Copenhagen Business School
Copenhagen, Spring 2022

Deparadoxifying Strategic Decisions: An Integrated Approach Utilizing Machine Learning and Natural Language Processing

Analyzing strategic decision making within the UN security council by applying the BERT model and inferential statistics

Birgitte Ramm Bergo and Elias Bjørne-Larsen

Student numbers: 141271, 14882, contract no: 22745

Supervisor: Steffen Blaschke

Character count: 229,434

Normal page count: 119

Master thesis, Business Administration and Data Science

COPENHAGEN BUSINESS SCHOOL

Acknowledgements

We would like to thank our supervisor, Steffen Blaschke, for his excellent advice and guidance as we explored the unfamiliar territory of paradoxes. We express our sincerest gratitude for the level of support he has shown us throughout the project.

We would also like to thank Morten Lantow, our mentor at EY Denmark, which saw the exciting potential in this project and lent us his wisdom time after time. His council was greatly appreciated.

Our friends and family has shown us tremendous support throughout the writing of this thesis, for which we are ever grateful.

Copenhagen Business School

Copenhagen, 16 May 2022

Birgitte Ramm Bergo

Elias Bjørne-Larsen

Abstract

The purpose of this paper is to answer the call for empirical research on deparadoxification by demonstrating a new approach which utilize the rapid technological developments that has taken place since Luhmann introduced the concept of deparadoxificaiton. In doing so, the paper seeks to combine two academic fields which not yet have been connected, namely deparadoxification and machine learning. The paper demonstrates a feature-based approach with BERT and random forest, to classify the paragraphs in the United Nations Security (UNSC) meeting minutes into deparadoxification strategies. The contextual model will be compared with a non-contextual model (using TF-IDF) to investigate whether deparadoxification is context dependent or not. Due to the lack of existing labeled data, the authors constructs their own training dataset through iterative manual labeling using active learning with least confidence sampling. The model will be used to uncover the distribution of deparadoxification strategies and provide classified data for the regression analysis to investigate whether the strategies affect resolution voting outcome. The model reached 0.53 accuracy and 0.44 F1 macro after five iterations of labeling, followed by hyperparameter tuning. The non-contextual model reached 0.47 accuracy and 0.35 F1 macro. Both models outperformed ZeroR and the uniform dummy classifier (UDC) by a large margin. Both the labeled and predicted distribution suggested that the strategies does not follow a uniform distribution, but are rather imbalanced. The regression analysis suggested that the strategies (only based on occurrences) does not explain any of the variation in the voting outcome. We argue that this is due to the majority of resolution votes being unanimous. We believe, however, that as this merging of fields gets more attention, larger datasets, sufficient in size to train complex models, will be made available, which might lead to different results. While the regression analysis did not show significant results, the fact that both models outperformed ZeroR and UDC proves that they were in fact able to pick up on a pattern, demonstrating that it is possible to detect and measure deparadoxification through machine learning.

Source code is available at https://github.com/birgitterb/Master_thesis.git

Keywords – Deparadoxification, Machine Learning, Natural Language Processing, BERT, Active Learning, United Nations, Organizational Decision Making

Contents

List of Figures

List of Tables

Acronyms

AL active learning.

BERT Bidirectional Encoder Representations from Transformers.

BERT-RF Bidirectional Encoder Representations from Transformers - Random Forest.

CRISP-DM Cross Industry Standard Process for Data Mining.

CV coefficient of variation.

DBN dynamic Bayesian network.

FN false negative.

FP false positive.

GRU gated recurrent unit.

IDF inverse document frequency.

LC least confidence.

LSTM long short-term memory.

ML machine learning.

MLM masked language modelling.

NSP next sentence prediction.

OLS ordinary least square.

OOP object-oriented programming.

RF random forest.

RNN recurrent neural network.

TF term frequency.

TF-IDF term frequency-inverse document frequency.

TFIDF-RF term frequency-inverse document frequency - random forest.

TN true negative.

TP true positive.

TSS theory of social systems.

UDC uniform dummy classifier.

UNSC United Nations Security Council.

1 Introduction

For thousands of years, social systems have been an integral part of humanity. To tackle the continuous struggle for adaptation to the ever-changing environments that we find ourselves in, humanity has sought to master coordination and cooperation as means of survival. In modern times, this quest has taken the shape of longing for mastering one of modern society's foundational pillars, the organization. While the struggle for individual survival has been overcome, adaptation has become crucial for organizational survival and growth. The rapid changes in the technical, cultural, political, and economical environments necessitate the desire for organizational change and adaptation. Within this domain, decision making naturally becomes the center of attention. Our understanding of organizational decision making bears implications for how our modern society function. Modern technologies allow for new approaches to research how we humans interact and make decisions within organizations. However, the academic application of modern technologies in analyzing organizational decision making does not match the unprecedented speed of which new technologies emerge. To explore this sea of potential discoveries, this paper targets a field which until now has been neglected by domains outside of organizational and managerial studies: Deparadoxification. As Heinz von Foerster (2003) famously said, "only those questions that are in principle undecidable, we can decide." Following his lead, Luhmann (2006) offers three deparadoxification strategies: temporal, factual, and social deparadoxification.

While Luhmann's strategies have received some anecdotal empirical analysis (e.g., Andersen (2003)), there is to date no attempt to classify large data sets of decisions into any of these three strategies. Given the advancements in language representation, specifically embeddings, there has never been better opportunities to answer the call for empirical studies on deparadoxification, as expressed by Andersen (2003) and Knudsen (2006), through an applied statistical approach, namely machine learning (ML). By using a state-of-the-art language representation model, BERT, the authors wish to investigate whether and how Luhmann's deparadoxification strategies manifests themselves in the real world. The non-existence of similar studies poses both exciting potentials for contribution and obstacles that need to be overcome, the main one being the lack of data. Because

there exists no labeled dataset on deparadoxification strategies, the authors of this paper are constructing their own. In doing so, the paper demonstrates an approach to merging previously untouched organizational fields with supervised ML. The United Nations Security Council (UNSC) was chosen as the case study due to its highly structured meeting minutes, frequent discussions and decision making, and that the voting outcome of resolutions can be used to investigate the impact of deparadoxificaiton strategies.

1.1 Research Questions

We set out to remedy the shortcoming of empirical research on deparadoxificaiton by using BERT embeddings extracted from the UNSC meetings minutes to train a supervised ML model, random forest (RF), to classify the meeting minutes paragraphs into deparadoxification strategies. In order to do so, a training set will be created by manually labeling paragraphs, using a set of labeling instructions based on our conceptual framework. Due to the resource intensive labor of labeling paragraphs, which might be a barrier for researchers to approach deparadoxification with supervised ML, active learning (AL) will be used to iteratively increase the size of the training set, with the purpose of investigating approaches to make the labeling process more efficient. When the training set is complete, the model will be compared to a non-contextual model to explore whether deparadoxification strategies are contextually dependent or not. Furthermore, the distribution of strategies within the UNSC will be predicted and discussed, and finally, a regression analysis will be performed to determine whether deparadoxification strategies affect voting outcome or not. The purpose of the paper is to contribute to answering the call for empirical studies on deparadoxification. Based on this, we seek to answering the following four research questions (RQs):

RQ1: To what extent does the chosen NLP model respond to active learning when classifying deparadoxification strategies?

RQ2: Do contextual embeddings outperform non-contextual embeddings and do they respond differently to hyperparameter tuning?

With an NLP model for the classification of deparadoxification strategies in place, we continue to find answers specifically for deparadoxification strategies used in the political communication of the UNSC:

RQ3: Is there a uniform or otherwise distribution of deparadoxification strategies and what could be the underlying causes for this distribution?

RQ4: Does the use of any of the three strategies affect the voting outcome of resolutions?

In addition to answering these questions, we will suggest and discuss possible underlying causes that seem plausible. By answering the research questions, we hope to demonstrate a possible approach to merging the fields of deparadoxification and ML, hopefully inspiring future research.

1.1.1 Topic Delimitation

In relation to RQ1, the paper does not seek to compare multiple active learning approaches to conclude which one works best. Only least confidence (LC) sampling will be used, after the initial labeling using random sampling. The paper goes on to discuss other types of sampling techniques for active learning that might yield different results. Regarding RQ2, the paper will only compare two models; one using contextual embeddings and the other non-contextual embeddings. In doing so, we suggest that the results have implications for contextual vs. non-contextual embeddings in general when it comes to detecting deparadoxification, but the results are not meant to be conclusive for all embedding models. For RQ3, the main goal of the researchers is to uncover the distribution, while the underlying causes are of secondary priority. The potential underlying causes will not be analyzed using inferential statistics, but rather suggested and discussed based on the conceptual framework, case, and descriptive statistics. To investigate whether the use of any of the strategies affect voting outcome, a regression analysis will be applied. The results of the regression analysis only suggest whether the independent variables explain variation in the dependent variable. It does not take into account the timeline, or how the different strategies are connected. Hence, RQ4 is not meant to yield a decisive conclusion for whether deparadoxification has affect on decision making, but rather whether the occurrences of different strategies, as classified by our model, independent of time and each other, can explain variation in the voting outcome. The data used to answer the RQs are limited to the period of 1995 to 2020. Furthermore, given the intricacies of deparadoxification strategies, it is important to note that the paper only investiagte the *occurences* of the strategies, as opposed to how they are related to time or each other.

1.2 Thesis Structure

This chapter aims to outline the paper to assist the reader by providing a brief description of the topics covered in the following chapters.

Chapter 2 - Conceptual Framework: Explains central theoretical concepts of existing literature within social systems theory, organizational decision-making, and deparadoxification. This chapter lays the foundation of the theory applied in this paper by constructing an understanding of decision making, how paradoxes appear within decision-making and how to avoid paralyzation of decision-making.

Chapter 3 - Technical Background: The chapter presents essential underlying technical concepts of the thesis. This includes the concepts of embeddings, TF-IDF, BERT, RF, evaluation metrics, and AL. The purpose is to give the reader a conceptual understanding of how the applied models work and a technical introduction to the most vital techniques.

Chapter 4 - The United Nation Security Council: Describes the United Nations Security Council as an organization, including its purpose and how they make decisions. In addition, the chapter will further elaborate on the reason for choosing this specific organization for analyzing deparadoxification through ML.

Chapter 5 - Methodology: Presents and explains the methods and techniques applied in the thesis, from data understanding and labeling, to preprocessing, modelling, hyperparameter tuning, active learning, and regression analysis.

Chapter 6 - Results: Presents the results and performance of the models presented in Chapter 5. Specifically, AL iterations, hyperparameter tuning, model evaluation, the predicted distribution, regression analysis, and word clouds.

Chapter 7 - Finding and Discussion: Discusses the findings from Chapter 6 and correspondingly answer the research questions before presenting the limitations and future research.

Chapter 8 - Conclusion: Summarizes the result and findings and then concludes on the presented research questions.

2 Conceptual Framework

As this thesis is an interdisciplinary study of social science and data science, focusing on deparadoxification and its strategies combined with a ML approach, it is necessary to first understand the phenomenon of interest, deparadoxification. Therefore, this chapter will present and explain the theory of deparadoxification, in the context of social systems, organizational decision-making, and previous quantitative research.

2.1 Social Systems Theory

Andersen's (2003) description of decisions as a communication-theoretical systems theory phenomenon is predominantly based on Niklas Luhmann's theory of social systems (TSS) (Luhmann, 1995). According to Luhmann, a social system is a system that can reproduce itself through communications, and Luhmann categorizes three types of social systems: society, organization, and interaction (Seidl and Becker, 2006a). The theory has its roots in general systems theory, a transdisciplinary field of study seeking to explain behavior found in complex and organized systems (Whitchurch and Constantine, 1993). This is apparent in how Luhmann emphasizes that any social operation is part of a system (Luhmann, 1995).Nassehi (2005) explains the term system as referring to "a holistic structure that controls all constituent phenomena, with each and every particular in subordination to the general structure of the encompassing system" (p. 179-180). TSS stresses that social systems construct their problems in addition to the related functional solutions utilizing their resources (Nassehi, 2005).

There are primarily two concepts of TSS that are essential to understanding Andersen's (2003) description of decisions and deparadoxification, which is communication and observation. Communication is crucial to understand deparadoxification, while observation is, in essence, the core of TSS, and important in order to understand communication (Andersen, 2003). Observation refers to Luhmann's perspective on Spencer-Brown's (1969) theory about observations as operations of differentiation and his calculus of form. As described by Seidl and Becker (2006a): "An observation is any type of operation that makes a distinction in order to indicate either side of the distinction" (p. 408).

Distinctions can also be looked at as a selection between options. By the same token, an observation can be considered as an indication within the scope of a distinction. A distinction has two sides: An inner and outer side, while the observer does not see the space from which the observation is made. The inner side is marked, and the outer side is unmarked. When something is indicated, e.g., slow bureaucracy, the observation of slow bureaucracy cannot observe the space from which slow bureaucracy was observed. The unity of this distinction is what constitutes the form of observation. Furthermore, the unity is what constitutes the observation's blind spot.

According to Luhmann (1995), communication is a unity between selecting information, utterance, and understanding. In the classical notion of communication, information refers to what is being communicated. In other words, it refers to moving information from a transmitter to a recipient. Utterance refers to how the information is being communicated and why the communication takes place. Understanding is how the information is supposed to be understood, and it is the distinction between utterance and information (Luhmann, 2006). Understanding can also be viewed at as how following communication might link up with previous communication. This implies that communication happens retrospectively, because there must be a reply to define it as communication. This excludes, e.g., monologues from being communicated. It also implies many possible connections to all communication, allowing for different interpretations and interactions even though the information stays the same. This entails that the connecting communication is the deciding force for whether the communication takes place and how it takes place. As Andersen (2003) explains, understanding is the selecting of a connection from the pool of possible connections. Another way of looking at communication is as a flow of selection between these three elements; information, utterance, and understanding, constantly linking itself to prior communication in a retrospective manner (Andersen, 2003).

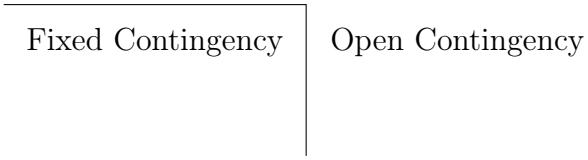
Based on communication's threefold unity and the flow of selection, decisions are not first made and then communicated, but the decision itself is communication. There are two types of communication: Ordinary communication, which communicates already

selected content, and decisions, which communicates that a selection has been made, implying that other alternatives were not selected. This further implies that even though decisions are communication, it is still possible to communicate about decisions without that communication is a decision in itself (Schoeneborn, 2011).

2.2 Organizational decision-making

Organizations are driven by a constant need to carry out selections in the form of decisions (Luhmann, 1988; Schoeneborn, 2011). In addition, the organizations serve as both the producer and product of decision necessities. This distinguishes an organization from other social systems, such as interactions (Seidl and Becker, 2006b), when past decisions become the premise for future decisions. To observe decisions as communicative observations, rather than individual choices, Andersen (2003) suggests observing decisions as a form of observation. This entails that a decision is "not an object that one looks at but a specific distinction that one looks through" (p. 7).

Andersen (2003) proposes that the organizational form of communication in which decisions take place in the form of all communication, referring to both ordinary communication and decisions. Therefore, decisions as a form of communication involve consideration of social expectations. These expectations are Luhmann's conceptualization of social structures in social systems (Seidl and Becker, 2006a). This entails that expectations are, in reality, the communication of expectations that occur based on the situation. If an expectation is met, it is confirmed and will likely continue to function as a social structure. If not, the expectation might be changed. All decisions are solely directed at the social expectations inherent to the organization. Therefore, a decision can be defined as communication that involves consideration of social expectations. With this view, decisions only create and fill existing expectations among the organization members and do not determine the future. By filling expectations about what will happen in the organization, expectations of future decisions arise, which is why decisions create further social expectations and following decisions.



Decision

Figure 2.1: A form of decision

Decisions generate further social expectations and decisions by installing a boundary in the communication, separating it into "before" and "after" a decision is made. The "before" is not established until after the decision, as a decision has to be made to claim that there ever was a "before". Therefore, the "before" becomes the point of open contingency. Contingency is the status of propositions that are neither necessarily true nor false. During open contingency, there are multiple different solutions available. This contingency becomes fixed after reaching a decision, implying that only one alternative was selected while the others were not. Every organizational decision shapes this distinction between open and fixed contingency concerning the social expectations (Andersen, 2003).

The distinction between open and fixed contingency can be illustrated using notation from Spencer-Brown (1969), as seen in Figure 2.1. It is the unity of this distinction that a decision represents. By looking through the distinction, the contingency appears to be both open and fixed simultaneously. This implies that a decision fulfills social expectations and generates insecurity by communicating that other alternatives could have been selected. Given that a decision is the unity of that which it divides, a paradox is installed.

2.3 Deparadoxification

2.3.1 Introduction to Paradox

In the last couple of decades of organizational and managerial studies, the recognition of paradoxes has increased (Cunha and Putnam, 2019). Research in innovation, change, communication, and rhetoric was sparked by paradoxical discussions in the late 1980s (Smith & Lewis, 2011). According to Putnam et al. (2016) review, over 850 publications

were identified to focus on related topics of organizational paradox. A possible explanation of the emerging research interest could be the change in corporate environments. The corporate environment have become more global, dynamic, and competitive, resulting many of paradoxicalities (Smith and Lewis, 2011). Organizational and managerial studies explain the reality of organizational environments by definitions, but it is assumed to be highly complex, reflecting the reality. Researchers have expressed a need to confront present paradoxes found in organizations to understand and explore organizations (Braathen, 2016). Earlier research has shown the difficulties of transferring formalization, e.g., applied in mathematics, to other research fields, e.g., organizational and sociological studies, due to the need of uncover social conditions and formal structures to handle these as values in a formalized format (Luhmann, 1995). Paradox studies examine how organizations might simultaneously meet opposing needs. A paradox viewpoint suggests that long-term sustainability necessitates ongoing attempts to fulfill numerous different needs (Smith & Lewis, 2011).

A paradox exists when the criteria for an operation's possibility are also the conditions for its impossibility (Seidl et al., 2021). In Luhmann's organizational theory, the paradox of decisions is central, as if the decision is communicating real alternatives for the decision, the decision made will not be recognized as it has been decided (Seidl et al., 2021). A formal definition is that paradoxes are re-entry of a distinction into itself, meaning that outside of the distinction is entering its inside (Seidl et al., 2021). Consequently, the inside of the distinction is both inside and outside of the distinction. Due to this fluctuation, there is no apparent connection for future operations, leading to paralysis. An example of a paradox within the notion of Luhmann's work is form. Form has no reference other than itself, meaning that form contains an inherent paradox (Andersen, 2003). In order to unfold the paradox, the observer's blind spot needs to be transferred to a "less disturbing place" (Luhmann, 2006), which will be explained in Chapter 2.3.3.

The term paradox is derived from the Greek word paradoxos, which means beyond or outside (para) and to think (dokein) (Braathen, 2016). In other words, a paradox is a statement that is opposed or contradictory to common sense and yet is true. An example

from ancient Greek is the liar paradox presented by Chrysippus (Braathen, 2016). The paradox is as follows: "A Cretan sail to Greece and says to some Greek men that All Cretans are liars.". However, this is a paradox as the Cretan says that all Cretans are liars, but then the question is, is he lying or telling the truth? Common assumptions are 1) that a liar always tells lies and the antonym of liar, a truth-teller, always tells the truth, 2) If the Cretan is not a liar, then the Cretan would be a truth-teller. The third assumption is that the Cretan is not the only Cretan. Therefore, this is a paradox as it seems impossible to solve if the statement is true or not, and if he is lying, then the statement would not be accurate. Either way, the statement cannot be confirmed with the known assumptions, and it is the same with decisions. Every decision is, in fact, paradoxical (Luhmann, 1995). This will be elaborated on in the next section.

2.3.2 The Paradoxy of Decisions

According to Luhmann (2006), a decision is a result of attribution, and if an organization could be observed as a network of decisions, then decision-making would be what differentiated organizations from each other. Furthermore, as Luhmann stated, all decisions are paradoxical. The form of decisions is self-defined, meaning that decisions must turn inward and only devote themselves to themselves, which is the paradox of decisions. Andersen (2003) is using the term *paradoxy of decisions* to describe the inherently paradoxical nature of decisions.

As an extension to the paradoxy of decisions, Andersen (2003) presents a threefold paradoxy of decisions, meaning that decisions are paradoxical in three different ways. Firstly, Andersen (2003) states that only fundamentally undecidable questions can be resolved, meaning that decisions cannot be reached as they will always have the effect of creating potentially new decisions. Therefore, forced freedom is a paradox. Luhmann uses the Heinz von Foerster (1992, p. 14) quote to explain how decisions are, in fact, a paradox as they are undecidable, "Only those questions that are in principle undecidable, we can decide" (Knudsen, 2006). If the question can be resolved through a calculation, it is not a decision. As Knudsen (2006) describes, for a decision to be valid, the options presented need to be assumed to be of the same weight; if not, the options would not have been recognized as valid options to choose. Therefore,

the given alternatives are equally valid; there are no better or worse alternatives – otherwise, these would not be authentic alternatives. If the alternatives were of different value (in which case they would not be real alternatives), there would be no need to decide between them anymore, as the decision situation would have already been decided.

Secondly, as noted earlier, decisions fulfill social expectations of the future, but the decisions are always reached retrospectively. Therefore, it is impossible to determine whether: 1) the decision was resolved, 2) the expectations were fulfilled, or 3) the contingency was fixed or not. The argument is that the following potential decision seizes the essence of the decision. In other words, decisions continually determine whether previous interactions may be considered decisions and can be utilized as a foundation for future decisions(Andersen, 2003). To summarize, decisions create new decisions. It is a paradox as the decision is not valid before it is confirmed as a decision hypothesis, making decisions determined by other decisions and not by itself as an individual decision.

The third part of the paradox is that deciding whether a decision was made is a decision in itself, meaning that it may not be evident even in retrospect. The exception is a new decision deciding if the previous decision was a decision. Therefore, a decision must decide whether it is a decision, which is basing the definition of a decision on a paradox. For an organization, this means that the organization must make decisions and decide what a decision within the organization is. Therefore, there are several measures that an organization needs to address when deciding on a decision, e.g., organization and context, and due to the paradox, these measures can only be partially fixed, which is visualized as Figure 2.2 (Andersen, 2003).

