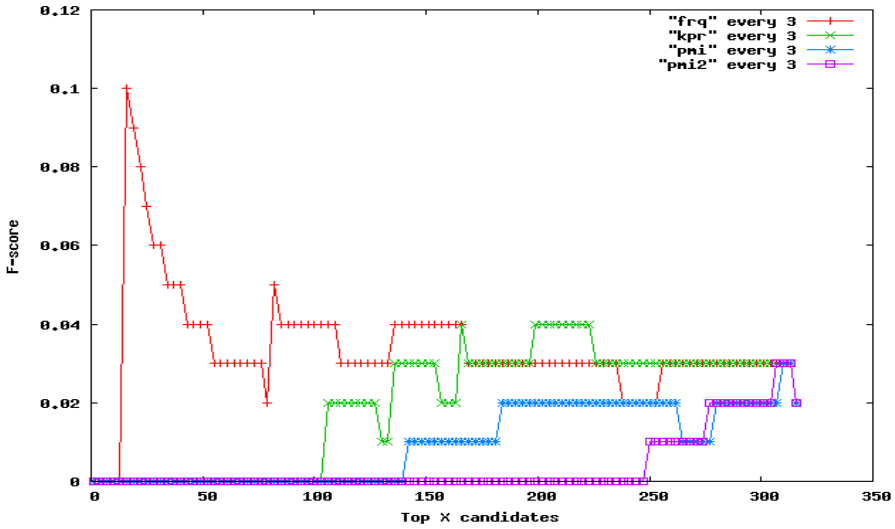
8.43 selenium may_prevent X - F-scores



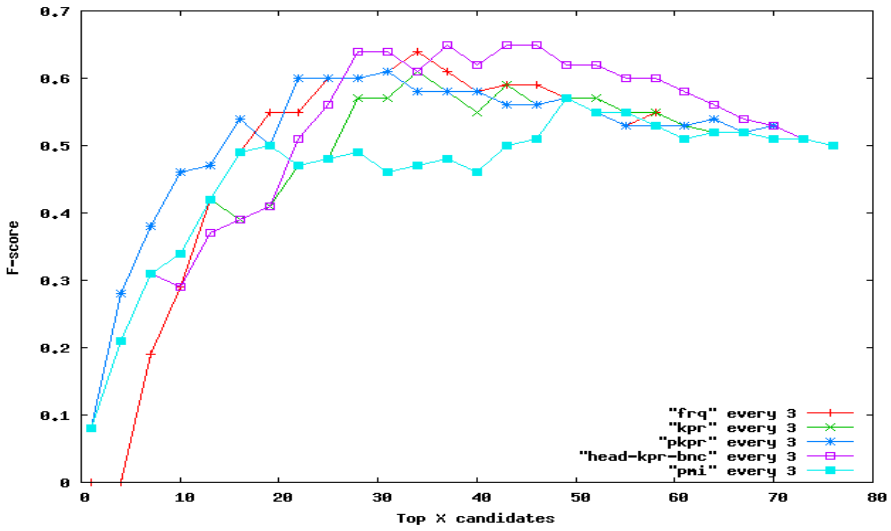
8.44 X induces vomiting - F-scores



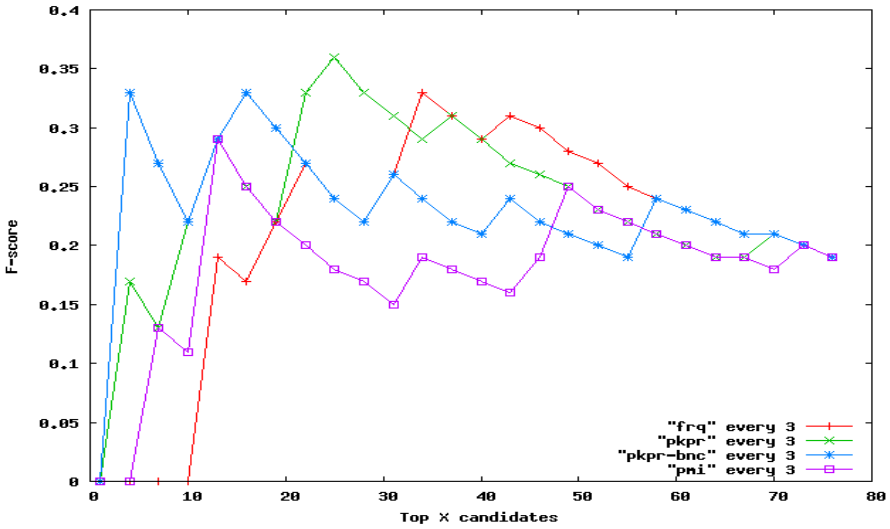
8.44.1 {drugs} induces vomiting - F-scores