2.3.3 Deparadoxification Strategies

As earlier stated, decisions are a communicative form, and according to Andersen (2003), any communication introduces a degree of contradiction into the conversation. When a communication encounters its contradiction, it becomes paralyzed by the realization of it being unable to decide. Therefore, the decisions need to avoid a collision with their paradox and avoid being perceived as boundless. In order to do so, deparadoxification

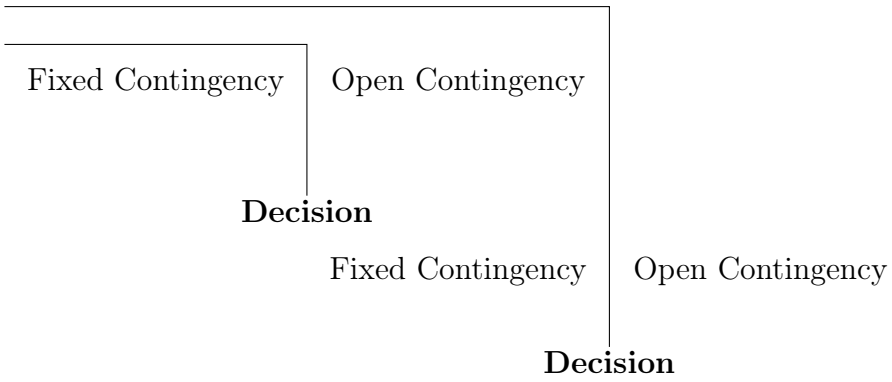


Figure 2.2: A form of re-entry, adopted by Andersen (2003)

must be applied. Deparadoxification is a phenomenon described by several researchers, under different but similar terms, e.g., deparadoxization, and de-paradoxify (Seidl and Becker, 2006a; Knudsen, 2006; Sohn, 2021).

The practicality of Luhmann's TSS for empirical research was demonstrated by Knudsen (2006) by examining modernization processes whereby decision-making within the Danish public sector, explicitly analyzing Frederiksborg County Health Authority. Following the Frederiksborg County Health Authority for approximately 20 years, Knudsen (2006) showed that the Frederiksborg County Health Authority emerged within referencing to itself and at the same time handling the paradoxy of decisions by deparadoxification. An empirical example, the county created two hospital plans in 1980 and another one in 1997; both plans communicate the decisions made and legitimize the decisions by self-referencing, e.g., §11, subsection in law number 324 and the county health committee (Knudsen, 2006). From 1980 to 1997, Frederiksborg County Health Authority effectively defined itself as an organization by creating connected decisions and decision premises (Knudsen, 2006). The establishment of decision premises is part of the deparadoxification strategies. The County Health Authority displaced contingency related to all the mentioned deparadoxification strategies; temporal, social, and factual.

Deparadoxification is the term for moving the paradoxy of a decision out of sight to prevent paralyzing the decision-making (Seidl, 2006). There are three distinct types for strategies of deparadoxification of decisions: temporal, social, and factual deparadoxification (Luhmann, 2006; Andersen, 2003). The strategies are related to the dimensions of meaning in

communication, and they lay the foundation for avoiding the paralyzation of decision-making. Whether and how these theoretical concepts manifest themselves in practice will be elaborated on in chapter 7. The following paragraphs will explain the three types of deparadoxification, termed *deparadoxification strategies.*

2.3.3.1 Temporal

A temporal deparadoxification strategy can be perceived as a reaction to the urgency of the moment (Andersen, 2013). Temporal deparadoxification refers to a strategy related to a time momentum, and these tensions are created by communicating expectations within the time horizon and the field of experience. As Derrida (1992) states, to encounter the paradoxy of decision, the decision must appear as the decision is required immediately, "right away", to be valid. In addition, it must furnish itself with infinite information and complete knowledge of conditions, rules, or hypothetical imperatives that could justify it. As Derrida argues, despite how much time the decision would need to meet the condition, the moment of decision will always be finite of urgency (Derrida, 1992). The strategy makes it appear that the decision-making cannot be postponed any further, and therefore a decision has been made without immobilization of the threefold paradox (Andersen, 2013). Typical phrases associated with temporal are "time is ripe" and "the time has come", referring to the moment of making the decisions and urging for a decision to be made.

As Andersen (2003) describes, the undecidability of the decision is minimized by splitting the decision, meaning that parts of the decision can be mature for a decision. An example of this is when considering significant decisions, e.g., building a bridge or a tunnel, which may be too significant to decide. The temporal deparadoxification strategy is constructing more feasible decisions by splitting the main decision into smaller and more feasible ones, e.g., location, design, or technical choices. Consequently, the decision premises are changing over time, and appear more manageable, but also creating future sequences of decisions, like in a construction plan; future scenarios are created once the decision of building a bridge is divided. By dividing the decision, future scenarios are created, which is part of temporal deparadoxification. The intention of creating the future is to change "what can still be changed" to "what cannot be changed", as it would make

the decision appear as a choice made at present. Andersen(2003) also argues that some future scenarios are staged as more agreeable than others with different decision premises, making it appear as if some decisions are appealing long term despite what could be presented as a short-term advantage.

Temporal deparadoxification strategy has been identified by Knudsen (2006) in practice, and during an interview in 1999, an officer manager that was making a draft expressed the following:

> "The closer we get to the finishing line the less clear the papers become in order for people to connect to them. It is a well-known situation; it is often like that. It doesn't make it any easier for the rest of us when we have to follow up on the decisions with the unions – for what exactly have they agreed to?" (Knudsen 2006,p.121-122).

The quote describes what happens in decision-making when a decision emerges. As the office manager expressed, the contingency concerning the decision is masked. The contingency is shifted to the future, making it possible to perform decisions. The construction of moving the contingency for the future is typical for temporal deparadoxification.

2.3.3.2 Social

When a decision appears as if it has already been made and the only thing necessary to complete it is formal requirements, it is called social deparadoxification (Andersen, 2003). Social deparadoxification is about using central players for a decision and assigning these players traits, e.g., power, interests, and actions, and communicating that the decision has already been made. Applying this technique avoids the paradox of a decision as it seems that the decision is already decided. When a decision appears as if it has already been made and the only thing necessary to complete it is formal requirements, it is called social deparadoxification (Andersen, 2003). The social deparadoxification differentiates itself as a strategy, as the expectation of the decision is addressed towards the communicator of «them», «me», and «us». As Andersen (2003) states, no social space can be constructed

with “us” without having an existence of “them” and that the social space is created every time this tension occurs.

The decision premises that create a paradox are solved when applying social deparadoxification. For example, if it appears that the CEO is following a strategy of deciding on the alternatives with the lowest costs. As it seems that the CEO follows this strategy, the decision on the different options is already decided before it takes place (Seidl,2021). The decision, then, is no more or less than a resolution of the parties to accept or lead towards the oblivious result, in other words, a formalization. In other words, when deciding upon a decision what is in the CEO’s strategy is constructing that the decision appears to be already taken.

An empirical example of social deparadoxification strategy was identified in the works of Knudsen (2006), a decision-proposal regarding a general plan for the county. A decision-proposal were sent for hearings within the organization, and as Knudsen (2006) argues, hearings can be recognized as testing virtual decisions, which involve committees, stakeholders, and related groups. After hearings, the proposal is adjusted, and a report of a hearing response contains a significant number of pages from more than 80 respondents. Then the decision is finally confirmed. The contingency is displaced by assigning the decision to a person or an institution. In this case, it was achieved by connecting the decision to the institution through hearings, a social actor.

2.3.3.3 Factual

Factual as a deparadoxification strategy is best described supplying the decision-maker options to choose from or between different options (Luhmann, 2006). Factual deparadoxification moves the paradoxy of a decision out of sight, by creating alternatives for the decision maker to choose from (Andersen, 2003). When using factual deparadoxification, the decisions appear as results of the circumstances, enabling the decision-maker to decide upon a decision. The alternatives in a decision-making situation are also a decision, and the reason for this is that it decides on what grounds the decision will be decided on. The factual deparadoxification strategy is typical as decisions are

usually perceived as choices between alternatives. As Andersen (2003) describes, by referring to the circumstances of the decision, e.g., “environment”, “market”, “globalization” or “economy”, these circumstances are shaped like a “someone” or “something” that determines whether a decision should be executed. Part of the strategy is the decision to communicate themes and objects, whereas themes and objects are defined as the distinction of “being-one-thing-and-not-another” (Andersen, 2003).

Factual deparadoxification can also be perceived as creating decision rules is one of the ways of moving the paradoxy of decision out of the situation, which can allow different ordered alternatives, e.g., "choose the option that has the lowest financial costs" or "choose the option that is the most pragmatic one" (Seidl et al., 2021). The paradoxy of decision is still present, but by choosing a selection of decision rules, the decision is not paralyzed as the paradox is shifted to which decision rule the decision-maker should decide. Therefore, the situation is undecidable, and by moving the paradox out of the situation, the delay in determining the paradox constructs that it will never be resolved.

An example is when a CEO states that a new business strategy has been decided, the CEO also needs to communicate the existence of other alternatives. Otherwise, the decision would not exist as there would be nothing to decide (Seidl et al., 2021). The more the other alternatives are communicated as valid options, the more the audience will question the decision. Moreover, constructing that the decision may be undecided, and for this reason, the CEO is facing a paradox where it is necessary to convince the audience that there was a definite decision with alternatives, but that the other options should not be considered as the decision has been made (Seidl et al., 2021). The mentioned example is hypothetical and not empirically shown, but an empirical example of factual deparadoxification is presented in the next paragraph of Knudsen (2006).

A real example of factual deparadoxification, described by Knudsen (2006), is proposal by the Danish health authorities which states that there should be independent management for the entire health service. The memorandum lists more than 30 purposes and positive effects of the suggestion, which implies contingency. There are many purposes, meaning

that the connectivity to the decision proposal cannot be taken as decided (Knudsen, 2006). However, the decision's contingency is shifted to the purposes, and the purposes' contingency is shifted between the purposes (Knudsen, 2006). Consequently, the reader cannot examine the substantial number of chains.

2.4 Quantitative Research Context

The Luhmannian way of conceptualizing deparadoxification is acknowledged and recognized by researchers within managerial and organizational studies (Andersen, 2003; Knudsen, 2006; Smith& Lewis, 2011). Nevertheless, there is still a demand for further research within this field. As both Andersen (2003) and Knudsen (2006) have recognized, there is a need for empirical studies, such as quantitative analyses (Seidl et al., 2021). Moreover, as Andersen (2003) argues, there are different ways organizations can apply deparadoxification strategies within decision-making, which makes further empirical examinations necessary.

Cohen et al. (1972) translated organizational decision-making to a simulation model, known as the garbage can model, and demonstrated possible applications of using such models. Even though the garbage can model is ineffectively solving the tasks it was designed for, it showed how decision-making theory could be used in practice with a quantitative approach (Cohen et al., 1972). Furthermore, the study provided an understanding that organizational design and decision-making can recognize the existence of the garbage can model. Finally, the model exemplified how organizational theory can use data science. Ever since the model was developed in FORTRAN during the 1970s, there has been an unprecedented development in the field of data science, implying the potential for better models combining data science and organizational studies. This remarkable technological development, combined with the call for empirical studies of deparadoxification to fill the research gap, motivates the quantitative approach chosen by this paper: applying modern machine learning algorithms to detect and analyze deparadoxification strategies.

2.5 Conceptual Framework Conclusion

This chapter has presented and explained the conceptual framework for this research paper. The theory of deparadoxification relies on Luhmann's theory of social systems. As Luhmann states, a social system is a system that can reproduce itself, and an organization is a type of social system which reproduces itself by decisions. According to Luhmann, organizations are based on decisions, meaning that decisions are moving the organization onward. Decisions are a type of communication that inherent a boundary in its communication, separating before and after a decision. Before a decision is made, there are multiple alternatives, meaning that it is an open contingency, while after a decision is made has a fixed contingency. The "before" of a decision is not confirmed before the "after" of the decision. In other words, as open and fixed contingency occurs simultaneously, it is a paradox. Therefore decisions are inherently a paradox, and as paradoxes cannot be solved, the paradox has to be moved to "out of sight" to avoid paralyzation of decision-making.

The concept of shifting the paradox is termed deparadoxification, and there are three specific strategies: temporal, social, and factual deparadoxification. Andersen (2003) further developed these strategies established by Luhmann (1993). Temporal deparadoxification is regarded as a reaction to the immediacy of the situation (Andersen, 2003). Social deparadoxification masks the paradox of the decision by providing alternatives for the decision-maker to consider (Andersen, 2003). Another way of shifting the paradox is by constructing alternatives for the decision-maker to choose from (Andersen, 2003). Researchers within the area, e.g., Andersen (2003) and Knudsen (2006), call for more empirical research within the research area of deparadoxification. Research conducted by Cohen et al. (1972) applied data science techniques to decision-making. The research results were ineffective, but as data science has advanced rapidly over the years, modeling decision-making could be more successful. This is what constitutes the main purpose of this paper: To fill the research gap of empirical studies on deparadoxification. The authors have chosen a approach which to their knowledge has not been tried before. Namely, combining the field of machine learning and deparadoxification, with the goal of improving our understanding of how organizations make decisions and how deparadoxificaiton can

be analyzed following a quantitative approach. While this chapter has explained the deparadoxification component of the paper, the next chapter, Technical Background (Chapter 3), will explain the machine learning component.

3 Technical Background

Given the advancements in language representation, especially within embeddings, there have never been better opportunities to answer the call for empirical studies on deparadoxification,as expressed by Andersen (2003) and Knudsen (2006), through an applied statistical approach, namely ML. Using a recently developed state-of-the-art language representation model, the authors wish to investigate whether and how Luhmann's deparadoxification strategies, as explained by the conceptual framework, manifest themselves in the real world. This will be done using three key ML methods: embeddings, classification, and AL. All will be explained in this chapters. The goal is to see whether a ML model can learn to recognize deparadoxification strategies by analyzing the patterns found in the meeting minutes of the UNSC in order to answer the research questions, which can be found in chapter 1.1.

This chapter will introduce essential ML concepts to the reader. While it is not necessary to have a an expert understanding of these concepts in order to understand the methodology, it is recommended that the reader has a conceptual understanding of how the models work. In addition, to interpret the results correctly and grasp how the models were tuned, having a technical understanding of the evaluation metrics is also recommended. As a prerequisite, the reader should have a general understanding of ML. Fundamental concepts, such as the difference between *supervised* and *unsupervised* ML, or what a neural network is, will not be explained in detail. In addition, it is assumed that the reader is familiar with regression analysis, which will be used to answer RQ4.

3.1 Embedding

Within NLP, sequence modeling and transduction problems such as language modeling are prominent subfields (Chowdhary, 2020). Language modeling is a set of statistical and probabilistic techniques for determining the probability distributions of linguistic units, for example, words or sentences (Rosenfeld, 2000). Some of these techniques concern *embedding* the data, which is an integral part of modern NLP approaches. *Embeddings* are distributed representations trying to capture the syntactic and semantic properties

of the linguistic data (Turian et al., 2010). To illustrate, by embedding a set of words, each word is represented by a real-valued vector, called the embeddings. This is why embedding is sometimes referred to as vectorization or feature extraction because the method extracts features from the text in the form of vectors. Embedding linguistic data allows it to be used by ML algorithms, which can only read numerical data. Consequently, traditional ML tasks, such as classification, can be performed on linguistic data if it is first embedded. The two embedding techniques which will be used to represent the meeting minutes are the term frequency-inverse document frequency (TF-IDF), which will be used as the primary benchmark model, and the Bidirectional Encoder Representations from Transformers (BERT).

3.2 Term Frequency-Inverse Document Frequency

An embedding technique common to baseline models is TF-IDF (Aizawa, 2003). The technique is used to find the meaning of documents, normally sentences, based on the words they contain. In this context, each paragraph from the meeting minutes constitutes a document. TF-IDF is constructed in such a way that it assigns a weight to each word based on the calculated relevance. The idea behind TF-IDF is to solve the problem of former embedding techniques, such as bag of words, which only considers word frequency but fails to recognize word importance. This is partly why TF-IDF is the baseline embedding technique, as it is designed to consider relevance, but fails to account for sequential order and synonyms. In other words, TF-IDF is both *non-contextual* and *non-semantic*. Comparing TF-IDF to BERT, investigates whether using a *contextual* (and therefore also semantic) embedding technique, like BERT, is beneficial when trying to extract and transform deparadoxification strategies into feature vectors. Furthermore, TF-IDF is proven to be an effective vectorization method with broad applicability, and it is substantially cheaper than BERT in terms of computational resources. Suppose TF-IDF was to outperform BERT, or nearly match its results. In that case, one could argue that context is less important when determining what constitutes a deparadoxification strategy, instead of which words are used independently of each other.

The weight assigned to each word using TF-IDF is a product of two different statistics.

The first one is term frequency (TF). TF is similar to bag of words in the sense that it refers to how often each word occurs in a given document, which is calculated using Equation 3.1.

$$TF(w,d) = \frac{Number\ of\ occurrences\ of\ word\ w\ in\ document\ d}{Number\ of\ words\ in\ document\ d} \tag{3.1}$$

The second statistic, inverse document frequency (IDF) is what makes TF-IDF pay attention to relevance. IDF measures how significant a word is for the whole corpus. It does so by offsetting frequently occurring words in many of the documents, hence making them less relevant. This counteracts the inflated valuation of a word that TF might cause. It is calculated using Equation 3.2. Combining these two metrics by multiplication leaves us with TF-IDF, as defined by Equation 3.3.

$$IDF(w) = \log\left(\frac{Number\ of\ documents}{Number\ of\ documents\ with\ word\ w}\right) \tag{3.2}$$

$$TF\text{-}IDF(w,d) = TF(w,d) * IDF(w) \tag{3.3}$$

The numerical representation of a specific word, which TF-IDF yields, does not provide much information on its own, but when comparing the TF-IDF weights of two different words, the one with the highest value has the highest relevance for the document it is contained in. Training a supervised ML model, such as RF, with TF-IDF weights allows it to learn and recognize patterns and differentiate between the documents, depending on the quality and variability of the training data provided.

3.3 Introduction to BERT

There are many different types of neural networks applied within different domains and for different ML problems. One of these types is called recurrent neural networks (RNNs). RNNs were designed to persist information by making every input dependent on each other, providing each data point with a context. This was especially useful for sequence-related tasks such as language modeling. While impressive at their time, they suffered from short-term memory, having difficulties maintaining information over long sequences. This

is known as the vanishing gradient problem. As a result, long short-term memory (LSTM) and gated recurrent unit (GRU) models, both being a type of RNNs, were introduced to solve this issue. Their key features are the memory cell and their gates. LSTMs and GRUs were considered state-of-the-art approaches to language modeling for a long time (Vaswani et al., 2017). In 2014, the encoder-decoder model for RNNs was introduced to push further the boundaries of LSTMs and GRUs (Cho et al., 2014; Sutskever et al., 2014). The encoder consists of stacks of RNN cells, e.g., LSTM cells. Its purpose is to convert the input data into the required format, known as the hidden state. For example, in the context of the UN dataset, the encoder would convert a paragraph into a two-dimensional vector, the hidden state, which tries to capture the context and sequential dependencies between the words. Finally,the decoder tries to convert the vector to the desired output sequence.

3.3.1 Attention and the Transformer

In 2016, the attention mechanism, an extension to the encoder-decoder model, was introduced to make it easier for the model to deal with longer sequences (Bahdanau et al., 2015). The general idea of the attention mechanism is to allow the decoder to use the most relevant pieces of the input sequence in a flexible approach. It uses the weighted sum of all the hidden states as a context vector to focus the decoder's attention. While encoder-decoder models improved language modeling, they still struggled with long-term dependencies, and their architecture prohibited parallelization. To deal with these problems, the transformer model was introduced by Google Brain (Vaswani et al., 2017).

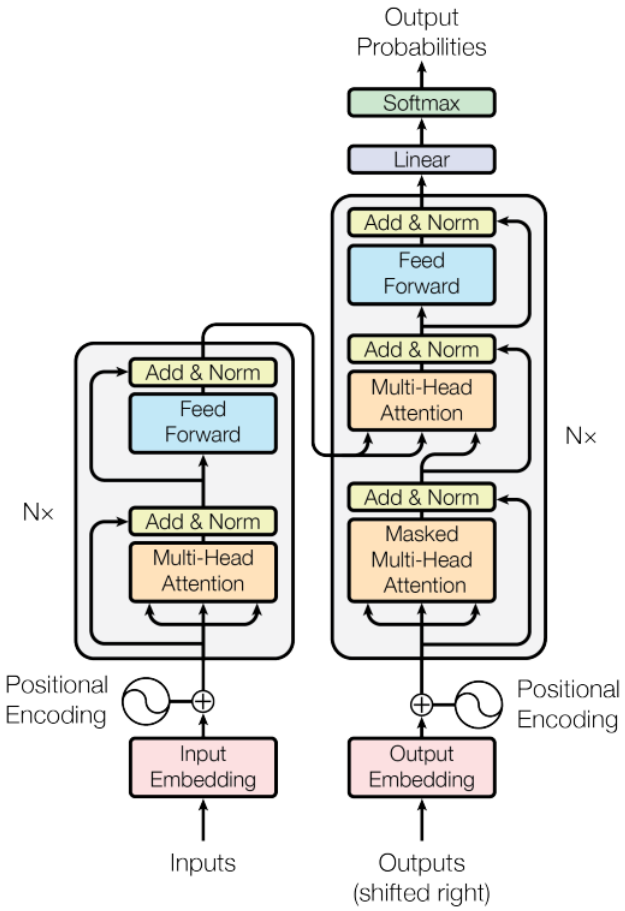


Figure 3.1: The Transformer - model architecture (Vaswani et al., 2017).

The transformer is designed as an encoder-decoder model for handling long-term dependencies (Figure 3.1). In contrast to LSTM and GRU encoder-decoder models, the transformer relies entirely on self-attention to compute representations of the inputs and outputs without using sequence-aligned RNNs. As opposed to a regular attention mechanism, self-attention allows the inputs to interact with each other. The main intention is to handle the dependencies between the inputs and outputs entirely with attention mechanisms and recurrence. As illustrated by Figure 3.1, the transformer mainly consists of two blocks: The encoder and the decoder. The encoder contains a multi-head attention layer and a regular feed forward neural network layer. The attention layer is referred to as multi-head attention because it computes the self-attention multiple times in parallel, and the outputs are concatenated and linearly transformed. The decoder has a similar structure, except that it has, in addition, a masked multi-head attention layer at the beginning.

3.3.2 BERT

In 2018, Google released a language model based on the transformer: BERT (Devlin et al., 2019). BERT applies bidirectional training of the transformer to language modeling. The transformer consists of an encoder and decoder, which is useful, e.g., machine translation. However, BERT only requires the encoder part (left side of Figure 3.1) since its goal is to generate a language model, instead of predicting sequential outputs. Hence, no decoding is necessary. Furthermore, BERT utilizes transfer learning, a ML method where a pre-trained model is used as a starting point for specific task-oriented training. In other words, the model is trained on one task and then repurposed for another. Hence, BERT is trained in two phases: Pre-training and fine-tuning.

The pre-training consists of two unsupervised prediction tasks, the first one being masked language modelling (MLM), and the second one being next sentence prediction (NSP)next sentence prediction (NSP). Google has released two pre-trained versions of the model: BERT base, which contains 12 stacked transformer encoder layers, and BERT large, containing 24 layers. BERT base, which will be used in this paper, has 768 hidden layers and 110M parameters. The reason for choosing BERT base is that it requires significantly fewer resources than BERT large. Both models were pre-trained on the whole English Wikipedia and the Brown corpus. During the first pre-training phase, MLM, the model randomly masks a set of the input tokens before trying to predict them based on their context. The purpose of MLM is for BERT to understand the relationship between words. During NSP, BERT performs a binary classification task: Given two sentences A and B, is B the actual next sentence after A, or is B just a completely random sentence from the dataset. This phase allows BERT to understand the relationship between sentences. While there are considerably more details and technicalities that allow BERT to function, this brief introduction is sufficient in providing an idea of how BERT learns to understand language during pre-training.

There are two distinct approaches BERT can be used to solve ML problems: *Fine-tuning* and *feature-based*. Fine-tuning entails adding an extra output layer at the end of the model, before further training the model for the specific task. The feature-based approach entails

using BERT only to extract the embeddings from the task relevant data, before using a different ML model to actually perform the task. As BERT's authors state, the incentive to use the feature-based approach is the major computational benefit of only using BERT once to retrieve all the embeddings and then run experiments on the embeddings with cheaper models (Devlin et al., 2019).This major cut in computational resources and processing time is the main reason why this thesis project uses the feature-based approach. A cheaper model allows the authors to run more experiments with different data subsets and hyperparameters. Ultimately, this leads to a more thorough investigation of how deparadoxification strategies can be detected using ML. The model chosen to be used on top of the BERT embeddings is RF. To summarize, BERT will be used to transform the textual data into numerical data, called the embeddings, and RF will be trained on the embeddings to perform the classification. This model combination of BERT and RF will be referred to as BERT-RF. RF will also be used on top of the TF-IDF embeddings to compare it with BERT. This model will be referred to as TFIDF-RF.

3.4 Random Forest

RF is a supervised ensemble learning algorithm used to solve classification and regression problems. It has broad applicability, and despite being around for over 27 years, it is still one of the most popular machine algorithms (Ho, 1995; Ray, 2019). An ensemble learning algorithm aggregates the predictions of a group of independent predictors. In the case of RF, these predictors are decision trees. A decision tree consists of three elements: a root node, decision nodes, and leaf nodes (Géron, 2017). Each tree works by letting the data sample flow from the top (root node) to the bottom (leaf nodes), splitting the sample at each decision node based on some parameter. The leaf nodes of trees designed for classification tasks have categorical outcomes. For example, in the context of classifying deparadoxification strategies, each tree contains four types of leaf nodes, each representing one class: Factual, social, temporal, and not relevant. The amount of decision nodes depends on the hyperparameters chosen for the model.

The concept of hyperparameters and how they were chosen during this study is explained in detail in Chapter 5.7.5. The decision nodes split the sample by asking Boolean questions

regarding the features of the sample. These questions are generated based on a chosen criterion, e.g., entropy or gini, which are both functions to measure the quality of the split based on statistical dispersion. RF typically combines hundreds or thousands of trees, training them using the bagging method. By using this method, each tree is trained using randomly selected, with replacement, samples from the data. By doing this, the overall variance when aggregating the trees are lower compared to that of a single tree. In addition, even though using a sample dataset for training the trees, instead of the whole dataset, increase each tree's bias, the overall bias is barely affected when aggregating the trees (Daumeé, 2017). The reason for choosing RF specifically as the classifier is that even though it is a fairly cheap model that takes less time to train than, e.g., a deep neural network, it is good at avoiding overfitting, and it handles higher dimensions and large datasets well (Sun et al., 2020). It also has a good classification and generalization ability (Li et al., 2010). In addition, RF is a non-parametric model, which entails that the complexity increase as the number of training examples increase, implying that the model is quite flexible and needs less training data than parametric models. This was beneficial to the project because even though it was unknown how extensive the training dataset would be or how the data was distributed, it was expected that the size of the training data would be rather small.

3.5 Active Learning

Because no labeled deparadoxification data exists, to the authors' knowledge, the training data, in this case, the UNSC meeting minutes, must be manually labeled before it can be used for supervised ML. AL will be used to boost this labeling process. AL is a subfield within ML that aims to make and labeling processes more economical by allowing the algorithms to influence the acquisition of the training data (Settles, 2011). In other words, AL is computing the statistically optimal way of selecting training data (Cohn et al., 1996). This is useful when the dataset is unlabeled, which is true for the UNSC dataset, and when labeling the data is resource-demanding. In addition, it is well established that using AL can lead to higher accuracy models with fewer annotated instances. Considering the labor-intensive activity of reading and evaluating hundreds of paragraphs of political speeches, applying AL ensures a more efficient labeling process and possibly a better model, and hence fits well with the purpose of this paper.

There are different approaches to AL, depending on the purpose and context of the project. This paper use pool-based sampling, a method suitable when there is a large pool of unlabeled data from which you want to draw out the most informative instances (Wang, 2014). All instances, or subsets, of the data are assigned a confidence score based on how informative each instance is. Then the most informative instances are selected to be labeled by the annotators, termed oracles. To illustrate, in a scenario where a linear classifier is used, the most informative instances would be the ones closest to the decision boundary. Labeling one of these instances and then re-training the model would likely move the decision boundary more than if an instance far away from the decision boundary was labeled. In other words, the most informative instances are the ones in which the model is the most uncertain of. However, there are different ways how evaluating informativeness. The strategy of choosing which data examples to label next is called the *query strategy*. The strategy chosen for this paper, *least confidence sampling*, will be elaborated on in the AL chapter (Chapter 5.7.4).

3.6 Evaluation

Choosing the right metrics to evaluate the model plays an important role in achieving the optimal classifier. The evaluation metrics function as feedback based on how well the model performs, which is especially helpful during hyperparameter tuning. This chapter will describe the most fundamental evaluation metrics chosen to iteratively evaluate the model, and how they function. The metrics that will be described are the confusion matrix, accuracy, precision, recall, F1-score, and the difference between micro and macro measures. While these are not the only evaluation metrics that will be used, these are the most fundamental ones that the reader should understand well (Skansi, 2018).

Accuracy shows the ratio of correct predictions over the total number of evaluated instances (M and M.N, 2015). In other words, it shows the fraction of predictions that were correct. It does so by adding the true positive (TP) and the true negative (TN), before dividing the sum with the total number of instances, including the TP, false positive (FP), TN, and false negative (FN) (Equation 3.4).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.4}$$

Precision indicates how well the classifier avoids FPs by dividing the TPs with the sum of the TPs and FPs (Equation 3.5). Recall indicates how many possible TPs the classifier managed to classify correctly. It is calculated by dividing the number of TPs with the sum of TPs and FNs (Equation 3.6). In other words, recall shows how many class instances the model managed to classify correctly. Precision and recall are competitive metrics, because increasing precision will lower the recall, and the other way around. For instance, if the model would predict that every paragraph is of the class *temporal*, then the recall for *temporal* would be 100% because the model successfully detected every instance of the recall class. However, the accuracy would likely be far off, as the model predicted that all data instances in the dataset are *temporal*.

$$Precision = \frac{TP}{TP + FP} \tag{3.5}$$

$$Recall = \frac{TP}{TP + TN} \tag{3.6}$$

A confusion matrix shows the true values for each class, displayed in the rows of the matrix. The columns show the values that the chosen model predicted. Because this paper investigate four different classes (*not-relevant, factual, social* and *temporal*), the matrix will contain four rows and four columns. Investigating the confusion matrix values can give insights into how well the model distinguishes the different classes.

Table 3.1: Confusion Matrix - Example

Predicted Positive Class	True Positive (TP)	False Negative (FN)
Predicted Negative Class	False Positive (FP)	True Negative (TN)
	True Positive Class	**True Negative Class**

The F1-score shows the harmonic mean of precision and recall (Takahashi et al., 2022). In other words, it combines the two metrics. This is especially useful when the goal

is to maximize precision and recall, which otherwise are competitive metrics. The F1-score is normally used when FPs and FNs are more or less equally undesirable. This makes it suitable to be used as the main metric for evaluating the models in this paper, considering that the goal is to maximize precision and recall. In addition, FPs and FNs are equally undesirable. It is worth to note that the micro averaged F1-score for multi-class classification when each data point can only be assigned to exactly one class, is the same as accuracy. The concept of micro averaging will be elaborated on in the next paragraph. The F1-score is also quite useful when the classes are imbalanced, which might be the case for the deparadoxification strategies, but this remains to be seen. The F1-score is calculated using Equation 3.7. F1 is also a good choice when comparing different models, which this paper will do by comparing a contextual language model (BERT-RF) with a non-contextual model (TFIDF-RF).

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.7}$$

Because this paper deals with a multi-class classification problem, averaging different metrics will be relevant for the model tuning and evaluation. There are several ways to do this, but the two most relevant are *micro* and *macro averaging*. These different techniques provide different perspectives on the model, each insightful in their own way, hence both of them will be used during evaluation. In short, macro averaging does not take data distribution into account, while micro averaging does. Macro averaging computes the given metric, e.g., F1, for each class independent of each other, before averaging the scores. Hence, the Macro F1-score treats all the classes equally, assigning just as much importance to, e.g., *factual* as *not-relevant*. On the other hand, micro averaging considers the contribution of each class by being biased by the class frequency. It does so by calculating the averages globally instead of first considering each class independently. This is relevant when one wishes to investigate how well a classifier performs on an unbalanced data distribution when taking the distribution into account. If the distribution is perfectly uniform, micro and macro averaging yields the same results. See Sklearn's documentation on metrics and scoring for further reading (Sklearn, 2022).

3.7 Technical Background Conclusion

This chapter has introduced the essential models, techniques, and evaluation metrics that will be used throughout this paper. To summarize, to investigate the deparadoxification strategies, the three main ML concepts that will be applied are embedding, classification, and AL. First, each paragraph in the dataset set will be embedded, which entail transforming the textual data into numerical vector representations. Two embedding approaches will be applied, BERT and TF-IDF. These approaches will be compared to each other to investigate whether a contextual and complex model, such as BERT, is better for extracting the patterns of deparadoxification than a simple non-contextual model, such as TF-IDF. In order to use the embeddings to answer the research questions, a RF classifier model will be trained using the embeddings as training data. This will result in two models, BERT-RF and TFIDF-RF, whereas TFIDF-RF will function as the main benchmark model. The goal of these models is to predict which deparadoxification strategy each data point belongs to. The data points are the embeddings, and each embedding represents a paragraph in the UNSC meeting minutes. The deparadoxification strategies will be represented using four classes: *Not-relevant, factual, social* and *temporal*. Because there is a complete lack of available training data on deparadoxification strategies, the data must be manually labeled. In order to make this process more efficient, AL will be applied. AL is a technique for letting the model decide which data points should be labeled next to maximize the learning based on a query strategy. Different evaluation metrics will be used to evaluate and tune the models, both with and without micro and macro averaging. These metrics are mainly precision, accuracy, recall, and the F1-score, in addition to the confusion matrix. The F1 micro score will be the main choice of metric. In the context of non-multi-label classification, F1 micro is the exact same metric as accuracy.

n

4 United Nations Security Council

This chapter introduces the organization chosen as the case study for analyzing deparadoxification through ML, namely the UNSC. The purpose of the chapter is to give the reader an understanding of why the UNSC was chosen for this project, in addition to a basic understanding of the UNSC's purpose and how they function.

There are four main reasons why the UNSC, precisely its meeting minutes, was chosen for this project. Firstly, the UNSC has highly structured and consistent meeting minutes. Considering the complexity and intricacies of NLP and ML, especially when combining it with deparadoxification theory, having a well-structured and consistent dataset is of high priority. This will make it easier to process the data and train the model, possibly allowing higher predictive performance. Furthermore, it allows for more time to be invested in model building and analysis, instead of data cleaning and data preparation. Secondly, because the core activity of the UNSC is to decide upon resolutions, the dataset consists fundamentally of different social actors discussing and making decisions. This fits perfectly with training a model that looks for patterns of deparadoxification. Thirdly, because the dataset contains the voter outcome of resolutions, it is possible to investigate relationships between deparadoxification strategies and decision outcomes. Because of this reason, not only does the dataset allow for the training of a classifier, but it makes it possible to explore whether and how deparadoxification strategies affect decisions. Lastly, because the UNSC is arguably one of the most important peacekeeping organizations, understanding of their communication and decision-making further expands society's knowledge base for successful peacekeeping solutions.

The UNSC is considered the centerpiece of the UN, with all 193 UN member states accepting its decisions as binding, despite criticism regarding its undemocratic character (Luck, 2006). The purpose of the council is to maintain international peace and security. However, it has no direct obligation to take responsibility for every international security crisis (Bellamy and Dunne, 2016). The UN charter allows the UNSC to define most of its own agenda and adopt whatever course of action. Enforcement measures such as

economic sanctions, ceasefire directives, or collective military action might be initiated if a dispute cannot be settled by those involved. Due to limited resources, the council's decisions often anticipate cooperation with regional organizations (Malone and Malone, 2004).

The council consists of 10 non-permanent members, elected for a two-year term, and five permanent members: the USA, China, Russia, France, and the UK (Basu, 2004). The members take turns in holding the presidency on every month. The president calls and conducts the meetings and approves the provisional agenda (Security Council Report, 2019). While some of the meetings are held in private, the large majority are public, in the sense that the meeting minutes are published (Sievers and Daws, 2014). In addition, non-members may be invited to a meeting if their input is needed. The provisional agenda of each meeting is constructed by the UNSC's Affairs Division and then approved by the Secretary-General. There are six official languages of the UNSC, of which all meetings shall be translated to, however, any representative may speak in any language they prefer, as long as they provide interpretation into at least one of the official languages (Report, 2019).

The president calls the meetings at any time he deems necessary, in addition to periodic meetings, those requested by either a UNSC member of the General Assembly or if any UN member state brings a situation that might threaten international peace and security to the attention of the council (UN, 1983). The meetings typically follow the same structure, as specified by the Provisional Rules of Procedure (Sievers and Daws, 2014). The provisional agenda is circulated to all the member states at least 21 days before any period meeting and three days for non-periodic meetings. Urgent circumstances, however, allow for exemptions.

The meetings start with the adoption of the agenda. The president then calls upon the representatives to speak in the order they have signified their desire. Next, proposed resolutions, amendments, and substantive motions are presented in written form to all representatives. A resolution constitutes a formal decision made by the council. While any UN member state can submit proposals, only UNSC members can request that a vote be held (Security Council Report, 2019). There are two main types of decisions: Votes

on procedural matters and those on substantive matters. A procedural matter requires nine affirmative votes, while substantive matters require, in addition, an affirmative vote from all the permanent members (Security Council Report, 2019). If a draft resolution receives the required votes, normally obtained by a show of hands, it is adopted and is given a resolution number. The resolution numbers are especially important for tracking decisions for the regression analysis in Chapter (5.7.6).

5 Methodology

5.1 Introduction to Methodology

The following chapter will present and describe the methods and techniques used in this study and how they were applied. The purpose is to connect the research purpose, conceptual framework, technical background, and the research process. The Cross Industry Standard Process for Data Mining (CRISP-DM) model was applied as a guide for the methodology and will therefore be explained initially. CRISP-DM will also visualize how the different applied techniques and research activities fits together to answer the research questions. In addition, a methodology outline based on CRISP-DM will be presented.

CRISP-DM was developed by practitioners who contributed with their knowledge and real-world data mining experience as a blueprint for data mining (Shearer, 2000). The purpose of the CRISP-DM model is to provide generic guidance on data mining across industries (Shearer, 2000). Even though the paper was published over two decades ago and research within this field has emerged, the model has not been revised (Martínez-Plumed et al., 2019). It is common tool used within data science; it is still highly relevant to apply as it breaks down a data mining project into six specific phases. Data mining is the process of extraction and exploration of patterns in data sets using approaches from machine learning, statistics, and database systems. The study is applying machine learning and statistics, and for this reason the CRISP-DM framework is highly suitable for this research domain.

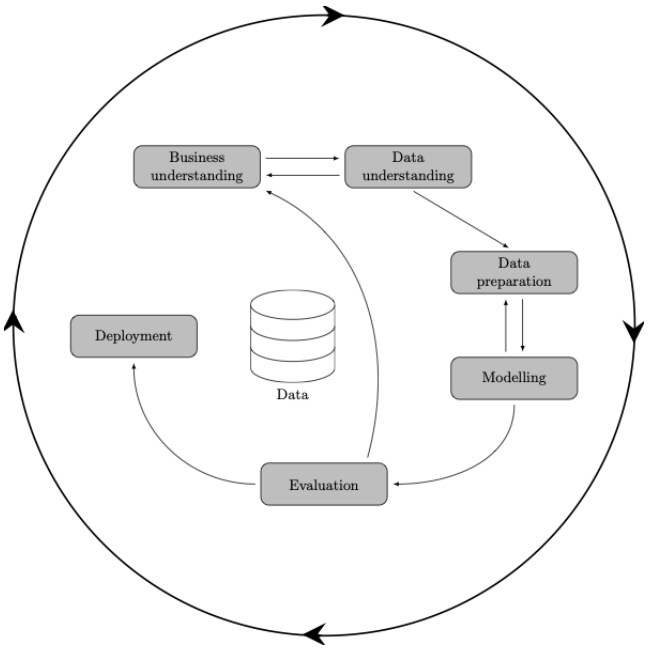


Figure 5.1: CRISP-DM model, adopted by Shearer (2000)

Figure 5.1 shows the different phases of CRISP-DM, which are: 1) business understanding, 2) data understanding, 3) data preparation, 4) modeling, 5) evaluation, and 6) deployment. As Figure 5.1 visualizes, the CRISP-DM model has frequent dependencies between the phases and is iterative by nature, meaning that the users have to go both back and forth to achieve set goals.

To illustrate where each stage of CRISP-DM can be found in this paper, Table 5.1 provides an overview of CRISP-DM with the relevant chapters of each stage. The first step is business understanding, which for this paper consists of understanding deparadoxification and its different strategies, how the UNSC function, and how the research questions should be approached. The second stage is data understanding, which not only concerns data understanding itself (Chapter 5.2), but also how it should be labeled (Chapter 5.3). Data preparation focus on preprocessing (Chapter 5.6) and labeling, in addition to setting up the machine learning pipeline (Chapter 5.4). Modeling (Chapter 5.7) entails building and training the different models and improving them through AL and hyperparameter tuning (Chapter 5.7.5). Through iterative evaluation, the models will be improved, before finally being tested on the test set. The results from the final evaluation (deployment) will be presented in Chapter 6 and discussed in Chapter 7. At the center of the model is the

data, which for this paper is the UNSC meeting minutes.

Table 5.1: CRISP-DM Chapter Overview

CRISP-DM Stage	Chapter	
	Name	Number
Business Understanding	Conceptual Framework	2
	United Nations Security Council	4
Data Understanding	Data Understanding	5.2
	Data Labeling	5.3
Data Preparation	Data Labeling	5.3
	ML Pipeline	5.4
	Preprocessing	5.6
Modelling	Modelling	5.7
	Active Learning	5.7.4
	Hyperparameter Tuning	5.7.5
Evaluation	Active Learning	5.7.4
	Hyperparameter Tuning	5.7.5
Deployment	Results	6
	Findings and Discussion	7

Even though the CRISP-DM framework is used as a guide, there are still some shortcomings of the framework (Martínez-Plumed et al., 2019). According to Martínez-Plumed et al. (2019), the CRISP-DM framework work well with goal-oriented or process-driven projects, but as highlighted the framework lack of flexibility for exploratory or data management projects. CRISP-DM still fit the purpose for data science projects. As this paper has both social science, exploratory, and technical elements that may be challenging to follow, the authors has created a more suitable outline for parts of the methodology specifically for data understanding, data preparation, modeling and evaluation (Figure 5.2). The figure shows a high level outline of the different steps from data selection to results and how they are connected.

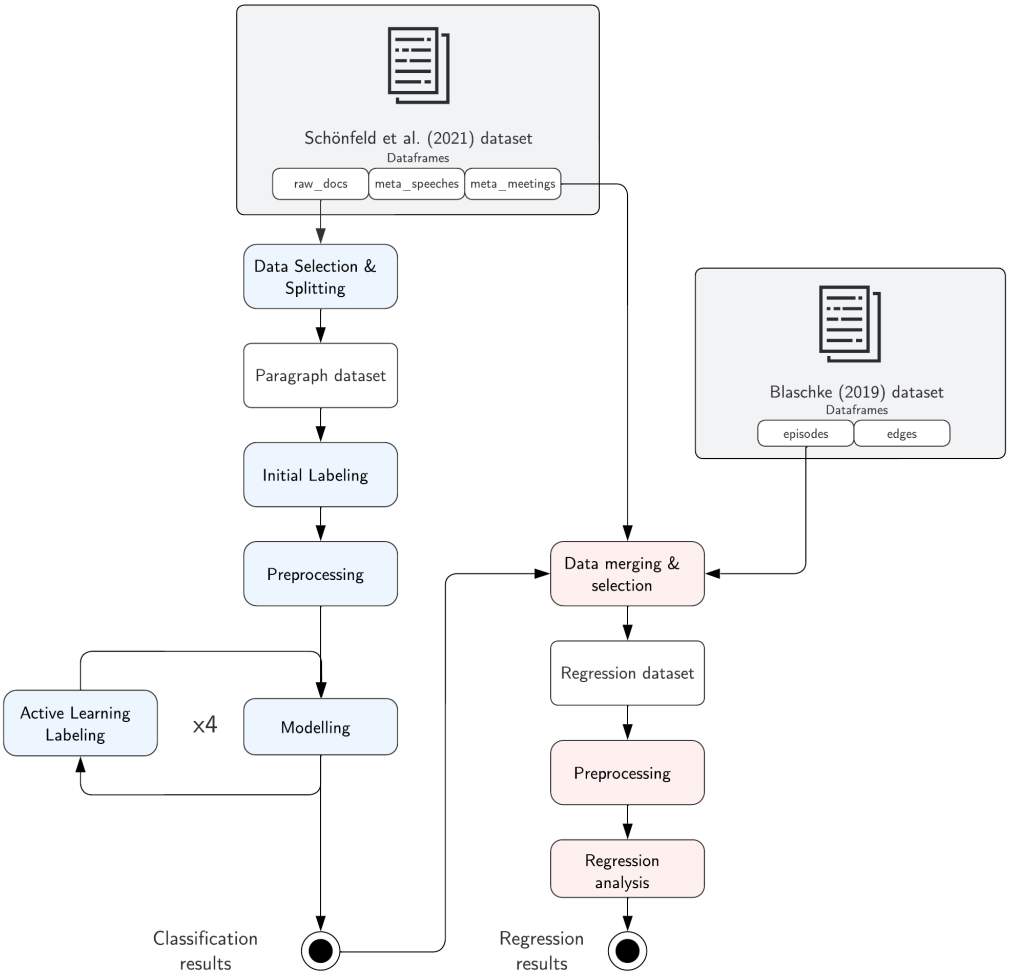


Figure 5.2: Methodology Outline

5.2 Data Understanding

This chapter will explain the setting and context of the data. This includes considerations of processing data related to natural persons, data collection, and data description. Data exploration and descriptive statistics will also be presented. The labeling of the data will be explained in its own Chapter (Chapter 5.3)

5.2.1 Data Context

In the context of deparadoxification, the literal language used by the representatives is of absolute importance. Considering that any of the six official languages can be used, there always exists a natural possibility of interpretation error or ambiguity. But more importantly, the diplomatic setting must be considered given its significant

implications regarding how the representatives are formulating their messages. Diplomacy, and therefore diplomatic language, is the activity of managing international relations, according to Press (2022). It differs from regular day-to-day language in the sense that it focuses on presentation and convincing, while trying to transcend cultural boundaries. Normally formalized, it also requires a high degree of situational understanding and political context (Kurbalija and Slavik, 2001). It should rather be understood as an instrument of soft power as opposed to merely means of communication (Jungblut, 2017). Language wise, diplomats tend to use words that are precise yet elastic enough to suggest alternative meanings to please multiple stakeholders (Constantinou et al., 2016). In addition, policy rhetoric tends to be bureaucratic and programmatic (Wodak and Krzyżanowski, 2008). All these factors might influence how evident the deparadoxification strategies are in the meeting minutes, and how the strategies are used as compared to an organization that is not based on diplomatic communication.