8.45 X induces emesis - F-scores

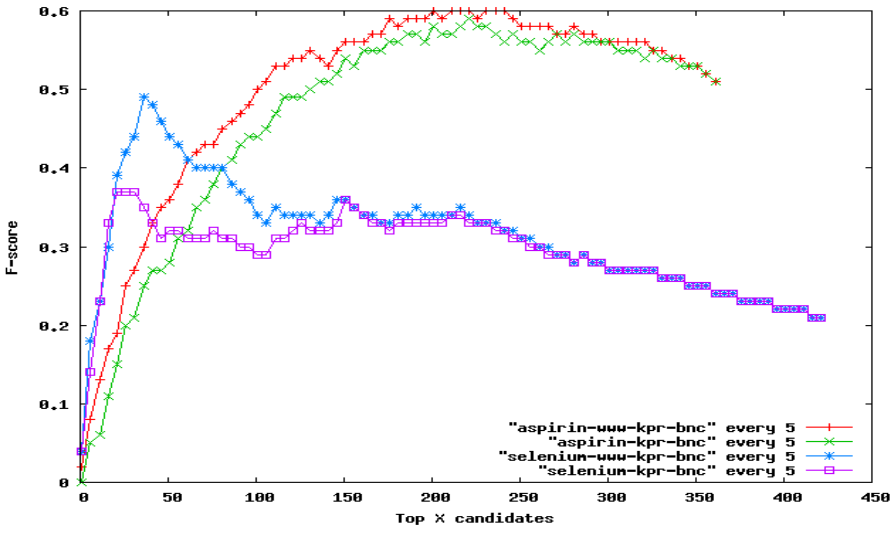


8.45.1 {drugs} induces emesis - F-scores

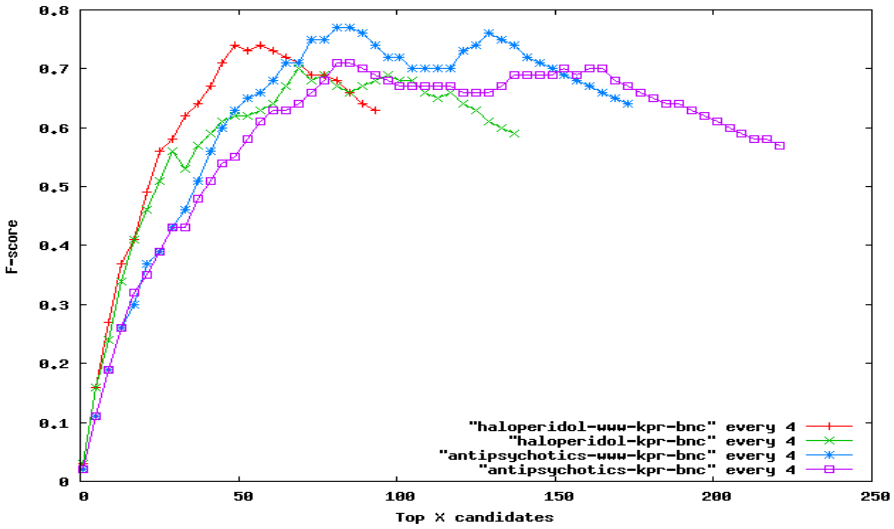


8.46 Correlation with WWW

8.46.1 causal experiments - F-scores



8.46.2 ISA experiments - F-scores



References

[Agbago and Barrière, 2005] Agbago, A. and Barrière, C. (2005). Corpus construction for terminology. In *Proceedings of Corpus Linguistics 2005*.

[Agichtein and Gravano, 2000] Agichtein, E. and Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the International Conference on Development and Learning (ICDL 2000)*.