When studying diplomacy communication and meetings minutes, there may be personal data, such as names, that are being processed. As the data contain personal data, and when processing such data, cf. Art. 4 (1) the General Data Protection Regulation applies. For this case, the personal data processed is the names of the participants, who are natural persons, written in the meeting minutes. According to regulation, it is required to have a valid lawful basis, and for this research, the lawful basis is identified as legitimate interest, cf. Art. 5 (1)(b). The data is publicly available by the UNSC, and the UNSC has specific rules for publications of meeting minutes, such as provisional rules of procedure Chapter IX, which are known to the participants. To justify the legitimate interest, purpose has to be clear, it needs to be necessary and do not override The purpose is to research of deparadoxification as a phenomenon, and it is necessary to be processed as one of the strategies is regarding social actors (natural persons). As the data is already made publicly available by the UNSC and is compliant with the rules of the UNSC, which the member state and participants are aware of. Additionally, the natural persons in the data are public persons, and it is likely to believe that the individual interests do not override the interest of this research.

5.2.2 Data Collection

According to CRISP-DM, data collection is the initial part of data understanding. The data for the research project was collected through two existing datasets. The first dataset, created by Schönfeld et al. (2021), was publicly available through Harvard Dataverse. This will be referred to as the Schönfeld et al. (2021) dataset. The second dataset, attained from Steffen Blaschke, will be referred to as the Blaschke (2019) dataset. Both datasets contain the public meeting minutes from the UNSC, from almost the same period. Both dataset have had extensive cleaning steps, and they are consider to be of high quality by the authors.

5.2.3 Data Description

Figure 5.2 presents the methodology outline, but also an overview of the datasets from Schönfeld et al. (2021) and Blaschke (2019) and their composition of subdatasets, which will be referred to as dataframes. The Schönfeld et al. (2021) dataset contains three dataframes: *raw_ docs*, *meta_ speeches*, and *meta_ meetings*. The Blaschke (2019) dataset contains two dataframes: *episodes* and *edges*. To construct the *paragraph dataset*, which will be used for classification, only *raw_ docs* was used. To construct the *regression dataset*, which will be used for the regression analysis, a combination of *meta_ meetings*, *episodes*, and the classification results was used.

5.2.3.1 Schönfeld et al. (2021) dataset

Raw_docs The *raw_ docs* dataframe contains two columns (*doc_ id* and *text*), and 82,165 rows, each representing a speech. The dataframe contains all public meetings during the time period 1995 to 2020. Table 5.2 shows an overview of its contents. The *text* column contains the raw text data from the meeting minutes, in the form of speeches, which will be used as training data after going through a series of preprocessing and labeling steps. A speech is defined as taking place from the moment a representative starts talking, until she or he stops (indicated by another representative starting to talk or the meeting ending). The *doc_ id* identifies each individual speech by combining the "S/PV"-meeting record identifier and the numbering of the speeches within a given meeting (Schönfeld et al., 2021).

Table 5.2: Overview of docs_raw

Metrics	Columns	
	doc_id	text
count	82,165	82,165
unique	82,165	73,946
frequency	1	92

Table 5.2 shows that all values in *doc_id* are unique. However, not all of the speeches are unique. In fact, 10% of the speeches are duplicates. This is due to the share of communication associated with meeting etiquette and formalities, such as welcoming statements from the President or presentations of the voting outcome. The frequency shows how often the most common value occurs. This imply that the most frequent speech appears 92 times in the data. This specific speech is "The President: I now give the floor to(...)".

Meta_speeches The *meta_speeches* dataframe contains 18 columns and 82,165 rows, each representing a speech. The purpose of the dataframe is to present a deeper understanding of *raw_docs* and the context of the speech.The dataframe contain columns such as date of the speech, the speaker, speaker nationality and role in the UN, meeting number, overall topics, and number of types, tokens and sentences. The full columns description can be found in Appendix A1.1. Table 5.3 shows the descriptive statistics. The average speech had 261.43 types, 636.66 tokens and 22.33 sentences. The types are the number of unique tokens in each meeting. Moreover, the standard deviation is almost the same as the mean, meaning that the speeches have high variation in terms of number of types, tokens, and sentences.

Figure 5.3 shows that there has been an increasing number of meetings per year from 1995 until 2020. Additionally, the average number of speeches per meeting has doubled during the same period. In 1995, the number of meetings was 136, and the average number of speeches per meeting was 10, resulting in 1,374 speeches for that year. In 2020, the average amount of speeches was 18 per meeting, and the number of meetings was 300, resulting in a steep increase in the number of speeches since 1995, total in 82,165. However, by comparing 2019 and 2020, the number of speeches decreased by 30%. It is

Table 5.3: Descriptive statistics for meta_speeches

Metrics	Columns		
	types	tokens	sentences
mean	261.43	636.66	22.33
std	208.61	610.09	21.84
min	6	6	1
25%	37	48	3
50%	274	601	20
75%	390	936	33
max	2,380	13,569	661

a decrease both in the number of meetings per year but also the average number of speeches.

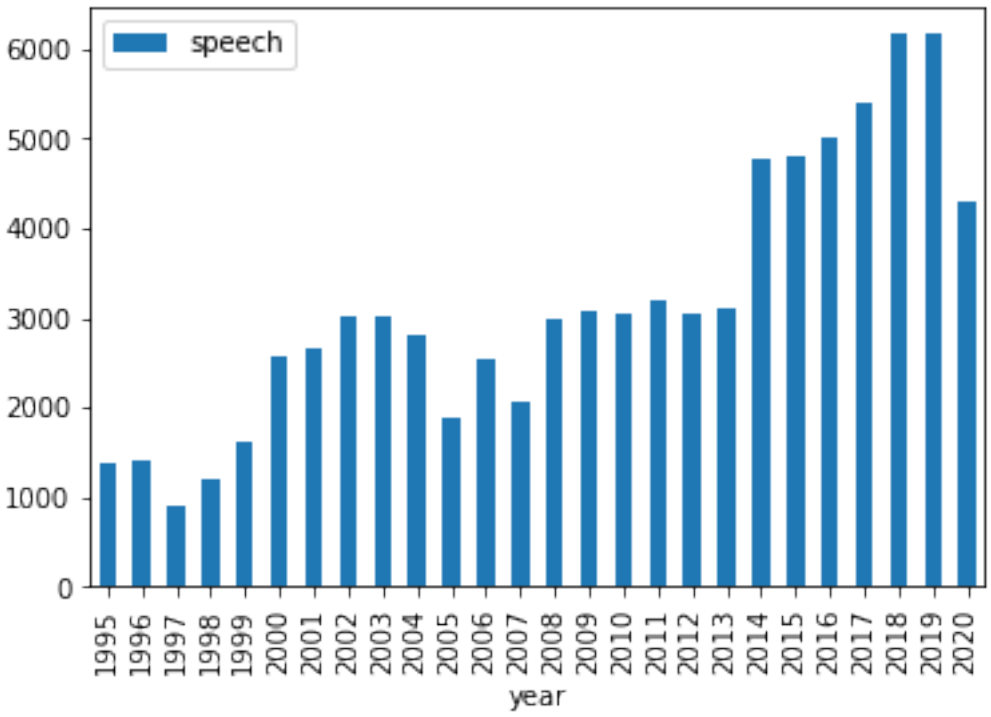


Figure 5.3: No of speeches per year from 1995 to 2020

It is likely that the number of speeches and meetings are affected by large-scale conflicts. The year 2017, which is the year with the most meetings, is also the year of e.g., the civil war in South Sudan. As shown in Figure 5.3, there are more than 2500 speeches annually in the period from 2000 to 2004, which is 1000 more annually than in 1999. During this period, there were ongoing wars in Iraq and Afghanistan. Another

observation is that the top five nationalities amongst the speakers are the P5 countries, which is not surprising as they consequently participate in all meetings. Furthermore, each speech's percentage of P5 nationality is seemingly consistent in the given period, which is interesting considering that the P5 countries are often not the center of the debate.

Meta_meetings The *meta_meetings* dataframe consists of nine columns and 5,748 rows, each representing a meeting. The list of columns and its description can be found in Appendix A1.2. According to descriptive statistics, the average meeting had 14 speeches for the period 1995 to 2020. The shortest meeting during this period had 0 speeches, while the longest had 178 speeches. For the 5748 meetings, there were 541 different topic discussed. On average a topic is discussed in more than 10 meetings, but with a standard deviation on 20.56. The minimum number of times a topic was discussed is one, while the maximum is 178.

Figure 5.4 shows the top most 15 discussed topic per meeting. The is clear, the most discussed topic per meeting is the situation the Middle East both with or with the Palestinian question. The figure in combination with descriptive statistics says something about the topics within in the UNSC, and there are very many topics that only frequently reoccur and that it is the same topics discussed as they might not have solved it.

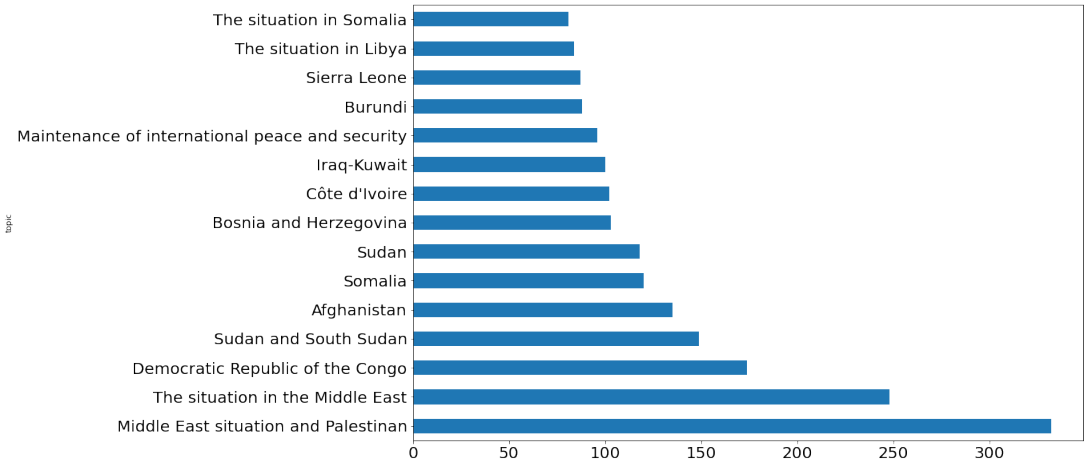


Figure 5.4: Top 15 meeting topics, 1995-2020

5.2.3.2 Blaschke (2019) dataset

The Blaschke (2019) dataset contains two dataframes: *episodes* and *edges*. The *episodes* dataframe was extracted from the meeting minutes, while *edges* was constructed using

data in the meeting minutes to create a network of nodes consisting of meetings and resolutions. As only *episodes* are used for this paper, *edges* will not be discussed.

Episodes The *episodes* dataframe contains eight columns and 4,820 rows, each representing a meeting (Table M:tab:episodes). The meetings are observed during the period of 1. of June 1944 to the 31. of December 2016. The columns as described in Appendix A1.3. Especially noteworthy is the *meetingrecord*, *vote*, and *resolution* column, which will be used for the regression analysis. *Vote* will be normalized and used as the dependent variable, while *meetingrecord* and *resolution* will be used to locate the relevant paragraphs in the *paragraph dataset* (which obtains its data from splitting the speeches in *raw_ docs*).

Table 5.4: Descriptive statistics for Blaschke (2019) episodes

	Meeting record	speakers	duration
count	4820	4312	3807
mean	2410.50	9.	69.48
std	1391.56	11.05	92.44
min	1	1	0
25%	1205.75	1	5
50%	2410.50	3	25
75%	3615.25	16	120
max	4820	86	750

5.3 Data Labeling

As part of the data understanding phase, labeling the data is essential for data modeling. Data labeling is defined as adding a label denoted to a data point. The label ascribe a high-level fact of particular interest, in this case a deparadoxification strategy, to a data point. Labeling data is in general assumed to be costly and challenging due to the involvement of human labor. The content analysis framework by Krippendorff (2004) was applied for the data labeling to make valid inferences from the meeting minutes to the context of deparadoxification. The framework, consists of six essential components: 1) text, which is available for the author, 2) research question, which the author should answer by analyzing the text, 3) context, which is a choice made by the author on how to interpret the text, 4) analytical composition that establishes what the author has

interpreted about the text, 5) reasoning that is meaning to answer the posed research question, and then at last 6) validation of evidence (Krippendorff, 2004).

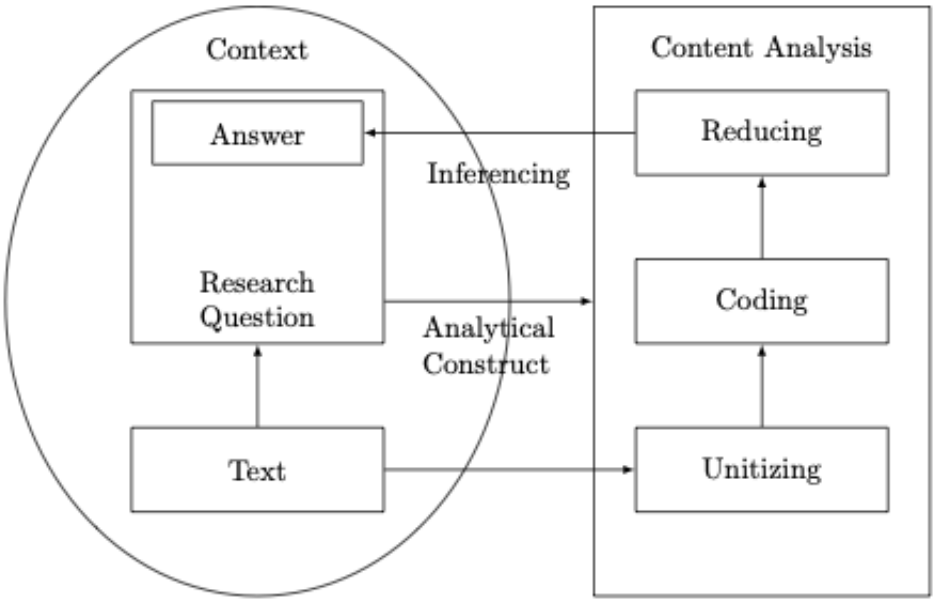


Figure 5.5: Content Analysis Framework inspired and adopted by (Krippendorff, 2004)

Figure 5.5 highlights the most essential part of the content analysis. The model is constructed as follows; the ellipse and its content represent the first three components. The context, visualized as the ellipse, is the author's understanding, and the context contains three nodes elements named as texts, research question and answers to the research questions. Both elements are visualized in the ellipse to highlight that they need to be interpreted within the context. Finally, the content analysis assumes some correlation between the understanding and the potential answers to the research question, representing the last three components of the framework. The rectangle represents the correlation on the right-hand side, which includes the following steps: unitizing and sampling the text data suitable for understanding, followed by the transformed units used for the resulting are either coded manually or automatically (Krippendorff, 2004). Then the answer to the research question is deduced from the findings of the analysis, represented by pointing back at the context, precisely the answer.

As with any empirical research, content analysis starts with data. In common with most content analyses, and this case, the data analyzed is not indented to be analyzed for a specific purpose, e.g., research question (Krippendorff, 2004). Although the text is meant to be interpreted explicitly, that may not always be the case. As Krippendorff (2004)

argues, the text is meant to be read by other than the authors, meaning that the readers may decompose the text into units to recognize compelling structures (Krippendorff, 2004). However, as Krippendorff (2004) states, there must be an assumption that authors would be able to be understood by others. Furthermore, as part of the framework, the data will be decomposed into units made by the authors that are more sensible for the analysis and understanding. In this case, the meeting minutes were the original text data, unitized as speeches in the data set created by (Schönfeld et al., 2021). In this case, the data was decomposed into smaller unit sizes, explained in the following subchapter.

5.3.1 Unitizing

As the research question and context are already set by Chapter 1-4, the next step in the content analysis is unitizing. Unitizing is about systematically recognizing segments of data relevant to the analysis, unit of analysis. When deciding on unit of analysis, the human annotators needs to be taken into considerations, and for this study a human annotator is referred to as a oracle. A oracle refers to the human expert(s) or labeling source that provides the correct labels, which in this research are human annotators.

There are three types of units: sampling, coding, and context units. Sampling units define units by selected and included data in the analysis (Krippendorff, 2004). Meaning the sampling unit for this study is the UNSC meeting minutes from 1995 to 2020. It could be argued that issues of newspapers are not independent because of events unfold in time and are connected to previous publications. It is the same for the meeting minutes from the UNSC. Even though the meeting minutes are highly connected to the resolutions, as the text often refers to resolutions, it is not relevant for the research. Therefore, there are no connections that seem to be necessary to include other than the meeting minutes, but by not including older meeting minutes than 1995, there may be a bias. However, the connections of the meeting minutes are not the focus of the research.

Then the coding units are defined as separated units for the purpose of labeling and transcription. As coding units are a part of the sampling units, a coding unit will, by definition, never exceed a sampling unit. Krippendorff (2004) highlighted that a coding

unit is preferably significantly smaller than the sampling units as the sampling units generally contain too much data, which can make it complicated to describe accurately. In the guide by Krippendorff (2004), it is not decided on how large or small a coding unit can be as it is up to authors to settle within the context of the study. In this paper, paragraphs have been decided as the coding units, and is the unit size to be labeled. Any limitations regarding this decision will be discussed in Chapter 7.

Last but not least, the context units are units that limit the quantity of data that can be considered by coding units (Krippendorff, 2004). In other words, to identify the meaning behind a coding unit, e.g., a word from a list of dictionary entries, it is necessary to examine the context of the coding unit. In this study the context unit and coding unit are of the same size. The best practice of deciding on context units is as large as meaningful and as small as feasible (Krippendorff, 2004). For example, a speech includes multiple paragraphs, and could be a good suggestion for a context unit. At the same time, as a speech may consist of many paragraphs, and the reliability goes down. Consequently, it would be more difficult for the oracle(s) to label the correct category for a speech. However, it would be the same trade-off if the context and coding unit were a sentence. As of this research, the coding units are paragraphs that include enough context to identify if there are any deparadoxification strategies identified. The concrete examples presented by Knudsen (2006) have in common that the examples do not exceed a paragraph, which can justify the use of paragraphs as both coding and context unit. Paragraphs appear to include enough context without resulting in too much lower reliability.

Meaning that the unit of analysis, is *paragraphs* of the UNSC meeting minutes given the period of 1995 to 2020. As the unit of analysis is decided, there is was a need for creating labeling instructions before starting with the initial data labeling. The next chapter will present the chosen labeling instructions for this study.

5.3.2 Labeling instructions

As highlighted by Krippendorff (2004), to be consistent with scientific standards, it was necessary to create clear instructions for labeling to achieve results that could be replicated. The labeling instructions, inspired by Krippendorf's (2004) framework, which was created

to accurately label the deparadoxification strategies contains three components: The first

1: Qualification list for oracles

2: Practical descriptions of the units and how to distinguish them, practical description of the syntax, and semantics of the data language, meaning the categories which the oracles will apply when data labeling.

3 : Explanation of the tool applied for data labeling.

part of constructing the data labeling instructions was to decide on the qualification list for the oracles, and whom should be the oracles.The authors of this paper has labeled the data, and during this process the authors are referred to as the oracles. There are especially two critical factors for oracles to qualify: cognitive abilities and appropriate background (Krippendorff, 2004). Data labeling needs consistency and is a repetitive task which requires cognitive abilities such as attention to detail and being able to focus over time. Furthermore, the oracles need to have the appropriate background for ensuring high reliability of the labeling. It must be explicitly described so that other authors with similar research propositions can find suitable oracles comparable to the ones used for this research paper. As Krippendorff (2004) applies the term appropriate background, it is related to similar involvement with texts, education, and social sensitivities. Additionally, it is stated that familiarity with the phenomena under consideration is another essential part of the appropriate background (Krippendorff, 2004). For this paper, knowledge of deparadoxification was highly valued. Therefore, it makes sense that the oracles have similarities in either cultural, educational, or professional backgrounds. For this paper, the oracles have good cognitive abilities and similar educational and cultural backgrounds. In addition, the oracles have an adequate understanding of deparadoxification and TSS. It is important to note that the qualification criteria must not be too narrow as enough potential oracles should be available within the population, in order to ensure that the study is replicable.

5.3.2.1 Qualification list for oracles

Based on the discussed qualification evaluations, the following criteria were set for the oracles: Preferably, the roles of oracles and researchers should be separated. The

1: Good cognitive skills, especially in terms of attention.

2: Familiar with the phenomenon of deparadoxification and Luhmann's TSS.

3: The group of oracles should have similarities in either cultural, educational or professional backgrounds. The more similar background, the better.

researchers may have obtained an unspoken understanding of which new oracles will not posses (Krippendorff, 2004). This puts constraints on other scholars wanting to use the instructions. The authors of this paper are well-aware of this limitation of being authors and oracles, but due to resource restrictions, such as sufficient time to find and train suitable oracles that fulfills the qualifications list, it was necessary. However, by applying cross-checking as a criteria for the labeling, the negative implications of having the authors assume the roles of oracles are alleviated to some degree. The limitation will be further discussed in Chapter 7.

5.3.2.2 Operational category description

The second component of constructing labeling instructions is to define operational category descriptions. There are two requirements for data labeling when defining a set of categories. Firstly, the categories needed to be exhaustive, meaning that the categories should represent all possible coding units. Secondly, the categories have to be mutually exclusive; in other words, it has to be non-overlapping categories for representing the phenomena of deparadoxification. All the four categories together represent the all possible coding units outcome .The first three categories represents the different strategies of deparadoxification: factual, social, and temporal. However, these three strategies do not represent all the possible outcomes of communicational categories. Hence, there is a need for two additional categories; mixed and not-relevant. By adding the category mixed, which will be discussed and justified later in this paragraph, the categories do overlap, meaning that the categories do not accurately reflect the text.

Then, creating the labeling instructions, and the initial step, for creating labeling instructions, was to obtain deep knowledge of the conceptual framework, presented in Chapter 2. Then decide on a strategy, which was to utilize available literature and theories within deparadoxification, based on the works of, inter alia, Luhmann (1995) and Andersen (2003). The purpose of choosing this strategy was to ensure a higher reliability and replicability (Krippendorff, 2004) The operational descriptions were created through an iterative process of discussing the research questions, context, and the goal of the labeling. The process started with definitions from Andersen (2003) on temporal, social, and factual deparadoxification. These were used to construct not-relevant as a negation of Andersen's (2003) definitions. In addition, the operational description contained a list of examples, including specific words, which was created for each category. The words and phrases were listed to help navigate the oracles to identify the appropriately categorize the coding units. However, the definitions, as obtained from the relevant literature, are the main source of guidance for labeling. This is because deparadoxification strategies are assumed to depend on context, not just a single word or phrase.

Moreover, the labeling instructions were tested before applied in the initial labeling iteration, to ensure its functionality. This was done by conducting two workshops with the purpose of testing the labeling instructions to ensure conceptual alignment and to test whether the instructions needed further adjustments. These workshops were performed by providing each participant with a spreadsheet of 25 randomly samples paragraphs to be labeled without communicating with the other participants. The first workshop was conducted with three participants, whereas the results showed that 44% of the labeling was perfectly aligned (unanimous agreement), while 52% were partially aligned (one participant disagreed). As a result, the workshop lead to further development of the labeling instructions, before another workshop was conducted to repeat the process. As a result of these workshops, a few improvements to the labeling instructions were made: 1) Examples from earlier studies were added, namely from the works of Knudsen (2006) and Ask et al. (2007). 2) Examples from the *paragraph dataset* were added. 3) Assumptions, as a means of guidance in interpretation of the literature, were added. 4) The categories mixed and not-relevant were added.

Table 5.6, shows the table labeling instructions created for the category, factual. Labeling instructions are found in the Appendix for A2.1, social A2.2, not-relevant A2.3 and mixed A2.4.

	Operational Definitions: Factual deparadoxification
Definition	"Factual deparadoxification is fundamentally a matter of seeing decisions as reactions to "the nature of the case". The best well-known strategy for factual deparadoxification consists in supplying alternatives to choose from. This means that the choice of factual references are decisions as well, that is, decisions about which premises for decisions are defined in order for decisions to become decidable."(Andersen, 2003). "Spatialization acts in the same way to detach the contradiction but through separating the conflicting elements in space rather than time."(Ask et al., 2007) "The most basic form of deparadoxization is the attribution of the decision to a decision maker. In making this attribution, the assumption is made that the decision maker had his or her reasons for choosing one alternative over others (Luhmann, 2018)." Seidl, (2021).
Our Assumptions	When there is presented an alternative, then there is a decision, which is a factual strategy. In other words: When a clear alternative is introduced, or someone considers an option. It is a reaction to the nature of the case. Simply referring to a previous decision is not in itself factual deparadoxification. Referring to previous decisions is only factual if it also implies a future alternative
Word/Phrase List	Words and phrases associated with alternatives, such as: in this context, economies, environment, countries, region, decision, alternatives, option, opportunity, right, preference, agreement, choice,
Observered examples from research	"A proposal sent by the Health Authorities and the memorandum proposes that for the entire health service there should be an independent management. The memorandum lists more than 30 purposes and positive effects by the suggestion, which implies contingency as there are many purposes, meaning that the connectivity to the decision proposal cannot be taken as decided (Knudsen, 2006, p.122).
Examples from dataset	"In this context, the Adria pipeline is of especial importance, not only to the economy of Croatia, but to the economies of other countries of the region. " (UNSC, Resolution 3356) "We therefore welcome the decision the Council is considering today, which would enable the deployment of the S/PV.3326 6 (Mr. Bizimana, Rwanda) second battalion, to be based in the demilitarized zone, which will help consolidate UNAMIR's achievements." (UNSC, Resolution 3326).

Figure 5.6: Factual Labeling Instructions

Mixed As it is possible to communicate several deparadoxification strategies within a single paragraph, the category mixed had to be added. The example below shows how a mixed paragraph contains multiple strategies. This is a perfect example of how intertwined the strategies can be in some cases. By having the mixed category, the authors alleviate potential wrongly labeled paragraphs which is likely to occur if the oracles are forced to

put a clearly mixed paragraph into a discrete category.

> "Ultimately, however, it is only the Government of Croatia and the local Serb party that can breathe life into the Basic Agreement and make it a success. It is therefore right that the draft resolution before the Council stresses the need for them to cooperate fully on the basis of the Agreement and to refrain from any measures that might hinder its implementation. This also holds true for the Government of the Federal Republic of Yugoslavia. On 9 November, the International Criminal Tribunal for the former Yugoslavia charged three officers of the Yugoslav National Army from a Belgrade-based brigade with the mass killing of non-Serb men, who, after the month-long siege and eventual conquest of Vukovar four years ago, were forcibly removed from the Vukovar hospital. This is, in our View, a painful but appropriate reminder of the responsibility that the Federal Republic of Yugoslavia continues to bear for the unresolved situation in Eastern Slavonia. Consequently, the leadership in Belgrade must help actively to settle this question."

The first sentence of the coding unit reflects a social deparadoxification strategy as the decision ascribed points to one specific actor that can make an action. In other words, it is pointing at the Government of Croatia and the local Serb party as the social actors and places responsibility on these actors. The second sentence highlight that there is a need for a decision to be made. It constructs urgency a need for reaction to the gravity of the moment and, therefore, temporal deparadoxification) by communicating that there may be hinders to implementation, referring to the implementation as a future event. The communicator then describes a previous mass killing event, which substantiate the sense of urgency. The last two sentences demonstrate communication that places responsibility, again, on two social actors; the Federal Republic of Yugoslavia and the leadership in Belgrade. In the last sentence, the decision is communicated as it has already been decided. There are only formalities left on what leadership role the leadership of Belgrade will take in this decision.

It is arguable that due to how the strategies might be intertwined, as illustrated in the mixed example, multi-labeling should be applied. However, considering the already high level of complexity of the categories that the classification models will try to learn, having multi-labeling would likely confuse the models. This is especially true given the limited sample size that will be used as training data. Hence, it makes more sense to rather include mixed as a possible category. Considering that the research questions focus on

distinct strategies, as opposed to overlapping ones, employing multi-labeling would not be in the best interest of answering the research questions. The application of mixed category is elaborated on in Chapter 5.6.

Not-relevant The not-relevant category does not represent coding units that are decisively not relevant to this paper, but rather all coding units that does not fit in any of the informative categories. The machine learning models would also benefit from having this category because it ensures that the oracles are not forced to label coding units as deparadoxification strategies, when they in fact are not. If this was the case, the training data would consists of significant noise, making it impossible to train a well performing model. In addition, the last requirement for a set of categories, exhaustiveness, is fulfilled by having a not-relevant category.

5.3.2.3 Data Labeling Tool

For deciding on the labeling tool, the authors had a set of criteria: 1) The tool must be easy to use for the oracles. 2) It should be designed to minimize the number of mistakes to achieve adequate data quality. 3) It should allow for easy cross-checking. The selected tool, created by Lantow (2022), was obtained through the authors' collaboration with EY Denmark. By working with Lantow, the the tool was made available online, using PythonAnywhere as the web hosting service. A few design adjustments were first made to make the tool fit with the intent of labeling paragraphs as deparadoxification strategies. One of the main reasons for choosing this tool, was the direct access to the tool's source code and database, enabling the authors to tailor the tool to this research project in collaboration with Lantow (2022).

The labeling tool has a front-page where the oracles can log into their account with a username and password. Then for the next view, a randomly selected coding unit to be labeled would appear. The categories are represented with buttons on the side of the coding unit. When an oracle clicks on a category, the coding unit is labeled as that category. The information is then stored in a database before the next coding unit appears. If an oracle makes a mistake, there is a button to return to the previous coding unit,

allowing for re-labeling. The tool included a review page used for cross-checking. The review page provides the oracles with the alternatives of "correct" and "not-correct" when going through the coding units labeled by the other oracle. The tool was both used for the initial data labeling iteration and the AL iterations, which are explained in Chapter 5.7.4. All the labels were cross-checked to improve validity and minimize errors.

From Data Labeling to Machine Learning In Chapter 5.3, the oracle qualification list, the labeling instructions, and the labeling tool has been described. These prerequisites allow for the methodology to move on to ML. While ML mainly refers to the actual model building and training, there are a number of activites that first needs to be carried out, namely data selection, data cleaning, initial labeling, and preprocessing. During the initial labeling, the labeling tool will be applied by the oracles, while following the labeling instructions, to label the coding units according to the chosen categories. The initial labeling phase creates the first training dataset which will be used to train the classification model. After the first iteration of training, AL will be used to iteratively increase the size of the training data by labeling more coding units. The following chapter will explain the machine learning pipeline, which is the overall architecture for the machine learning related activities related to answering the reasearch questions.

5.4 Machine Learning Pipeline

While machine learning as a term often refers to a single application of a machine learning algorithm conducted on a chosen dataset, a *pipeline* represents an end-to-end structure that can take training data as input and produce a viable result (Polyzotis et al., 2017). The purpose of this paper is not to create a fully automated pipeline ready to be put into production, but because different models, embeddings, and datasets will be used, creating a pipeline unlocks several benefits. For instance, it allows for more reusability, modularity, and variety (Hapke and Nelson, 2020). Because this paper investigates different models, embeddings, and preprocessing, in addition to doing multiple iterations of training, constructing a pipeline was beneficial as opposed to having many different scripts and code redundancy. In addition, if future research is to be conducted, e.g., to test other models, the same pipeline can be used.

The pipeline was constructed using object-oriented programming (OOP). There are several reasons for why OOP was chosen for this task, namely how OOP allows for better structure, encapsulation, and customization (Lutz, 2010). Structure wise, OOP combines logic and data, avoiding redundancy while making it easier to deal with the different code components. This was important considering that different techniques, such as embedding, was used while handling different datasets and classifiers. In terms of encapsulation, OOP allows for changing method implementation without disrupting the users. For instance, changes to the *Embedding* class (not to be confused with the deparadoxification classes) could be done without erasing or altering already declared instances of the class. Customization was especially important as it was expected that the project would change over time based on the revelation of technical issues and solutions that was not apparent from the beginning. Customization allows for instance classes to be extended, and new subclasses to be added, without breaking code that already works well (Lutz, 2010).

A total of five classes were used in the pipeline (Table 5.5). A class in Python can be considered a template for creating *objects* (Lutz, 2010). The objects are instances of the classes, e.g., a specific person can be an instance of the class Student. An object is, in simple terms, a collection of data and associated behaviors (Phillips, 2010). The classes constructed for this project were *Data*, *Embedding*, *Classification*, *ActiveLearning*, *UpdateDB*, and *Visualize*. For each class a set of functions, called methods, were constructed, in addition to rules for inheritance. A total of 28 methods were defined.

Table 5.5: Machine Learning Pipeline - Classes Overview

OOP Class	Composition	
	Methods	Main Activities
Data	read_data(self) visualize_data_hist(self) concat_prod(self, df_list) preprocessing(self) get_sample(self) data_encoding(self)	Loading, sampling, and preprocessing the data
Embedding	load_model(self, model) text_embedding(self, text) get_embeddings_from_row(self) get_X(self)	Extracting the embeddings to be used for classification
Classification	load_classifier(self, classifier) split_dataset(self, X, y) fitting(self) classify(self) get_proba(self, pred_prob_X) cross_val(self) grid_search(self, params) randomized_grid_search(self, r_params)	Splitting the dataset, model training, predictions, hyperparameter tuning, evaluation
Visualize	plot_results(self, y_test, y_pred) plot_label_dist(self, df_list) precision_rec(self, clf, X_test, y_test, y_score) plot_roc_curve(self, clf, X_test, y_test, y_score) gather_results(self) gather_aggregated(self)	Gathering and plotting the results
ActiveLearning	merge_probas(self, df, arr) prep_sample(self) get_sample(self) transform_to_prod(self)	Preparing next iteration of samples to be labeled by the oracles

Note. OOP classes should not be confused with the models' classes derived from the deparadoxification strategies.

Class inheritance allows for a class to inherit the properties of another class, called the *parent class*. The class that inherits the properties is called the *child class* (Phillips, 2010; Taivalsaari, 1996). Because all classes directly follow each other and each is dependent on the implementation of the preceding class, they were all set to inherit from the preceding class. For instance, *Data* is the parent class of all the other classes, while *Embedding* is the child class of *Data* but also the parent class of the succeeding classes, and so on. For an overview of the different classes and their methods, see Table 5.5 which also shows which chapters that cover the processes happening in which class.

5.5 Data Selection and Initial Labeling

As earlier mentioned, the dataframes selected for this paper is *raw_ docs*, *meta_ meetings*, and *episodes*. From raw_docs, all the columns were selected and further cleaning steps were performed. An important part of data preparation is data cleaning, especially when the labeling is depending on the coding units, in this context a paragraph. The first part of the data preparation was to split the speeches in *raw_ docs* into paragraphs. This was done by splitting each speech whenever there was whitespace in between two pieces of text. The newly acquired paragraphs were stored in a new file under *paragraph_ text* together with the *doc_ id* and an added paragraph counter for each meeting, called *paragraph_ number*. This file contained 600,624 rows, each representing a paragraph. In order to speed up computational time and make the data easier to work with, 50,000 paragraphs were randomly sampled to constitute the *regression dataset*. It was discovered there there were some paragraphs in the *regression dataset* which did not contain any valuable information, e.g., just a single word or short sentence. For this reason, all the paragraphs containing less than 30 characters were dropped from the dataset, resulting in a new row count of 49,946. As visualized in the methodology outline (Figure 5.2), creating the *paragraph dataset* was the last step before the initial data labeling.

During the initial labeling, 921 data points (referring to the paragraphs) were labeled. The data points were randomly sampled from the *paragraph dataset*. Then, the labels were cross-checked by the oracles, resulting in 810 data points which the oracles agreed upon. These 810 labels were added to the *paragraph dataset* under a new column named *correct_ annotation*. Hence the *paragraph dataset* now contained four columns; *doc_ id*, *paragraph_ number*, *paragraph_ text*, and *correct_ annotation*. For the paragraphs that had not yet been labeled, the *correct_ annotation* column contained no value. The *paragraph dataset* was now ready to be used for the next step leading, as visualized in the outline (Figure 5.2), namely preprocessing.

5.6 Preprocessing

The main preprocessing steps that will be completed before the data can be used for modeling, are the crucial activities of tokenization and normalization (Bird et al., 2009). Tokenization, which is especially fundamental, is the process of decomposing linguistic data into smaller units of text, called a sequence of tokens (Song et al., 2021). It is also normally one of the earliest steps in data transformation during a NLP project (Grefenstette, 1999). A token is the technical term for a sequence of characters (Bird et al., 2009). A *type* is the class of all tokens containing the same sequence of characters (Wetzel, 2018). The vocabulary of specific linguistic data is the set of tokens that the data contains, entailing that duplicates are not included (Bird et al., 2009). The reason for tokenization is that most NLP tasks happens at token level, for instance document classification, part-of-speech tagging, and stop words removal are all dependent on tokenization (Bird et al., 2009). In general, having fewer tokens and a small vocabulary size speeds up computation, but it also gives the algorithms less information to work with (**?**)(Chen et al., 2019). There are different types of tokenization methods to choose from, depending on the needs of the task to be performed. As is the case with this study; TF-IDF and BERT prefer different kinds of tokenizers before extracting the embeddings. Therefore, two tokenization processes were conducted independent of each other, one for each embedding technique. Normalization refers to the practice of trying to standardize the text in order to reduce unnecessary information and thereby improving the efficiency of the algorithms (Bird et al., 2009). A simple example is reducing all letters to lower-case. There are also slightly more advanced techniques, such as *lemmatization*, which will be discussed later in this chapter. The following sections will discuss how the tokenization and normalization were conducted, first specifically to TF-IDF and then BERT.

5.6.1 TF-IDF Specific Preprocessing

TF-IDF uses one of the most common form of tokenization, namely splitting the text by spaces. This is done by enumerating through all the labelled paragraphs, splitting them into lists of words using whitespace as the separator. All special signs were removed from the paragraphs using regular expressions. This was done to avoid having special characters attached to the tokens which in this case is meant to only represent words.

The next step was case folding, a normalization technique which ensures that all letters are in lower case. The reason for this is to make it easier for TF-IDF to pick up on the relevance of words without needing to deal with how upper-case letters are used. In general, upper-case letters do not have a large effect on the meaning of words and can therefore be transformed to the benefit of how well TF-IDF attributes the right weights to each word. After case folding, the stop words were removed. Stop words are highly frequent words that does not bear significant meaning on their own, such as articles and pronouns. Stop words removal can be used to ensure that TF-IDF focus on meaning bearing words of importance. However, it is noteworthy that because TF-IDF is designed to assign smaller weights to frequent words, stop words removal does not guarantee that TF-IDF's performance actually improve. However, to further substantiate that TF-IDF should focus on specific meaning bearing words, as opposed to context, stop words were removed as part of the TF-IDF specific preprocessing.

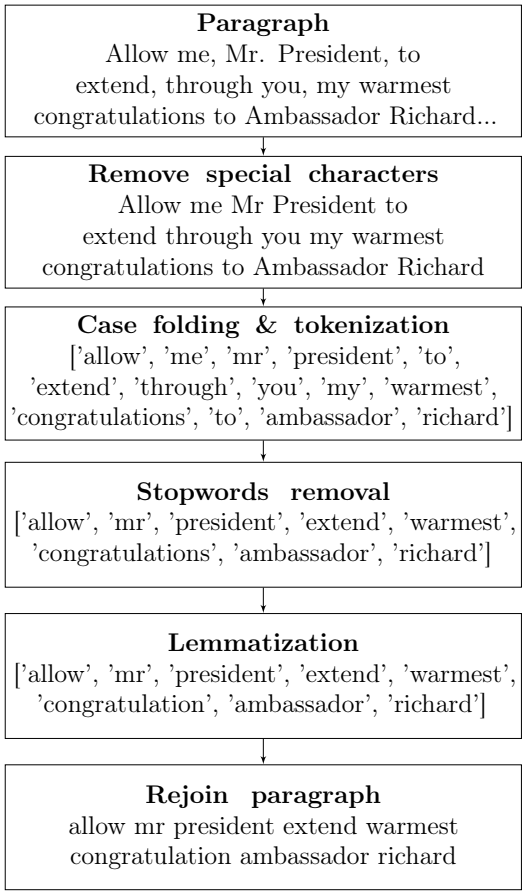


Figure 5.7: TF-IDF Preprocessing

The next step in the TF-IDF preprocessing stage was *lemmatization*, a normalization

technique designed to reduce the forms of the text to a common base form (Manning et al., 2008). The WordNet lemmatizer from the NLTK package was used for this task. WordNet is a vast and well known lexical database containing words and semantic relations (University, 2022). This lemmatizer removes *affixes* only when the resulting word is in its dictionary (Bird et al., 2009). Affixes are word elements that alter the meaning or form of a word. After lemmatization, the tokenization and normalization preprocessing for TF-IDF was complete. Figure 5.7 shows an example of how the different activities altered the text. The example sentence is derived from the dataset.

5.6.2 BERT Specific Preprocessing

The tokenization and normalization for BERT is quite different from that of TF-IDF. This is because BERT is a context based embedding model, as opposed to TF-IDF, and is therefore capable of learning the semantic relations between words. Therefore, it is not suitable to remove stop words for BERT as the stop words adds context to the intention of the communication(Dai and Callan, 2019). As Dai and Callan (2019) highlighted, contextual models achieves a deeper understanding when not removing stop words, due to that stop words provide essential evidence regarding the relevance of the text. All stop words are retained to offer sufficient context information, such as negation words (not, nor, never). This implies that for BERT preprocessing, we wish to retain as much information as possible, while for TF-IDF, a more standardized format is preferable. In addition to not removing stop words, there are two key differences between BERT preprocessing and TF-IDF preprocessing, namely that for BERT, no direct normalization techniques are applied, except for case folding. In addition, BERT uses a special type of tokenizer called the *WordPiece* tokenizer.

The WordPiece tokenizer, specifically the modified BERT tokenizer from the Transformers package by Hugging Face (Face, 2022), does not split the input strings based on whitespace, but on *subwords*. This technique is called subword tokenization. The purpose is to maintain a reasonable vocabulary size while also being able to learn context-independent representations (Song et al., 2021). WordPiece works by using the *maximum matching approach*: Iteratively pick the longest prefix (an affix at the beginning of a word) of the remaining text that matches a vocabulary token until the entire word is

segmented (Song et al., 2021). If a word cannot be tokenized at all, it is replaced with the unknown token [UNK]. The *suffixes* (affixes at the end of a word) are denoted with a double hash sign at the beginning of the token. What this effectively does is allowing BERT to recognize shared meaning between different words that contain similar word pieces. For instance, the terms *skateboarding* and *snowboarding* are different words, but by using the WordPiece tokenizer, BERT can draw similarities between them because each word contains the term *boarding*.

In addition to tokenizing the words using the WordPiece approach, certain special tokens required by the BERT architecture needs to be put in place. These are [CLS] and [SEP]. [CLS] is at the beginning of each sequence (in this case a paragraph) and it is used for classification tasks. It is also the last hidden state of BERT corresponding to the specific token $h_{[CLS]}$, where h = hidden state. The [SEP] token is used to separate the inputs from each other, as BERT only takes in a single long sequence. [SEP] is only relevant for tasks that require multiple inputs, such as question answering tasks, and is therefore not relevant for this project. However, due to the nature of BERT's architecture, it is still necessary to implement the token at the end of each sequence (Devlin et al., 2019). Figure 5.8 shows an example of how the BERT specific preprocessing transform a paragraph into tokens.