[Ahmad, 1993] Ahmad, K. (1993). Pragmatics of specialist terms: The acquisition and representation of terminology. In *Machine Translation and the Lexicon, 3rd EAMT Workshop proceedings*.

[Ahmad and Fulford, 1992] Ahmad, K. and Fulford, H. (1992). Semantic relations and their use in elaborating terminology. Technical report, University of Surrey, Computing Sciences.

[Alfonseca et al., 2006a] Alfonseca, E., Castells, P., Okumura, M., and Ruiz-Casado, M. (2006a). A rote extractor with edit distance-based generalization and multi-corpora precision calculation. In *Proceedings of ACL 2006*.

[Alfonseca et al., 2006b] Alfonseca, E., Ruiz-Casado, M., Okumura, M., and Castells, P. (2006b). Towards large-scale non-taxonomic relation extraction: Estimating the precision of rote extractors. In *Proceedings of the 2nd Workshop on Ontology Learning and Population*.

[Ananiadou and McNaught, 2006] Ananiadou, S. and McNaught, J. (2006). *Text Mining for Biology and Biomedicine*. Artech House.

[Aronson, 2001] Aronson, A. R. (2001). Effective mapping of biomedical text to the umls metathesaurus: The metamap program. In *Proceedings of AMIA 201*.

[Barrière, 2001] Barrière, C. (2001). Investigating the causal relation in informative texts. *Terminology*, 7(2):135–154.

[Barrière, 2002] Barrière, C. (2002). Hierarchical refinement and representation of the causal relation. *Terminology*, 8(1):91–111.

[Biber et al., 1998] Biber, D., Conrad, S., and Reppen, R. (1998). *Corpus Linguistics - Investigating Language Structure and Use*. Cambridge University Press.

[Biber et al., 1999] Biber, D., Johansson, S., Leech, G., Conrad, S., and Finegan, E. (1999). *Longman Grammar of Spoken and Written English*. Longman.

[Bodenreider, 2001] Bodenreider, O. (2001). Circular hierarchical relationships in the umls: Etiology, diagnosis, treatment, complications and prevention. In *Proceedings of American Medical Informatics Association (AMIA 2001)*.

[Bodenreider et al., 2002a] Bodenreider, O., Mitchell, J. A., and McCray, A. T. (2002a). Evaluation of the umls as a terminology and knowledge resource for biomedical informatics. In *Proceedings of the AMIA Annual Symposium 2002*, pages 61–65.

[Bodenreider et al., 2002b] Bodenreider, O., Rindflesch, T. C., and Burgun, A. (2002b). Unsupervised, corpus-based method for extending a biomedical terminology. In *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain*, pages 53–60.

[Bourigault, 1992] Bourigault, D. (1992). Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of COLING-92*.

[Bowden et al., 1996] Bowden, P. R., Halstead, P., and Rose, T. G. (1996). Extracting conceptual knowledge from text using explicit relation markers. In *Proceedings of the 9th European Knowledge Acquisition Workshop on Advances in Knowledge Acquisition*, pages 147–162.

[Brender, 2006] Brender, J. (2006). *Evaluation Methods for Health Informatics*. Academic Press.

[Brin, 1998] Brin, S. (1998). Extracting patterns and relations from the world wide web. In *Proceedings of the International Conference on Extending Database Technology (EDBT 1998)*.

[Buitelaar et al., 2005] Buitelaar, P., Cimiano, P., and Magnini, B. (2005). *Ontology learning from text: methods, evaluation and applications*. IOS Press.

[Burgun and Bodenreider, 2001] Burgun, A. and Bodenreider, O. (2001). Comparing terms, concepts and semantic classes in wordnet and the unified medical language system. In *Proceedings of NAACL 2001*, pages 77–82.

[Byrd and Ravin, 1999] Byrd, R. and Ravin, Y. (1999). Identifying and extracting relations in text. In *Proceedings of NLDB 1999*.

[Cabré, 2000] Cabré, M. T. (2000). Elements for a theory of terminology: Towards an alternative paradigm. *Terminology*, 6(1):35–57.

[Cabré, 2003] Cabré, M. T. (2003). Theories of terminology - their description, prescription and explanation. *Terminology*, 9(2):163–199.