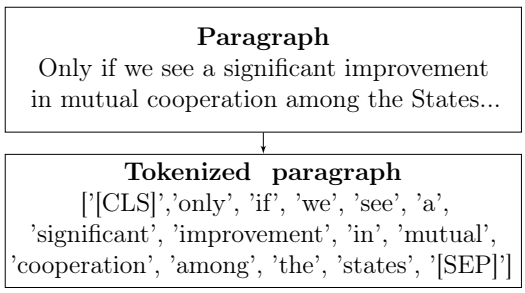


Figure 5.8: BERT Preprocessing

While the example in Figure 5.8 illustrates how the paragraph texts are tokenized before being fed into BERT, it fails to show how the BERT inputs are actually comprised of three arrays: Token embeddings, segment embeddings, and position embeddings (Devlin et al., 2019). The token embeddings are the actual word tokens as visualized in Figure 5.8. The segment embeddings represents whether the token is before or after the [SEP] in the given

sequence. In this case, all the tokens in the segment embeddings are considered to be before [SEP], because the only need for sequence separation is separating the paragraphs from each other. The position embeddings simply represent the position of a token within the current sequence.

5.7 Modeling

This model explains how BERT-RF and TFIDF-RF were build. The first step was to use BERT and TF-IDF to extract the embeddings which was used to train the classification models.

5.7.1 TF-IDF Embeddings

The TF-IDF embeddings were extracted from the lemmatized data where stop words and special characters had been removed. This was done by using the TF-IDF vectorizer from Sklearn's feature extraction module to calculate the TF-IDF weights. This submodule converts a collection of raw texts, in this case the preprocessed paragraphs, to a matrix of TF-IDF features (Pedregosa et al., 2011). Each paragraph, independent of character length, is transformed into a vector of floats in the form of a NumPy array. Each number is a 32-bit float. The length of the array depends on how many features the vectorizer finds. Keep in mind that each paragraph has already been preprocessed as described in Chapter 5.6. Extracting the TF-IDF embeddings only from the labeled paragraphs is a quick computational task taking less than a second. See Appendix A3 for all the runtimes and computer specifications.

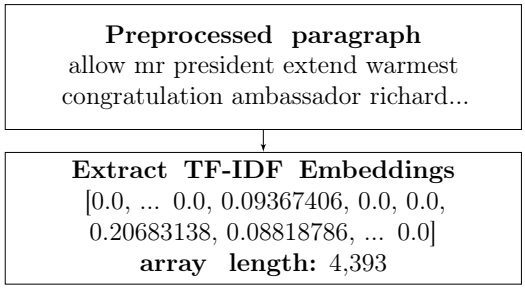


Figure 5.9: TF-IDF Embeddings Extraction

The final list of embeddings contained one array for each paragraph, each with a length of

4,393. An example embedding can be seen in Figure 5.9. The reason for only extracting embeddings from the labeled paragraphs, as opposed to the whole dataset, was that only the BERT embeddings will be used for the AL part (Chapter 5.7.4). Hence, the TF-IDF embeddings for the whole dataset is not needed.

5.7.2 BERT Embeddings

Since its inception, BERT has sparked numerous BERT based or inspired models (Koroteev, 2021). However, due to BERT's broad applications and well established and documented implementations, the original BERT model was chosen for this project. Applying other BERT-based models in further research on machine learning in deparadoxification strategies will be discussed in Chapter 7.3. The original BERT model is in actuality two different models with very similar architecture: *BERT base* and *BERT large* (Devlin et al., 2019). Both models performed substantially better than the state-of-the-art NLP models at the time on a wide range of tasks, with BERT large yielding better results than BERT base on all of the tasks. While BERT base has a similar size of OpenAI's famous GPT model, BERT large has 340M parameters, which is over three times as many as BERT base, which has 110M (Devlin et al., 2019). The reason for choosing BERT base over BERT large was the significant reduction in computational power needed for the embeddings extraction, both in terms of memory and speed, but also storage (Devlin et al., 2019). In addition to choosing between BERT base and large, there exists *cased* and *uncased* versions of both. The difference between these is that the uncased version is not designed to treat cased letter any different from uncased ones. In addition, it discards accent marks. By choosing the uncased model, the authors assume that casing and accent marks does not have a significant impact on the deparadoxification strategies. The reasoning for the choice is that the uncased version is known to perform better than the cased version in scenarios where casing and accents do not matter (**?**)(Ardimento Mele, 2020). If the cased version had been chosen, the preprocessing in Chapter 5.6 would have to be slightly altered for the choice to make sense. The change would have been to remove the case folding step.

The chosen model, *bert-base-uncased*, was loaded from Transformers. This BERT model has been pre-trained on a large corpus of raw text in a self-supervised manner (Devlin

et al., 2019). See Chapter 3.3 for more details on how BERT was pre-trained. The model consists of 12 layers, with 12 self-attention heads and a *hidden size* of 768, meaning that each layer has 768 neurons. Even though random forest when used as a classifier can only be trained on labeled data (Skansi, 2018), the embeddings from all the paragraphs were needed for the AL, which is why embeddings from all the paragraphs (referring to the sample of 49,946) were extracted. The following paragraphs will explain how this was done. Figure 5.10 illustrates the different steps taken.

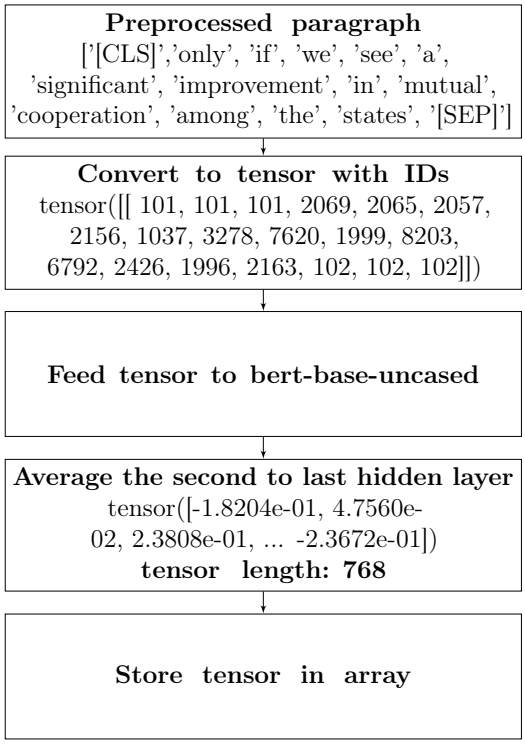


Figure 5.10: BERT Embeddings Extraction

Before the embeddings were extracted, the preprocessed tokens were converted to IDs. Then, they went through a series of steps, one by one. First, they were converted to tensors, which are PyTorch's version of arrays designed to run on either CPU or GPU (Paszke et al., 2019). Second, the BERT model was called for each tensor with its *tokens_tensor* and the *segment_ids* tensor, which represents the segment embeddings. Then, the hidden states from BERT's 12 layers were collected. With these hidden states, there are 13 vectors representing each paragraph. The reason that it is 13 vectors and not 12 (as the number of hidden states implies), is that the first vectors is the input embeddings (Paszke et al., 2019). The length of these vectors is 768, as mentioned in

Chapter 3. To make the embeddings usable for training random forest, a single vector for each paragraph is needed. To achieve this, a *pooling strategy* had to be applied. These types of strategies are all different techniques for extracting single vectors from BERT's hidden layers. The most simplistic approach would be to only take the first or last hidden layer as the single vector. The original BERT paper compared different pooling strategies, concluding that different strategies is likely to yield different results depending on the task. Hence, there are no objectively best strategy in all scenarios. The one chosen for this paper was the second-to-last hidden layer strategy, which performed well in the original paper (Devlin et al., 2019).

After the embeddings of a specific paragraph was extracted using the pooling strategy, the final tensor was appended to an array. When all the paragraphs had gone through this series of steps, the array contained one embeddings vector for each paragraph. The embeddings were now ready to be used for the supervised machine learning stage of this paper. Namely, training random forest to classify the deparadoxification strategies, now contained within the embeddings. The indexes of these embeddings were later used to map them to the correct deparadoxification labels in a dataframe in order to create the training and testing sets. Extracting the embeddings had a runtime of 450m 20s, see Appendix A3 for all the runtimes and specifications.

5.7.3 Random Forest

This chapter explains class encoding, classifier validation, and training for BERT-RF. All the techniques mentioned in this chapter were applied in the same manner for the TFIDF-RF model, but for the sake of avoiding redundancy, only BERT-RF will be discussed in this chapter. However, there is one crucial difference between how BERT-RF and TFIDF-RF were optimized: Even tough they were trained on the same data, only BERT-RF was used for obtaining the training data through AL. The purpose of only using BERT-RF for this was to avoid dedicating half the time to label datapoints specifically for TFIDF-RF, effectively cutting BERT-RF's training data in half. Furthermore, by using AL for both models, it would be more difficult to compare the models, as they would have been trained on different datasets.

While the textual data has up until now been preprocessed and transformed to embeddings, the actual data labels, as provided by the oracles, have not yet been preprocessed. Up until this point, the list of labels has consisted of textual data, namely *not-Relevant, factual, social* and *temporal*, while *mixed* was dropped during preprocessing. However, machine learning models can not understand language, which is why these classes needs to be transformed into numerical data, similar to the motivation of transforming the paragraphs into embeddings (Vanderplas, 2017). In the case of deparadoxification strategies, using discrete numerical values to represent the classes was appropriate, considering that the deparadoxification strategies themselves are labeled and investigated as discrete elements. Therefore, the classes got renamed in the following manner: $(0, 1, 2, 3) =$ (*not-relevant, factual, social, temporal*).

5.7.3.1 Validation, Overfitting & Underfitting

When training supervised machine learning models, such as random forest, there is always a risk of overfitting and underfitting (Daumeé, 2017; Ying, 2019a). Overfitting is an umbrella term used to describe certain unwanted performance drops in machine learning models (Roelofs et al., 2019). In general, the term refers to the case of a model not generalizing well from the observed data to the unseen data (Ying, 2019). This normally happens when the model is too complex and too flexible, resulting in high variance, and therefore learns the irrelevant information, termed noise, within the dataset (IBM, 2021). On the other hand, underfitting often occurs when the model is too simple, or the data is not significant enough to capture the relationship between observed data and its labels. This typically happens when the model has low variance and high bias, implying that it makes strong assumptions about the data (Ghojogh and Crowley, 2019). Overfitting and underfitting is directly linked with the trade-off between variance and bias (Ying, 2019).

In the context of this paper, the observed data is all the UNSC meeting minutes paragraphs that was labeled by the oracles, while the unseen data is all the unlabeled paragraphs. To put it in perspective; at the end of the project, 1,610 out of 616,969 paragraphs were labeled in total. The labeled dataset is considered both small and noisy by the authors. In terms of size, it is difficult to determine prior to a machine learning project what constitutes an appropriate amount of data. However, by looking at other

studies such as Dang et al. (2020) comparative sentiment analysis study, we can get an intuition by analogy. Dang et al.'s (2020) study compared sentiment analysis papers that analyzed relatively short and simple text, such as social media posts and movie reviews, using inter alia TF-IDF embeddings with LSTMs. While achieving good results, most of the datasets consisted of more than 10,000 labeled examples, all the way up to 1.6 million examples. By comparison, the labeled UNSC dataset had 1,610 examples in total after five iterations of labeling (Chapter 5.7.4). The authors assume that classifying deparadoxification strategies in meeting minutes is a relatively hard task to perform compared to sentiment analysis on social media data. Therefore, it is assumed that having 1,610 labeled examples implies that it will be challenging to achieve good results.

The authors assume that there will be a great deal of noise in the dataset, which poses both underfitting and overfitting challenges. This assumption comes from the initial testing of data labeling, where the oracles discovered the difficulties in distinguishing the deparadoxification strategies not only from each other but also from irrelevant information inherent in the text. Having noisy data and a limited dataset size might result in underfitting, as the model might struggle with capturing the relationships between the data and the labels. On the other hand, overfitting is also a risk, as the model might memorize the noise if it is too flexible. Kumar et al. (2020) specifically investigates how BERT responds to noisy data, concluding that BERT's performance on benchmark datasets when performing primary NLP tasks, e.g., sentiment analysis, significantly drops. However, Kumar et al. (2020) mainly investigates noise due to spelling errors, while the BERT-RF model needs to deal with noise in the sense of irrelevant words and sentences. To counteract overfitting and underfitting when training the model, three specific techniques will be utilized: Splitting the data into a training and test set, cross validation, and regularization. Regularization will be discussed in Chapter 5.7.5, while the first two will be explained in the current chapter. In addition to these techniques, acquiring more training data, as done through AL, and hyperparameter tuning also helps with preventing overfitting and underfitting (Daumeé, 2017; Ying, 2019)(Daumeé, 2017; Ying, 2019a). Moreover, random forest itself is a classifier suitable to deal with overfitting, as it help decrease variance while maintaing a low bias (Ghojogh and Crowley, 2019), which makes it particulary stuitable for this case.

Train Test Split For each training iteration during AL, the dataset was split into a training set and a test set. The training set represents the observed data, which will be used to train the model, while the test set represents the unseen data, which will be used to validate the model's performance. The size of the test set was set to 20 %, which is five percentage points lower than sklearn's train_test_split default value. The reason for this choice comes from the works of Kuhn & Johnson (2013), which explains that with small datasets, the model is likely to need as many data points as possible to adequately determine its parameters. Hence, the test set size was set smaller than Sklearn's default size (to increase the size of the training set). Another issue with having a small dataset, as Kuhn and Johnson (2013) points out, is that different test sets might yield different results, due to the uncertainty of test sets in small datasets. For this reason, cross validation was used both during AL, hyperparameter tuning, and the final evaluation.

Cross validation Cross validation is a resampling technique for assessing how well a model generalize to unseen data (Ghojogh and Crowley, 2019). It's basic form, which this paper will use, is called k-fold cross validation () (Refaeilzadeh et al., 2016). In k-fold cross validation, the chosen dataset, in this case the training set, is shuffled randomly before being split into k subsets. Then each subset is sequentially selected as the test set, while the remaining subsets are the training set. For each selection, the model is retrained, and the performance scores are stored. When all the subsets have been used as test set exactly once, the cross validation is over. By averaging the test scores, one can assess how well the model will generalize, without regard to the uncertainty of test sets in small datasets. See Figure 5.11 for an illustration of cross validation.

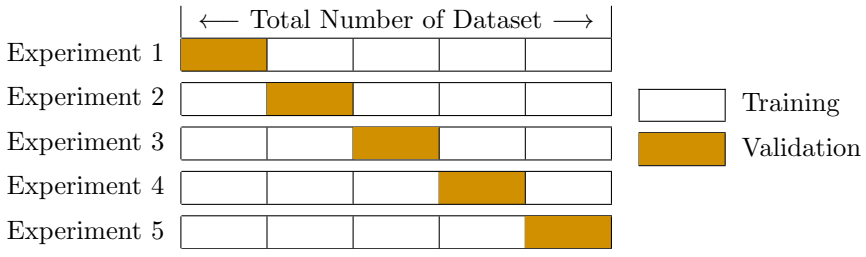


Figure 5.11: Cross Validation Illustration ($k = 5$)

According to Kuhn & Johnson (2013), resampling methods, such as cross validation, can produce reasonable predictions of how well the model generalize. For small datasets they recommend setting using 10-fold cross validation, implying setting k to 10. The small size of the labeled dataset in this thesis project is what led to the decision of following Kuhn & Johnson's (2013) recommendations. By only using cross validation on the training set, and not the full labeled dataset when e.g., tuning the hyperparameters, we can provide an unbiased evaluation of the final model by testing it on the test set. However, due to the insecurities associated with test sets in smaller datasets, cross validation will also be used as a part of the final evaluation (Kuhn& Johnson, 2013). The final evaluation takes place when both AL and hyperparameter tuning has been carried out.

5.7.3.2 Training

The model was trained over two different stages which will be explained in the following chapters, namely AL and hyperparameter tuning (Chapter 5.7.5). In the first stage, the model went through a series of labeling iterations in order to add more training data and improve the model's performance. During this stage, only the default hyperparameter values of Sklearn's *RandomForestClassifier* was used. The reason for not doing hyperparameter tuning for each labeling iteration, and rather wait until after the AL was finished, was to limit the increase in bias the model obtains for each iteration of labeling. If the model is perfectly tuned to for example the dataset obtained from the first iteration of labeling, then the second iteration of data labeling will be more biased towards the model. By not doing hyperparameter tuning for each iteration, the bias is decreased, allowing for a final model that generalize better.

During the second stage of the model training, hyperparameter tuning was conducted using various techniques that will be explained in Chapter 5.7.5. The outcome of the tuning constitutes what will be the final model of this paper. The results of both the AL and hyperparameter tuning will be explained the Chapter 6.

5.7.4 Active Learning

In addition to hyperparameter tuning, AL was used to improve upon the initial results of the BERT-RF model. This was done by iteratively labeling more examples and retraining

the model. For a brief introduction and motivation behind AL, see Chapter 3.5. For a description of how the oracles labeled the data, see Chapter 5.3. The project drew inspiration from the works of Agrawal et al. (2021) and their general AL algorithm, see Algorithm 1, as an overall approach.

Algorithm 1 General steps in AL algorithm (Agrawal et al., 2021)

1: Initially, select few seed data instances randomly from unlabeled train set U
2: Oracle labels the selected data instances and place it in labeled dataset L
3: Train classifier C using labeled dataset L
4: For each data instances in unlabeled train set U use trained classifier C to predict probability for each label
5: Select the most informative data instances from unlabeled train set U using query sampling strategy
6: If stop criteria not reached then go to Step 2 else stop

Where:

$U = Unlabeled\ dataset$

$L = Labeled\ dataset$

$C = Classifier$

While the general algorithm that this project follows is the same as that of Agrawal et al., (2021), there are especially two hidden intricacies that are worth to note:

1. U and L are both in the same dataset, but the absence of labels in U allows us to distinguish them. See Equation 5.1-5.3 for details.

$$U = \{p \mid p\ is\ not\ labeled\} \tag{5.1}$$

$$L = \{p \mid p\ is\ labeled\} \tag{5.2}$$

$$\{U, L\} \subseteq FD \tag{5.3}$$

   Where:

   $FD = Full\ dataset$

   $p = Paragraph$

2. For step 2-5, both U and L are actually referring to the BERT embeddings and not the paragraphs. However, the paragraphs are still needed in order for the oracles to label

them correctly in step 2. This is solved by having a mapping function which maps each paragraph p in FD to an embedding e in $E = Embeddings$ (Equation 5.4). This type of mapping is called *bijective*, because each element in each set is paired with exactly one element in the other set. The function maps FD onto E by using the indexes of the Pandas dataframe containing the full dataset and the indexes of the NumPy array containing the embeddings. For the sake of simplicity, U and L will be used to describe how the AL was carried out, but take note that these are not merely paragraphs, but also references to the embeddings.

$$f : FD \rightarrow E \tag{5.4}$$

AL Algorithm

This subchapter goes through Agrawal et al.'s (2021) Algorithm 1 step by step, to show precisely what actions were taken.

Step 1: "Initially, select few seed data instances randomly from unlabeled train set U." The first seed data instances, referring to unlabeled paragraphs, were selected by first randomly sampling from the full dataset. This amounted to 49,946 examples of meeting minutes paragraphs. Then, instead of selecting which of these instances should be labeled, all instances were shuffled, and labeling goals was set based on how many instances of each class was desirable. The original goal was labelling enough examples so that L contained approximately 100 instances, or more, of each class. The reason for setting a labelling goal, as opposed to simply selecting a sample to be labeled, was that at this point it was impossible to tell whether the dataset was unbalanced or not. Having a sampling goal allowed for the representation of minority classes, in case of imbalance.

Step 2: "Oracle labels the selected data instances and place it in labeled set L." This first labeling iteration resulted in the distribution seen in Figure 5.12. The iteration is called the 0^{th} iteration because the data points were sampled randomly and no AL has yet been conducted.

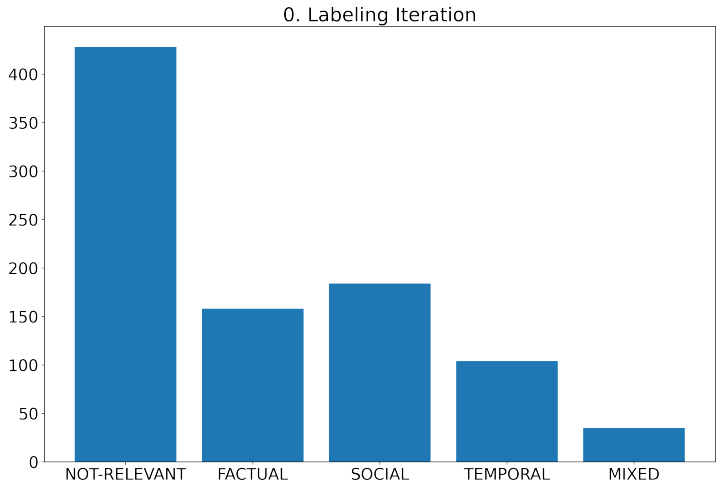


Figure 5.12: 0^{th} Iteration of labeling

The initial labeling resulted in 810 total labelled examples, whereas 364 were *not-relevant*, 184 *social*, 158 *factual*, 104 *temporal*, and 35 *mixed*. It was clear that L was imbalanced, heavily favoring the majority class *not-relevant*. A total of 5 iterations of labeling were conducted (4 of them using AL), increasing the size of L each time.

Step 3: "Train the classifier C using labeled set L." For this step, BERT-RF was retrained during each iteration using the *Classification* class (Chapter 5.4), the same way as it was trained during the first iteration. The default hyperparameters were used as a measure to alleviate bias towards the individual labeling iterations, as opposed to perfectly tuning the model every time. Hence, the tuning was conducted after the AL was finished (Chapter 5.7.5).

Step 4: "For each data instances in unlabeled train set U use trained classifier C to predict probability for each label". During step 4, BERT-RF was used to predict the class of each example in U. In addition, the probabilities of BERT-RF's predictions were stored to be used in the next step. The size of U decreased incrementally from 49,946, as more examples were labeled and hence transferred from U to L. The probabilities of BERT-RF's predictions were retrieved using Scikit-learn's predict_proba(X) method, where X was replaced with U. Because each decision tree in random forest only decides on a single class, random forest computes its class probabilities by taking the mean predicted class of all the trees. For instance, the predicted probability of *factual* for a specific paragraph is calculated using Equation 5.5.

$$P(factual) = \frac{number\ of\ trees\ predicting\ factual}{total\ amount\ of\ trees\ in\ the\ forest} \tag{5.5}$$

Each iterations of step 4 resulted in an array of the same length as number of paragraphs in U, containing decimal numbers between 0 and 1. These numbers, representing the class probabilities, are what will determine the data instances selection in the next step.

Step 5: "Select most informative data instances from unlabeled train set U using query sampling strategy." In step 5, there are many different possible approaches to take. The goal is to select the most informative paragraphs from U. How informative a data instance is, is based on how much it can reduce the uncertainty of a statistical model. In other words, it is how much a classifier can learn from the data instance. In this case it would be how much the embedding of a specific paragraph can reduce the uncertainty of BERT-RF. How the informative data instances are selected is called the *query strategy*. Having a query strategy, as opposed to simply randomly sampling the data instances, in which case calculating the probabilities in the previous step would be unnecessary, is shown to yield better results (Lewis & Catlett, 1994). There exist multiple such strategies, but the one chosen for this project is called least confidence (LC) sampling. LC sampling is an uncertainty-based sampling approach where the data instances to be sampled are the ones which the classifier had the least confidence in while labeling. To illustrate, in step 4, each paragraph in U is assigned four probabilities, one for each class. The highest of these four probabilities represents that class that BERT-RF assigns to a specific paragraph. The most informative paragraphs, according to LC sampling, are the ones where the predicted class has a very low probability. They are the most informative because these are the ones BERT-RF is the most uncertain of. Hence, labeling these paragraphs and retraining the model would reduce BERT-RF's uncertainty more than when labeling paragraphs with a higher predicted probability.

A problem with LC sampling is that due to its definition, it is not very optimal for imbalanced datasets (Aggarwal et al., 2020). Because the majority class might be favored, the initial imbalance might be reinforced, hence reducing the efficiency of the AL. As stated in step 2, the labeled dataset L is already imbalanced, favoring *not-relevant*. To combat reinforcing the imbalance, a modified version of LC sampling was used, drawing inspiration from the concept of *oversampling*, a technique used for changing the training data distribution to better represent the minority classes (Han et al., 2012). In oversampling, data instances are artificially constructed

by replicating the already existing labeled data instances using algorithms such as SMOTE or ADASYN (Chawla et al., 2002; He et al., 2008). Instead of synthetically generating new data points, LC was modified to improve class balance by having a purpose driven approach to the sampling. This was done by sorting the paragraphs in U based on predicted probabilities, as with regular LC sampling, but instead of simply selecting a set of the paragraphs with the lowest class probability (highest uncertainty), the 100 instances with lowest probability of each of the four classes were selected. This approach was partly inspired by Aggarwal et al., (2020) who demonstrated improving class imbalance while querying by preferring the minority class. This was done using Algorithm 2.

Algorithm 2 Modified Least Confidence Sampling

1: Sort the data instances in U based on each data instance's highest predicted probability, from lowest to highest
2: Select the 100 first occurrences of each predicted class in U

By following Algorithm 2, the query strategy successfully sampled the most informative data points, while at the same time alleviating both the class imbalance inherent in the dataset and the potential reinforcement of class imbalance which is associated with using traditional LC sampling. To reiterate the context, Algorithm 2 can be put in place of step 5 in Algorithm 1.

Step 6: "If stop criteria not reached then go to Step 2 else stop". There are multiple ways a stopping criterion may be selected. For instance, the AL algorithm can be stopped when there does not seem to be any improvement in the model with more iterations (Agrawal et al., 2021). Another approach is setting a minimum performance criterion based on a specific metric, such as the F1-score. Based on the scope of this project, demonstrating how the model can be optimized through AL is more important than reaching the perfectly optimized model. Therefore, the stopping criterion was set in the form of number of iterations. Based on the time it took to finish the 0^{th} iteration, the authors set the goal of conducting five labeling iterations in total. The results of the AL, in terms of both performance and final label distribution will be presented in Chapter 6.

5.7.5 Hyperparameter Tuning

Hyperparameters are used configure the different settings of a machine learning algorithm and they can have very varied effects on the model's performance (Claesen& De Moor, 2015).

Deciding upon the right hyperparameters normally happens by either searching manually, by grid, or automatically. A key consideration when chosen hyperparameters is handling the model's complexity. A high level of complexity is likely to result in overfitting, which is when the model learns the data very well, but generalize poorly (Claesen and De Moor, 2015). On the other hand, having too low complexity might result in underfitting, which is when the model fails to capture the underlying patterns inherit in the data. A very complex model exhibits large variance, while a non-complex one is easily strongly biased. Balancing variance and bias are often referred to as the bias-variance trade-off (Vanderplas, 2017). This trade-off is controlled by correctly tuning the hyperparameters (Claesen& De Moor, 2015). The most common methods for hyperparameter tuning are grid search, randomized search, and manual search (Bergstra and Bengio, 2012; Claesen and De Moor, 2015). Grid search evaluates all possible combinations of a set of hyperparameters and their values, which can be costly, but works well if the search space is small and researchers wish to try out every single combination (Géron, 2017). Randomized search evaluates samples of random combinations of hyperparameters for a specified number of iterations. This is beneficial when the search space is large, as random search does not evaluate every possible combination, and thereby saves a lot of time and computational power (Géron, 2017). Manual searching entails selecting hyperparameters for a single iteration of training which is known to be the slowest approach and is easily outperformed by automated approaches (Claesen& De Moor, 2015). To obtain the benefits of both randomized search and grid search, both methods were applied. Figure 5.13 shows an ordered list of how the hyperparameter tuning was conducted.

Step 1: Train model with default hyperparameters (baseline model).

Step 2: Preform wide randomized search using values in Table 5.6.

Step 3: Preform grid search to narrow down search span around the best hyperparameters found in Step 2.

Step 4: Preform second grid search to narrow down search span around best hyperparameters found in Step 3.

Figure 5.13: Hyperparameter Tuning Process

The hyperparameter tuning was conducted using both cross validation on the training set. As Elgeldawi et al. (2021) notes, in addition to using hyperparameter optimization strategies,

choosing the default values which are recommended by software package, in this case Sklearn, is often a relatively safe option as these are carefully based on the literature and experience. However, there is always a risk that the default values does not work well for a certain dataset (Elgeldawi et al., 2021). Taking this into consideration, together with the fact that optimizing all the 18 hyperparameters available to Sklearn's RandomForestClassifier would be too computationally demanding, only seven handpicked hyperparameters will be tuned, while the rest will maintain their default values. The chosen hyperparameters are mainly based on the works of Probst et al. (2019), which considers some of the most influential hyperparameters for random forest as: the number of trees, splitting criterion, sample size, node size, and mtry (number of drawn candidate variables in each split). The splitting criterion decides how the data is split into different nodes as it traverse down each tree. Sklearn provides two splitting criterions to choose from; gini impurity and entropy (based on information gain). Equation 5.6-5.8 shows how these are calculated. See Table 5.6 for an overview of the chosen hyperparameters and their search spans. Sklearn's hyperparameter names will be used for the rest of this paper.

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{5.6}$$

$$Information\ Gain = 1 - Entropy \tag{5.7}$$

$$Gini\ Impurity = 1 - \sum_{i=1}^{n} p_i^2 \tag{5.8}$$

Table 5.6: Hyperparameter Description & Search Span

Hyperparameter	Description & Tuning		
	Description	Total Search Span (A)	Sample Size
n_estimators	Number of estimators/trees	$A_{n\_estimators} = \{ x \mid 10 \leq x \leq 10,000, x \in \mathbb{N}\}$	1,000
criterion	Splitting Criterion	$A_{criterion} = \{gini, entropy\}$	All (2)
max_samples	Percentage of samples from the training data used to train each estimator (if bootstrap = True)	$A_{max\_samples} = \{ x \mid 0.1 \leq x \leq 1, x \in R\}$	10
min_samples_leaf	The minimum number of samples required to a leaf node	$A_{min\_samples\_leaf} = \{ x \mid 1 \leq x \leq 20, x \in \mathbb{N}\}$	10
max_features	Size of subset of features to be considered when splitting a node, relative to the number of features	$A_{max\_features} = \{auto, sqrt, log2\}$	All (3)
max_depth	Maximum depth of each tree	$A_{max\_depth} = \{ x \mid 1 \leq x \leq 50, x \in \mathbb{N}\}$	10
bootstrap	Whether to draw samples from the training data with or without replacement	$A_{bootstrap} = \{True, False\}$	All (2)

Note. Sklearn's notation is used for the hyperparameter names. The sample size represents how many values from the total search span were tried. The values were sampled using an uniform distribution.

An important aspect of choosing the right hyperparameter tuning is regularization (Zhu et al., 2018). Regularization is the concept of adding constraints on a model in order to prevent it from overfitting. As previously discussed, having a limited data set and noisy data increase the risk of overfitting (Tian and Zhang, 2022). By accurately tweaking the correct hyperparameters, overfitting can be, in varying degree, mitigated. For random forest, this tweaking would entail reducing complexity, such as putting constraints on tree growth (Probst et al., 2019). For instance, setting a lower min_samples_leaf leads to larger depth. As Segal (2004) demonstrated, setting smaller tree depth by increasing min_samples_leaf can in some cases be more suitable for noisy data. The max_features is also important for making sure that the model does not overfit. Therefore, in addition to choosing the hyperparameters discussed by Probst et al. (2019), max_depths will also be tuned, as it is one of the main hyperparameters for controlling tree growth, as noted by Sklearn's own description of their RandomForestClassifier (Buitinck et al., 2013). Bootstrap will also be tuned, as it is important for controlling variance by choosing whether or not to use the whole dataset when building trees (Buitinck et al., 2013).

In the first step of Figure 5.13, the RandomForestClassifier from Sklearn, with its default hyperparameter values, was trained to be later used as a baseline model. Having a baseline model allows for measuring whether the hyperparameter tuning positively affected the performance of the model or not. In step 2, randomized search was used to cover a large span of hyperparameters and hyperparameter values, while in step 3 the search was narrowed down by using grid search. In step 4, another grid search was conducted to investigate whether the results of step 3 could be

further improved. For all the tuning activities, the micro F1 score was used as the main metric of evaluation. See Chapter 3.6 for the motivation behind using the micro F1 score. The results of the hyperparameter tuning will be represented in Chapter 6.

5.7.6 Regression Analysis

To investigate whether the deparadoxification strategies have had any affect on the voting outcome, the ordinary least square (OLS) model from the Statsmodels package was applied with voting outcome as the dependent variable and the deparadoxification strategies as the independent variables. This was done by letting the independent variables be the sum of occurrences of each strategy leading up to a resolution vote within a given topic. To achieve this, BERT-RF was used to classify the relevant paragraphs. Because the paragraphs labeled for the classification model were partly randomly sampled and partly obtained through AL (49,946 out of 600,614), they could not be used for the regression analysis. This is because the regression analysis necessitate that the paragraphs are actually relevant for the votes that are held. Sampling random votes and random paragraphs would not make sense. Hence, a new dataset had to be created solely for the regression analysis (the regression dataset), using BERT-RF for the classification task. This was done using the resolution column in the Blaschke (2019) dataset to locate which paragraphs in Schönfeld et al. (2021) dataset should be included and how. The dependent variable (voting outcome) was retrieved from the vote column in the Blaschke (2019) dataset. The outcome comes in the format of *in favor-against-abstaining*, e.g., 14-1-2, where 14 members voted in favor, one against, and two abstaining from voting. In order to use these outcomes for regression, they were transformed to continuous values between 0 and 1. This was done by using Equation 5.9. The series of steps taken in constructing the regression dataset is described in Figure 5.14.

$$\frac{in\ favor * 2 - against * 2 + abstaining}{30} \tag{5.9}$$

Step 1: Left join *topic* from the Shoenfeld dataset on the Blaschke dataset using the meeting id (transformed using RegEx to have the same format), hence assigning a topic to each meeting.

Step 2: Group rows by topic (e.g., "The situation in Afghanistan", "The situation in Cyprus", etc.)

Step 3: Out of all groups containing at least one resolution vote, randomly sample 50 groups (the regression dataset).

Step 4: Left join *paragraph_text* from the Shoenfeld dataset onto the regression dataset.

Step 5: Use the *Embedding* class to retrieve and store embeddings for all the paragraphs (see Chapter 5.4 for the ML pipeline).

Step 6: Predict deparadoxification strategy for each paragraph (using BERT-RF), add predictions to new column *predicted_class*.

Step 7: Sort by [*topic*, *date*, *meeting_id*, *paragraph_number*]

Step 8: For each topic, find the index of each paragraph which is the last one where a unique resolution vote was held, store index in *index list*.

Step 9: For each index in *index list*, sum the occurrences of the different deparadoxification strategies in *predicted_class* leading up to that index from the preceding index (or the beginning of the topic). Store these sums in new columns; *sum_NN*, *sum_F*, *sum_S*, and *sum_T* for each index in *index list*.

Step 10: Only keep relevant rows (retrieved with indexes from *index list*), transform *vote* to continuous values using Formula 5.9, store in new column *points*.

Step 11: Preform regression analysis using *sum_NN*, *sum_F*, *sum_S*, and *sum_T* as independent variables and *points* as the dependent variable.

Figure 5.14: Regression Dataset Construction Steps

After going through the first 10 steps in Figure 5.14, the regression dataset contained five columns (*sum_NN*, *sum_F*, *sum_S*, *sum_T*, and *points*) and 103 rows, where each row represents a unique resolution vote. A total of 402 unique meetings, containing 57,067 paragraphs of 50 different topics lead up to these votes. Due to four rows having faulty voting values, these were dropped, changing the row count to 99. Extracting the embeddings of the paragraphs had a runtime of 530m 46s, see Appendix A3 for all runtimes and computer specifications.

Table 5.7: Regression Dataset Description

Metric	Variables				
	sum_NN	sum_F	sum_S	sum_T	points
count	99.00	99.00	99.00	99.00	99.00
mean	213.42	79.60	114.05	7.26	0.98
std	314.62	228.07	156.30	13.65	0.05
min	7.00	0.00	0.00	0.00	0.63
25%	41.50	2.00	20.00	0.00	1.00
50%	128.00	16.00	60.00	2.00	1.00
75%	265.50	50.50	139.00	7.50	1.00
max	2040.00	1547.00	851.00	87.00	1.00

Table 5.7 display descriptions of the different columns (called variables for the regression analysis) of the regression dataset. To iterate, *points* (the normalized voting outcome) is the dependent variable, while the other four are the independent variables. There are a few interesting observations to be made from Table 5.7. First, *points* has a mean of 0.98 and a standard deviation of 0.05, resulting in a coefficient of variation (CV) of approximately 5.1%. This indicates a very low dispersion around the mean, which entails a low variance in how the different UNSC members vote on resolutions. In fact, around 95% (two standard deviations) lays within 0.93 points. This distribution is visualized in Figure 5.15.

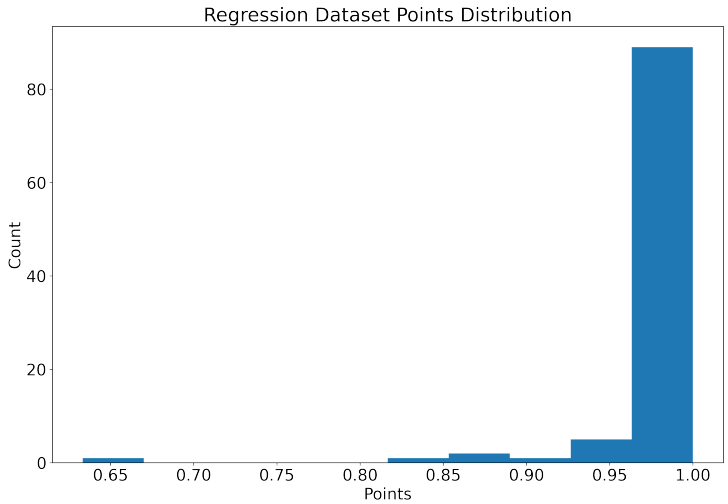


Figure 5.15: Regression Dataset Points Distribution

Furthermore, all the independent variables have very high levels of dispersion by looking at both the CV. Based on this, in combination with the low dispersion of *points*, it was assumed that it

was unlikely that the regression model would find any correlations between independent variables and the dependent variable, but this remains to be seen. The results of the regression analysis (step 11 in Figure 5.14) will be presented in Chapter 6.

6 Results

The purpose of this chapter is to display the results obtained by following the methodology as described in Chapter 5. First, the class distribution obtained through the labeling iterations will be presented, followed by the change in performance over each iteration. Then, the effects of hyperparameter tuning will be presented, followed by the evaluation of the model's performance. Following the evaluation, the predicted distribution and regression analysis results are displayed. The findings and their implications will be discussed in Chapter 7. The last part of the chapter includes the partial regression plot and word clouds for both the labeled and predicted distributions.

Then the predicted distribution will be presented, and lastly, the regression analysis. Finally, the results will be discussed in Chapter 7.

6.1 Active Learning Iterations

6.1.1 Data Labeling

A total of 1,610 paragraphs were labeled over five iterations. During the 0^{th} iteration, 810 paragraphs were labeled from random sampling, while 800 additional paragraphs were labeled through active learning from the 1^{st} to 4^{th} iteration (Table 6.1).

Table 6.1: Labeling Iterations

Iteration	Labeled Paragraphs	
	Non-Cumulative	Cumulative
0	810	810
1	200	1,010
2	200	1,210
3	200	1,410
4	200	1,610

Note. The 0^{th} iteration was conducted using random sampling, while rest used active learning.

Figure 6.1 shows the class distribution after the 0^{th} labeling iteration. It is clear that the dataset is imbalanced. Nearly half of the paragraphs (0.47%) were considered *not-relevant*.

Factual and *social* are relatively similar in size, while *temporal* is the least represented strategy. *Mixed* consisted of only 35 paragraphs.

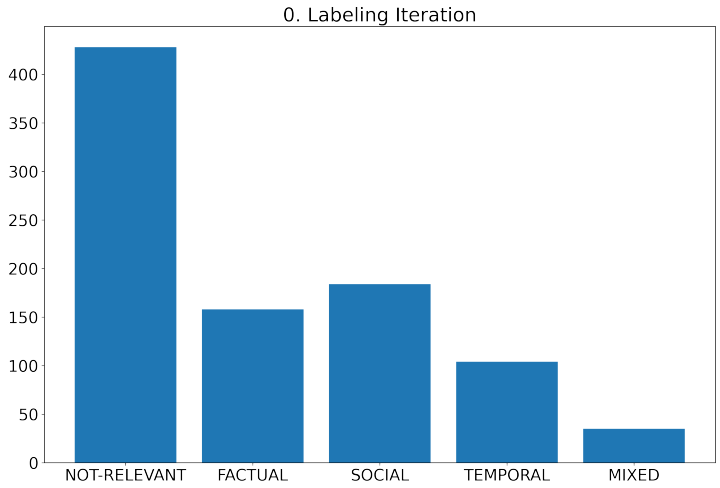


Figure 6.1: 0^{th} Labeling Distribution

Figure 6.2 shows the class distribution over the four iterations of active learning. Interestingly the two first iterations are relatively similar but different from the last two iterations, which are also similar themselves. *Factual* stayed proportionally constant, while *not-relevant* decreased for each iteration. *Social* increased for each iteration, while *temporal* decreased. *Mixed* increased for the first three iterations, but dipped slightly in the last.

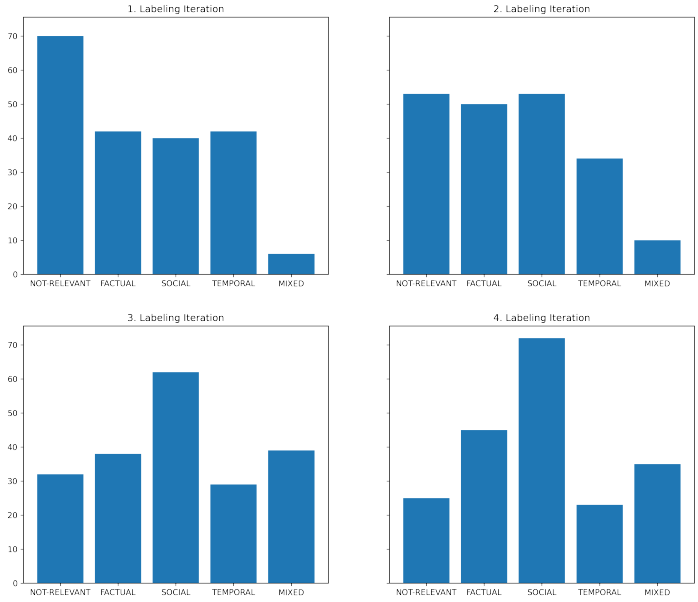


Figure 6.2: 1-4 Labeling Distributions

Figure 6.3 displays a side-by-side comparison of the 0^{th} labeling iteration and the final dataset, with all the labeling iterations aggregated. Again,the distributions are relatively similar, however *not-relevant* noteworthy decreased proportionally.

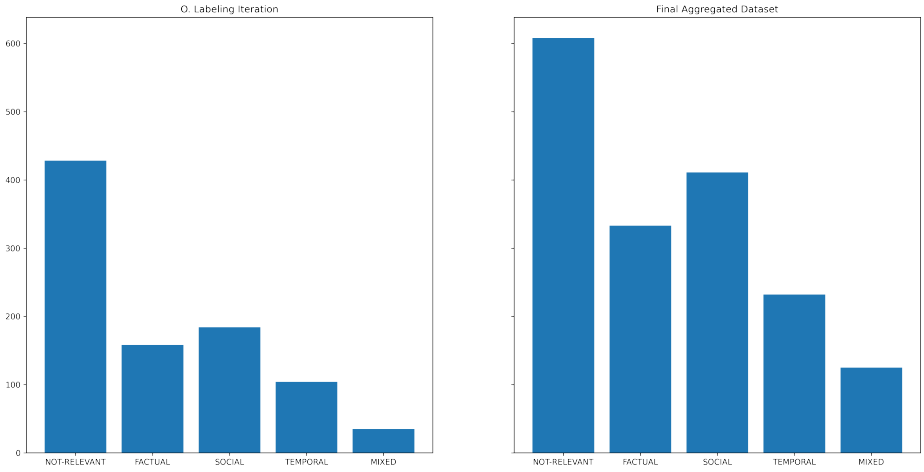


Figure 6.3: 0^{th} (left) and Aggregated Labeling Distributions

6.1.2 Change in performance

Micro F1 will be the main choice of metric for evaluating the models. See Chapter 3.6 for explanations and motivations regarding the different evaluation metrics.