[Charniak and Berland, 1999] Charniak, E. and Berland, M. (1999). Finding parts in very large corpora. In *Proceedings of ACL*, pages 57–64.

[Cheng et al., 2006] Cheng, W., Greaves, C., and Warren, M. (2006). From n-gram to skipgram to concgram. *International Journal of Corpus Linguistics*, 11:4:411–433.

[Christensen, 2002] Christensen, L. W. (2002). Danish verbs as knowledge probes in corpus-based terminology work. *LSP and Professional Communication*, 2(2):77–94.

[Christensen, 2005] Christensen, L. W. (2005). Hvordan ord sporer termer og andre terminologiske oplysninger. In *Proceedings of NORDTERM 2005*, pages 83–94.

[Cimiano and Staab, 2005] Cimiano, P. and Staab, S. (2005). Learning concept hierarchies from text with a guided hierarchical clustering algorithm. In *Proceedings of the workshop Learning and Extending Lexical Ontologies by using Machine Learning Methods (ICML)*.

[Clark et al., 2004] Clark, P., Thompson, J., and Porter, B. (2004). *Handbook on Ontologies*, chapter Knowledge Patterns, pages 191–207. Springer.

[Cohen and Hersh, 2005] Cohen, A. M. and Hersh, W. R. (2005). A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1):57–71.

[Cohen, 1999] Cohen, W. W. (1999). Interaction of heterogeneous databases without common domains using queries based on textual similarity. In *Proceedings of the Joint SIG-DAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

[Coxhead, 2000] Coxhead, A. (2000). A new academic word list. *TESOL Quarterly*, 34(2):213–238.

[Decker et al., 1999] Decker, S., Erdmann, M., Fensel, D., and Studer, R. (1999). Ontobroker: Ontology based access to distributed and semi-structured information. In *Proceedings of DS-8*, pages 351–369.

[Drouin, 2003] Drouin, P. (2003). Term extraction using non-technical corpora as a point of leverage. *Terminology*, 9(1):99–115.

[Etzioni et al., 2004] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Sodeland, S., Weld, D., and Yates, A. (2004). Web-scale information extraction in knowitall. In *Proceedings of WWW-04*.

[Evert, 2004] Evert, S. (2004). *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. PhD thesis, University of Stuttgart.

[Faatz and Steinmetz, 2005] Faatz, A. and Steinmetz, R. (2005). *Ontology learning from text: methods, evaluation and applications*, chapter An Evaluation Framework for Ontology Enrichment, pages 77–91. IOS Press.

[Fisher, 1987] Fisher, D. H. (1987). Knowledge acquisition via conceptual clustering. *Machine Learning*, 2:139–172.

[Fleiss, 1971] Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

[Frawley et al., 1992] Frawley, W. J., Piatetsky-Shapiro, G., and Matheus, C. J. (1992). Knowledge discovery in databases: An overview. *Ai Magazine*, Fall:57–70.

[Gaizauskas et al., 2003] Gaizauskas, R., Demetriou, G., Artymiuk, P. J., and Willett, P. (2003). Protein structures and information extraction from biological texts: The pasta system. *Bioinformatics*, 19(1):135–143.

[Gillam, 2004] Gillam, L. (2004). *Systems of concepts and their extraction from text*. PhD thesis, School of Electronics and Physical Sciences, University of Surrey.

[Gillam et al., 2005] Gillam, L., Tariq, M., and Ahmad, K. (2005). Terminology and the construction of ontology. *Terminology*, 11:1:55–81.

[Girju and Moldovan, 2002] Girju, R. and Moldovan, D. (2002). Text mining for causal relations. In *Proceedings of the 15th Florida Artificial Intelligence Research Society (FLAIRS) conference*, pages 360–364.

[Gómez-Pérez and Manzano-Macho, 2005] Gómez-Pérez, A. and Manzano-Macho, D. (2005). An overview of methods and tools for ontology learning from texts. *The Knowledge Engineering Review*, 19(3):187–212.

[Guarino et al., 1999] Guarino, N., Masolo, C., and Vetere, G. (1999). Ontoseek: Content-based access to the web. In *Proceedings of IEEE Intelligent Systems*, pages 70–80.

[Halskov, 2005a] Halskov, J. (2005a). Diasketching as a filter in web-based term extraction systems. In *Proceedings of Terminology and Knowledge Engineering 2005 (TKE 2005)*, pages 397–408.

[Halskov, 2005b] Halskov, J. (2005b). Nettet som specialiseret korpus. In *Proceedings of Nordterm 2005*, pages 42–55.

[Halskov, 2005c] Halskov, J. (2005c). Probing the properties of determinologization - the diasketch. *LAMBDA (Dept. of Computational Linguistics, Copenhagen Business School)*, 29.

[Halskov, 2005d] Halskov, J. (2005d). Web-based extraction of terminology and the issue of determinologization. In *Proceedings of the 3rd Computational Linguistics in the North-East workshop (CLiNE 2005)*.

[Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING-92*, pages 539–545.

[Hearst, 1999] Hearst, M. A. (1999). Untangling text data mining. In *Proceedings of ACL '99. The 37th annual meeting of the Association for Computational Linguistics*.

[Jacquemin, 1994] Jacquemin, C. (1994). Fastr: A unification grammar and a parser for terminology extraction from large corpora. In *Proceedings of IA-94*.

[Kageura, 2002] Kageura, K. (2002). *The Dynamics of Terminology - a descriptive theory of term formation and terminological growth*. John Benjamins.

[Kennedy, 1998] Kennedy, G. (1998). *An Introduction to Corpus Linguistics*. Longman.

[Kilgarriff and Grefenstette, 2003] Kilgarriff, A. and Grefenstette, G. (2003). Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29(3).

[Klavans and Muresan, 2000] Klavans, J. L. and Muresan, S. (2000). Evaluation of definder: A system to mine definitions from consumer-oriented medical text. In *Proceedings of the Annual Fall Symposium of the American Medical Informatics Association*.

[Kragh, 1995] Kragh, B. (1995). *Linguistic Aspects of Technical Language*. Copenhagen Business School.

[Kroeze et al., 2003] Kroeze, J. H., Matthee, M. C., and Bothma, T. J. D. (2003). Differentiating data- and text-mining terminology. In *Proceedings of SAICSIT 2003*.

[Kucera and Francis, 1969] Kucera, H. and Francis, W. N. (1969). Computational analysis of present-day american english. *International Journal of American Linguistics*, 35:71–75.

[Kumar and Smith, 2003] Kumar, A. and Smith, B. (2003). *The Unified Medical Language System and the Gene Ontology: Some Critical Reflections*, volume Lecture Notes in Artificial Intelligence of *Advances in Artificial Intelligence*, pages 135–148. Springer.

[Landis and Koch, 1977] Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174.

[Lyons, 1977] Lyons, J. (1977). *Semantics*. Cambridge University Press.

[Madsen, 1991] Madsen, B. N. (1991). In terms of concepts. *Copenhagen Studies in Language*, 14:67–91.

[Madsen, 2000] Madsen, B. N. (2000). *I terminologins tjænst*, chapter Alting på sin plads og plads til alting: om at ordne og udnytte viden om verden, pages 71–91. University of Vaasa.

[Madsen, 2007] Madsen, B. N. (2007). *Indeterminacy in LSP and Terminology: Studies in honour of Heribert Picht*, chapter Ontologies and indeterminacy. John Benjamins.

[Madsen et al., 2001] Madsen, B. N., Pedersen, B. S., and Thomsen, H. E. (2001). Defining semantic relations for ontoquery. Technical report, Copenhagen Business School.

[Madsen et al., 2004] Madsen, B. N., Thomsen, H. E., and Vikner, C. (2004). Comparison of principles applying to domain specific versus general ontologies. In *Proceedings of OntoLex 2004: Ontologies and Lexical Resources in Distributed Environments*, pages 90–95.

[Madsen et al., 2005a] Madsen, B. N., Thomsen, H. E., and Vikner, C. (2005a). Multidimensionality in terminological concept modelling. In *Proceedings of TKE 2005*.

[Madsen et al., 2005b] Madsen, B. N., Thomsen, H. E., and Vikner, C. (2005b). Repræsentation af inddelingskriterier i caos 2. In *Proceedings of Nordterm 2005*.

[Marshman, 2002] Marshman, E. (2002). The cause relation in biopharmaceutical corpora: English and french patterns for knowledge extraction. Master's thesis, School of Translation and Interpretation, University of Ottawa.

[Marshman and L'Homme, 2006] Marshman, E. and L'Homme, M. C. (2006). *Modern Aproaches to Terminological Theories and Applications*, chapter Disambiguation of Lexical Markers of Cause and Effect, pages 261–285. Peter Lang.

[McCray, 2003] McCray, A. T. (2003). An upper-level ontology for the biomedical domain. *Comparative and Functional Genomics*, 4:80–84.

[McEnery and Wilson, 1996] McEnery, T. and Wilson, A. (1996). *Corpus Linguistics*. Edinburgh University Press.

[McMurry, 1998] McMurry, J. (1998). *Fundamentals of Organic Chemistry 4th*. Brooks/Cole Publishing Company.

[Melby, 1995] Melby, A. (1995). *The possibility of language*, volume 14 of *Benjamins Translation Library*. John Benjamins.

[Meyer, 1997] Meyer, I. (1997). Metaphorical internet terms: A conceptual and structural analysis. *Terminology*, 4(1).

[Meyer, 2000] Meyer, I. (2000). When terms move into our everyday lives: An overview of de-terminologization. *Terminology*, 6(1):111–138.

[Meyer, 2001] Meyer, I. (2001). Extracting knowledge-rich contexts for terminography. In Bourigault, D., Jacquemin, C., and L'Homme, M.-C., editors, *Recent Advances in Computational Terminology*, chapter 14, pages 279–302. John Benjamins.

[Michalski and Stepp, 1983] Michalski, R. S. and Stepp, R. (1983). Learning from observation: Conceptual clustering. *Machine Learning, An Artificial Intelligence Approach*, 2:331–363.

[Moffett, 2005] Moffett, M. A. (2005). Language, communication and the paradox of analysis: Some philosophical remarks on plato's cratylus. *Philosophiegeschichte und logische Analyse*, 8:57–68.

[Mukherjea and Sahay, 2006] Mukherjea, S. and Sahay, S. (2006). Discovering biomedical relations utilizing the world-wide web. In *Proceedings of Pacific Symposium on BioComputing (PSB 2006)*.

[Nenadic and Ananiadou, 2006] Nenadic, G. and Ananiadou, S. (2006). Mining semantically related terms from biomedical literature. *ACM Transactions on Asian Language Information Processing*, 5(1):22–43.

[Nuopponen, 1994] Nuopponen, A. (1994). Begreppssystem för terminologisk analys. In *Acta Wasaensia*, number 38 in Spraakvetenskap. Wasa University.

[Nuopponen, 2005] Nuopponen, A. (2005). Concept relations - an update of a concept relation classification. In *Proceedings of TKE 2005*.

[Oakes, 1998] Oakes, M. P. (1998). *Statistics for Corpus Linguistics*. Edinburgh University Press.

[Oeser and Picht, 1999] Oeser, E. and Picht, H. (1999). Terminologische wissenstechnik. *Fachsprachen - Languages for Special Purposes*, 2:2229–2237.

[Orilia and Varzi, 1998] Orilia, F. and Varzi, A. C. (1998). A note on analysis and circular definitions. *Grazer philosophische Studien*, 54:107–115.

[Pantel and Pennacchiotti, 2006] Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of ACL 2006*.

[Pantel and Ravichandran, 2004] Pantel, P. and Ravichandran, D. (2004). Automatically labeling semantic classes. In *Proceedings of HLT/NAACL 2004*.

[Patwardhan and Riloff, 2006] Patwardhan, S. and Riloff, E. (2006). Learning domain-specific information extraction patterns from the web. In *Proceedings of the ACL Workshop on Information Extraction Beyond the Document*.

[Pearson, 1998] Pearson, J. (1998). *Terms in Context*. John Benjamins.

[Penagos, 2004] Penagos, C. R. (2004). Metalinguistic information extraction for terminology. In *Proceedings of COMPUTERM 2004*.

[Popescu et al., 2004] Popescu, A.-M., Yates, A., and Etzioni, O. (2004). Class extraction from the world wide web. In *Proceedings of the AAAI 2004 Workshop on Adaptive Text Extraction and Mining*.

[Ravichandran and Hovy, 2002] Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of ACL 2002*, pages 41–47.

[Rosario and Hearst, 2004] Rosario, B. and Hearst, M. (2004). Classifying semantic relations in bioscience text. In *Proceedings of ACL 2004*.

[Rosch, 1973] Rosch, E. (1973). Natural categories. *Cognitive Psychology*, 4:328–350.

[Sabou, 2005] Sabou, M. (2005). *Ontology Learning from Text: Methods, Applications and Evaluation*, chapter Learning web service ontologies: an automatic extraction method and its evaluation. IOS Press.

[Sager, 1990] Sager, J. C. (1990). *A practical course in terminology processing*. John Benjamins.

[Sharoff, 2006] Sharoff, S. (2006). Open-source corpora: Using the net to fish for linguistic data. *International Journal of Coprus Linguistics*, 11(4):435–462.

[Sinclair, 1991] Sinclair, J. (1991). *Corpus Concordance Collocation*. Oxford University Press.

[Smith et al., 2004] Smith, B., Kumar, A., and Schulze-Kremer, S. (2004). Revising the umls semantic network. *Medinfo 2004*.

[Sowa, 2000] Sowa, J. F. (2000). *Knowedge Representation - Logical, Philosophical, and Computational Foundations*. Brooks/Cole - Thomson Learning.

[Spasic et al., 2005] Spasic, I., Ananiadou, S., McNaught, J., and Kumar, A. (2005). Text miing and ontologies in biomedicine: making sense of raw text. *Briefings in Bioinformatics*, 6(3):239–251.

[Stevens et al., 2004] Stevens, R., Wroe, C., Lord, P., and Goble, C. (2004). *Handbook on Ontologies*, chapter Ontologies in Bioinformatics, pages 635–657. Springer.

[Suonuuti, 1997] Suonuuti, H. (1997). Guide to terminology. The Finnish Centre for Technical Terminology.

[Temmerman, 2000] Temmerman, R. (2000). *Towards New Ways of Terminology Description*. John Benjamins.

[Toft, 2000] Toft, B. (2000). *I Terminologins tjaenst - Festskrift foer Heribert Picht paa 60-aarsdagen*, chapter Terminologi og vidensteknik: En lykkelig alliance? University of Vaasa.

[Tognini-Bonelli, 2001] Tognini-Bonelli, E. (2001). *Corpus linguistics at work*. John Benjamins.

[Turney, 2006] Turney, P. (2006). Expressing implicit semantic relations without supervision. In *Proceedings of COLING-ACL 2006*.

[Vintar et al., 2002] Vintar, S., Buitelaar, P., and Ripplinger, B. (2002). An efficient and flexible format for linguistic and semantic annotation. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*.

[Vintar et al., 2003] Vintar, S., Sonntag, D., and Buitelaar, P. (2003). Evaluating context features for medical relation mining. In *Proceedings of the ECML/PKDD Workshop on Data Mining and Text Mining for Bioinformatics*.

[Widdowson, 1974] Widdowson, H. G. (1974). Literary and scientific uses of english. *English Language Teaching*, 4:282–292.

[Wierzbicka, 1992] Wierzbicka, A. (1992). *The search for universal semantic primitives*, pages 215–242. John Benjamins.

[Wierzbicka, 1995] Wierzbicka, A. (1995). Universal semantic primitives as a basis for lexical semantics. *Folia Linguistica*, 29(1-2):149–169.

[Woods, 1997] Woods, W. A. (1997). Conceptual indexing: a better way to organize knowledge. Technical report, Sun Microsystems Laboratories.

[Wüster, 1991] Wüster, E. (1979 (1991)). *Einführung in die allgemeine Terminologilehre und terminologische Lexicographie*. Romanistischer Verlag.

[Yu and Agichtein, 2003] Yu, H. and Agichtein, E. (2003). Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, 19:340–349.

[Yu et al., 2002] Yu, H., Hatzivassiloglou, V., Friedman, C., Rzhetsky, A., and Wilbur, W. J. (2002). Automatic extraction of gene and protein synonyms from medline and journal articles. In *Proceedings of the AMIA Symposium 2002*, pages 919–923.