Table 6.2: BERT-RF Performance over Labeling Iterations

Iteration	F1 Scores	
	Macro	Micro
0	0.47	0.60
1	0.44	0.55
2	0.41	0.51
3	0.41	0.50
4	0.41	0.49

Table 6.2 shows the change in BERT-RF's performance over the labeling iterations. At first sight, it looks like the performance went down for each iteration. Both macro and micro F1 decreased overall by six and 11 percentage points, respectively. The implications of these numbers will be discussed in Chapter 7.

6.2 Hyperparameter Tuning

Table 6.3 displays the result of the hyperparameter tuning. A total of seven different hyperparameters were tuned, see Table 5.6 in Chapter 5.7.5 for the search spans. The random forest baseline contains the default hyperparameters as set by Sklearn. The only hyperparameter were BERT-RF and TFIDF-RF ended up with the same value was *max_features*.

Table 6.3: Hyperparameter Tuning Results - BERT-RF

Model	Hyperparameters						
	n_estimators	criterion	max_samples	MSL	max_features	max_depth	bootstrap
RF Baseline	100	gini	None	1	auto	None	True
BERT-RF Tuned	2900	entropy	disabled	1	auto	39	False
TFIDF-RF Tuned	3780	entropy	None	3	auto	44	True

Note. MSL = min_samples_leaf.

The performance of BERT-RF and TFIDF-RF, both with and without tuning, are

displayed in Table 6.4. Two additional benchmark models were also used, the ZeroR classifier and the uniform dummy classifier (UDC). ZeroR predicts only the majority class, while UDC guesses the predictions based on a uniform distribution. BERT-RF Tuned performed best with a macro F1 score of 0.44 and a micro F1 score of 0.53. The second was the TFIDF-RF Untuned model with 0.35 macro F1 and 0.47 micro F1.

Table 6.4: Model Comparison - Cross Validation

Model	F1 Scores	
	Macro	Micro
BERT-RF Untuned	0.41	0.49
BERT-RF Tuned	0.44	0.53
TFIDF-RF Untuned	0.35	0.47
TFDF-RF Tuned	0.33	0.47
ZeroR Classifier	0.13	0.36
Uniform Dummy Classifier	0.23	0.24

6.3 Model Evaluation

6.3.1 Classification Report

The classification report displays all the chosen performance metrics for the best performing model, BERT-RF Tuned, when preforming predictions on the test set (Table 6.5). The support column display the test set's class distribution. *Not-relevant* had the highest F1 score (0.72), followed by *social* (0.54), *factual* (0.42), and lastly *temporal* (0.19). The reason F1 macro and micro in the classification report are different from the values in Table 6.4 is that cross validation was used for the comparison in Table 6.4, while the classification report only shows the performance on the test set.

Table 6.5: Classification Report - BERT-RF Tuned

Class	Metrics			
	Precision	Recall	F1 Score	Support
Not-relevant	0.66	0.79	0.72	102
Factual	0.51	0.35	0.42	68
Social	0.46	0.65	0.54	86
Temporal	0.43	0.12	0.19	48
micro avg			0.55	304
macro avg	0.52	0.48	0.47	304
weighted avg	0.54	0.55	0.52	304

6.3.2 Confusion Matrix

The confusion matrix displays a side-by-side comparison of the best BERT-RF and TFIDF-RF models. The color intensity signals the number of predictions for each cell. The diagonal pattern in BERT-RF's confusion matrix, from the top left corner to the bottom right, follows the correct predictions. As seen in the TFIDF-RF matrix, the model failed to a larger extent to make correct predictions, especially regarding *temporal* which did not receive a single correct prediction.

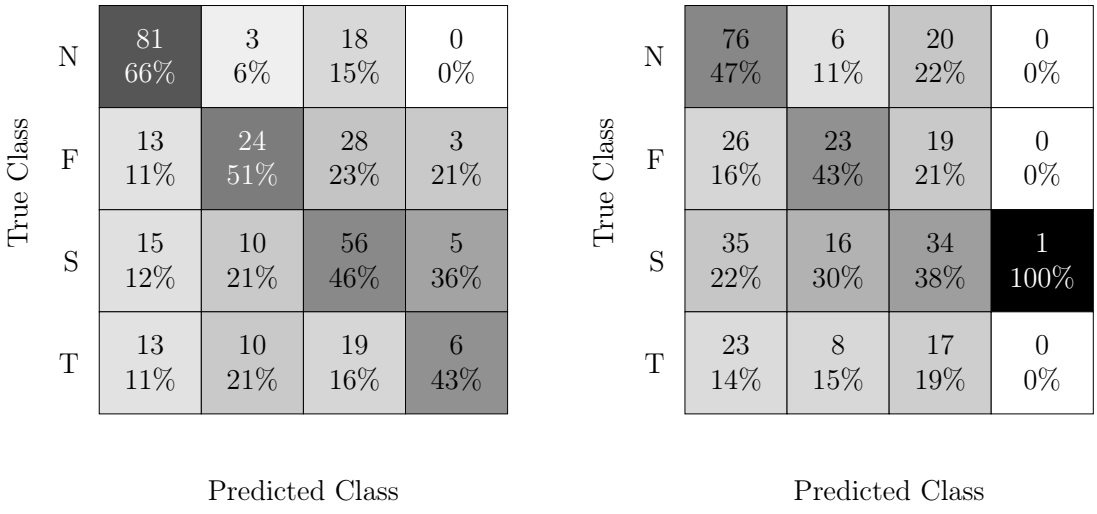


Figure 6.4: Confusion Matrices - BERT-RF (left) & TFIDF-RF
Note. The X-axis contains the strategies in the following order: (N, F, S, T). E.g., the bottom left corner of both matrices are (N,N).

6.4 Predicted Distribution

The predicted distribution shows the class distribution of the 49,496 samples as predicted by BERT-RF (Figure 6.5). It is noteworthy that the model was trained on only 1,810 labeled examples, called the labeled distribution, and that *mixed* was not included during training and hence not in the predictions. Table 6.6 shows the exact numbers for each class.

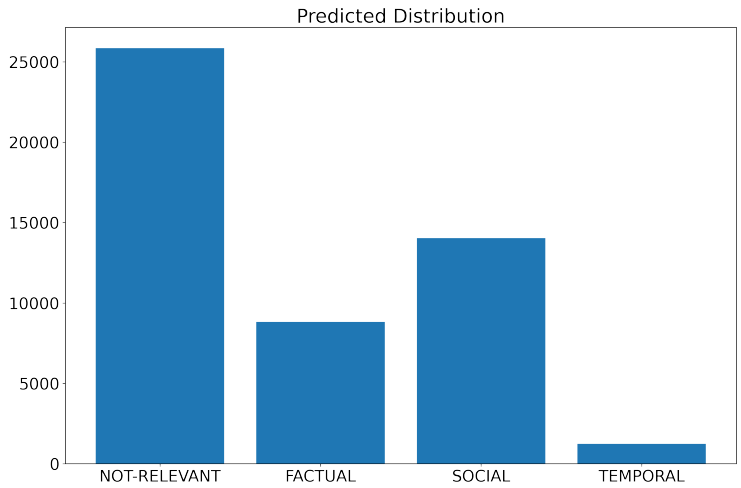


Figure 6.5: Predicted Distribution

Perhaps most interestingly, compared to the labeled distribution, as visualized in Figure 6.3, *temporal* is considerably smaller in size, constituting only 2.4% of the predicted distribution compared to 13.5% of the labeled distribution. *Not-relevant* on the other hand, grew from 35.5% to 51.7%. The implications of the distributions will be discussed in Chapter 7.

Table 6.6: Labeled and Predicted Distributions

Distribution	Classes					
	Not-Relevant	Factual	Social	Temporal	Mixed	Sum
Labeled (Aggregated)	608 (36%)	333 (19%)	411 (24%)	232 (14%)	125 (7%)	1,709 (100%)
Predicted	25,853 (52%)	8,820 (18%)	14,040 (28%)	1,233 (2%)	Not Considered (0%)	49,946 (100%)

6.5 Regression Results

The regression dataset consisted of 99 data points, containing the occurrences of deparadoxification strategies in approximately 57,000 paragraphs, in addition the the normalized voting outcome. The independent variables are the sum of occurrences of deparadoxification strategies (including *not-releveant*) leading up to a resolution vote within a given topic (*sum_NN*, *sum_F*, *sum_S*, *sum_T*). The dependent variable is the normalized voting outcome (*points*). The results of the regression models are displayed in Table 6.7 and 6.8. The most important metrics to note are the r-squared, adj. r-squared, prob (f-statistic), the coefficients and $P >| t |$. These are all highlighted in Table 6.7 and 6.8. The implications of their values will be discussed in Chapter 7. For additional metrics (which will not be discussed in Chapter 7), such as kurtosis and Durbin-Watson, see Appendix A4.

Table 6.7: OLS Regression Results 1/2

Model:	OLS	**Adj. R-squared:**	**-0.024**
Dependent Variable:	points	AIC:	-317.2797
		BIC:	-304.3041
No. Observations:	99	Log-Likelihood:	163.64
Df Model:	4	F-statistic:	0.4175
Df Residuals:	94	**Prob (F-statistic):**	**0.796**
R-squared:	**0.017**	Scale:	0.0022611

Table 6.8: OLS Regression Results 2/2

Variable	Metrics					
	Coef.	Std.Err.	t	**P> |t|**	[0.025	0.975]
const	**0.9826**	0.0062	158.5162	**0.0000**	0.9703	0.9949
sum_NN	**-0.0000**	0.0000	-0.0687	**0.9454**	-0.0001	0.0001
sum_F	**0.0000**	0.0000	0.3212	**0.7488**	-0.0001	0.0001
sum_S	**-0.0000**	0.0001	-0.3834	**0.7023**	-0.0002	0.0001
sum_T	**0.0006**	0.0005	1.0609	**0.2914**	-0.0005	0.0016

The partial regression plot (Figure 6.6) is a set of regression plots showing the effect of each independent variable when the other independent variables are eliminated. In other words, it displays the bivariate relationship between each independent variable and *points*. The implications of Figure 6.6 will be explained in Chapter 7.

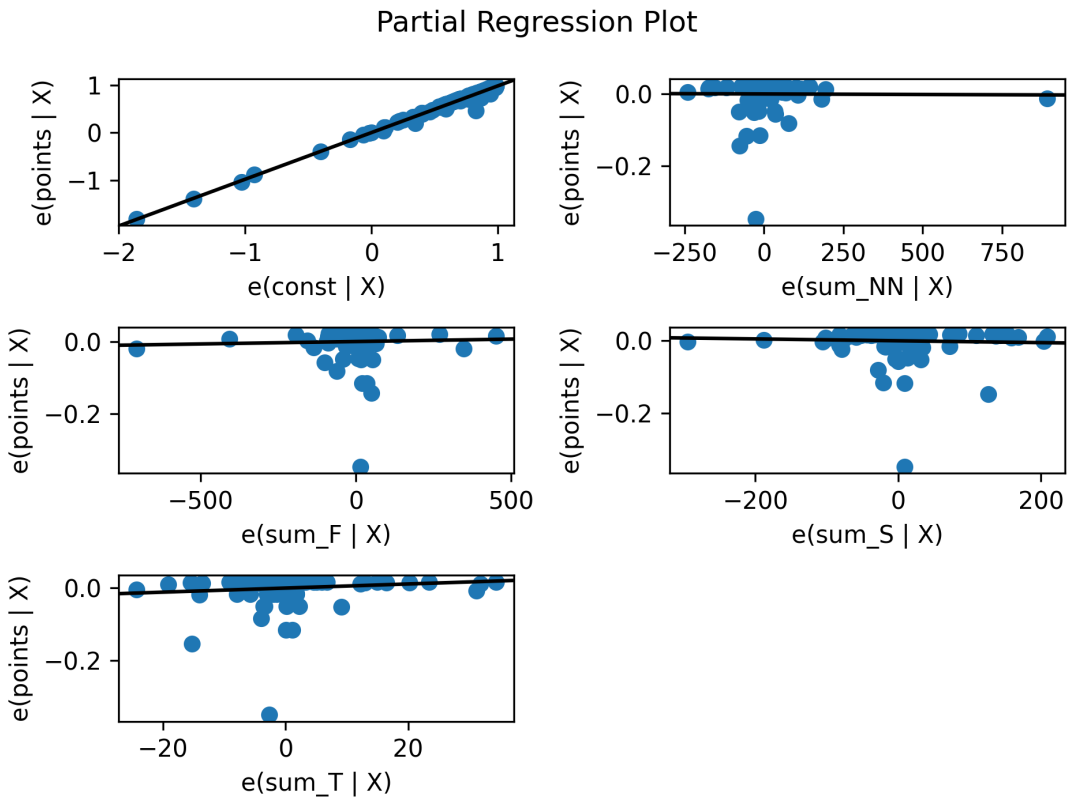


Figure 6.6: Partial Regression Plot

6.6 Word Clouds

The word cloud plots visualizes the 200 most frequent words for each class for both the labeled and predicted distribution (Figure 6.7-6.11). Stop words from the NLTK package was applied to remove stop words. In addition, a few highly frequent words were removed in order to give the reader a stronger impression of the differences: "United Nations", "Security Council", "Council", "United", "Nations", "security", "will", "peace", and "must". The word clouds will not be used in the discussion, except for Figure (*not-relevant*), but are rather meant to give the reader a visual intuition of the data before going into discussion.

(a) Labeled Distribution **(b)** Predicted Distribution

Figure 6.7: Temporal Word Clouds

(a) Labeled Distribution **(b)** Predicted Distribution

Figure 6.8: Social Word Clouds

(a) Labeled Distribution **(b)** Predicted Distribution

Figure 6.9: Factual Word Clouds

(a) Labeled Distribution **(b)** Predicted Distribution

Figure 6.10: Not-relevant Word Clouds

(a) Labeled Distribution **(b)** Predicted Distribution

Figure 6.11: All classes Word Clouds

7 Findings and Discussion

7.1 Answering the Research Questions

7.1.1 Active Learning Influence on Classifier Performance

RQ1: To what extent does the chosen NLP model respond to active learning when classifying deparadoxification strategies?

Initially, 810 samples were labeled based on random sampling, while 488 were later labeled through active learning. It is clear from Figure 6.3 that the label distribution did not substantially change after going through four iterations of active learning, despite the apparent distribution differences in Figure 6.2. The only noteworthy change is that the proportion of Not-Relevant decreased by nine percentage points. These findings have two important implications: Firstly, the oracles have been fairly consistent in their labeling, implying that the labeling instructions were adequately precise and that the labeling process was well designed and executed. Secondly, due to the design of the active learning algorithm (Algorithm 1), the oracles succeeded in making the distribution more balanced by sampling 100 uncertainty cases of each class for each iteration, as opposed to random sampling. As discussed earlier, a more balanced dataset is favorable to the model's performance.

The immediate impression of Table 6.2 suggests that the model's performance got worse from increasing the training data size through active learning. However, this interpretation is not necessarily correct. The authors suggest two reasons, each with their own implications, for the overall decrease in the F1 scores. Firstly, because the dataset of the 0^{th} iteration is substantially smaller than that of the 4^{th}, the chance of overfitting is naturally higher (Ying, 2019). For each iteration, the size of the test set increases, which gives the F1 scores more credibility. Hence, the decreasing F1 scores do not necessitate that the model performs worse. It might simply indicate that the model became less overfitted for each iteration. In addition, the high level of noise further increases the chance of overfitting, which again can be mitigated by expanding the training data (Ng,

2018; Ying, 2019).

While it is plausible that the later iterations mitigated some of the overfittings, hence the decreasing slope, the authors also believe it is plausible that the model's performance did, in fact, not improve over time. The authors hypothesize that the model was initially (0^{th} iteration) heavily underfitted. Having only 810 examples was not enough for the model to learn the underlying patterns of the deparadoxification strategies. This was likely due to the high level of noise and the inherent complexities in detecting deparadoxification. Furthermore, as the oracles experienced; it was often the case that large parts of the text, both at the paragraph level and sentence level, did not directly imply deparadoxification, especially when looking at the parts separately from each other. This made it especially difficult for the model to learn the underlying patterns. Hence, the model was underfit after the 0^{th} iteration. The authors further propose that using the LC sampling technique, as opposed to random sampling, lead to the model querying the outliers before it had established a firm understanding of each class, hence increasing the variance too early. In other words, the model was underfit to begin with. Instead of giving the model time to learn to underlying patterns by increasing the training data through random sampling, the query strategy feed the model the outliers it was the most uncertain about. Hence, the model started overfitting by trying to learn the outliers, which it was fed through each iteration. This resulted in the model becoming partly underfit and overfit at the same time. In short, the model failed to learn the underlying patterns properly, hence underfit, while also being too flexible towards the outliers and noise which it was fed, hence overfit (Ying, 2019) (Ying, 2019). For further reading on simultaneous overfitting and underfitting, the authors recommend the works of Ng (2018). Measures that can be taken to prevent this will be discussed in Chapter 7.3.

In conclusion of RQ1, while it is clear that BERT-RF did respond to active learning, whether and to which degree it had a positive or negative impact calls for a nuanced answer. The apparent decrease in performance can be partly attributed to the model suffering from overfitting at the 0^{th} iteration, however, the authors hypothesize that the model simultaneously suffered from early underfitting (implying high variance and high bias at

the same time). This entails that the model was not able to properly learn the underlying patterns in the 0^{th} iteration, and hence the uncertainty-based query strategy resulted in the model adapting to outliers before it had established a firm understanding of the different deparadoxification strategies. From these observations, the authors conclude that in order for active learning to be suitable for training a model to classify deparadoxification strategies without pre-labeled data, a few adjustments must be made. If enough time and resources are invested into labeling, e.g., increasing the number of iterations twentyfold, random sampling can be used for multiple iterations before transitioning into LC sampling. This would allow the model to learn the fundamentals of each class before being exposed to outliers. It is also possible that LC sampling is not suitable for classes as complex as deparadoxification and that random sampling or some other query strategy should be used instead. Examples of other query strategies are suggested in Chapter 7.3.

7.1.2 Contextual vs. Non-Contextual Embeddings

RQ2: Do contextual embeddings outperform non-contextual embeddings and do they respond differently to hyperparameter tuning?

In terms of hyperparameter tuning, the BERT-RF responded positively in terms of both macro and micro F1 (Table 6.4). Macro F1 increased by three percentage points, and micro F1 by four. Given that the model went through a relatively vast randomized search span, in addition to two grid searches, the authors find it unlikely that the model's performance would increase significantly through further tuning. The tuning resulted in increasing the number of trees from 100 to 2900, implying that adding complexity was favorable to the model's performance. This might be attributed to the complex patterns that make up deparadoxification. Interestingly, the model preferred a decrease in maximum tree depth from *None* (trees expand until all leaves are pure or all leaves contain less than *min_samples_split* samples (Buitinck et al., 2013)) to 39, which lowers the model's variance. Such a regularization implies that the model puts constraints on itself to avoid overfitting. This, in combination with the significant increase in trees and the relatively low overall performance, is a sign that overfitting is causing problems for the model.

The tuning also resulted in changing the splitting criterion from gini to entropy, but considering how similar these criteria are, it does not provide any significant insight into how the model learns the classes (Hastie et al., 2008). The last change from the default Sklearn hyperparameters to those of tuned BERT-RF, was that *bootstrap* was set to *False*, entailing that the main source of variation of the model comes from the random subset of features that are used on each split, which was set to $auto = \sqrt{n\_features}$. This resulted in subset sizes of $27 = [\sqrt{768}]$. Interestingly, the TFIDF-RF model was not able to improve through tuning, hence the untuned TFIDF-RF with the default hyperparameter values was superior, making the preferred hyperparameters quite different from that of BERT-RF. One of the main differences is that untuned TFIDF-RF contains 96.6% less trees than tuned BERT-RF, implying that the best TFIDF-RF model is less complex than BERT-RF. However, untuned TF-IDF has no maximum tree depth, which increases the variance. Nevertheless, based on these observations it is plausible that because the BERT embeddings are significantly more complex than those of TF-IDF, a simpler random forest model is preferred for TF-IDF.

To compare BERT-RF and TFIDF-RF, their F1 scores are plotted in Table 6.4 together with two benchmark models complementary to TFIDF-RF. Those benchmark models' scores are as expected: ZeroR performs better in terms of micro F1 because micro F1 is more suitable in imbalanced cases, and ZeroR only predicts the majority class. UDC performs better in terms of macro F1 because macro F1 works better for balanced data, and UDC acts as if the data is perfectly balanced by having a uniform prediction distribution. Because the dataset is imbalanced and micro F1 is the main choice of metric, ZeroR will function as the main benchmark model (complementing TFIDF-RF). Both BERT-RF and TFIDF-RF significantly outperformed ZeroR. This implied that BERT-RF and TFIDF-RF are able to pick up an existing pattern, hence the deparadoxification strategies can be detected and quantified using machine learning.

Nevertheless whether these patterns are that complex and context dependent that a contextual model is better than a non-contextual model is yet to be answered: Looking at both macro and micro F1 for the tuned and untuned versions, BERT-RF

outperforms TFIDF-RF every time. It is a six percentage point micro F1 difference between tuned BERT-RF and untuned TFIDF-RF. This shows that the contextual model is better, implying that deparadoxification is more complex than simply assessing which words are used, but rather partly inherent in the context. However, the fact that TFIDF-RF managed to achieve a micro F1 score of 0.47 shows that a substantial part of deparadoxification directly manifests itself in the literal words used, which can be detected with disregard for the context. Yet, to achieve a near perfect F1 micro score, through e.g., experiments with substantially more labeled data, the authors assume that considering the context is necessary.

Regarding the performance of BERT-RF, the authors has drawn these conclusions from the classification report: Firstly, the reason *not-relevant* has the highest F1 score (Table 6.5) is due to the natural distinguishability between meeting formalities/etiquette and language where actual problems and decisions are discussed. As the oracles experienced, *not-relevant* was the easiest class to label, which is reflected by the model's performance on that class. *Social* had the highest F1 score out of the three strategies (0.54). This is likely due to both the fact that *social* was the majority strategy in the training data and that the paragraphs in general contained very clear wording implying social deparadoxification, especially when responsibility was attributed to a social actor. *Factual* comes second with a F1 score of 0.42. It was apparent during the labeling that *factual* was the most context dependent class, hence likely posed challenges for TFIDF-RF. It was often the case that it was not before reading the whole paragraph that it became obvious that it should be *factual*. Hence, learning the underlying patterns of such a class can be quite difficult for a machine learning model. In addition, with the large amount of noise that the model has to circumvent, it is not surprising that *factual* has a relatively low F1 score, especially considering the limited training data. Lastly, *temporal* had the lowest F1 score of 0.19. While this is somewhat surprising given that *temporal* paragraphs often stood out by using a set of distinct time oriented words, as described in Chapter 5.3, it is however the least represented class in the data distribution (Table 6.6) and the test set (Table 6.5), which helps explaining the low F1 score.

The confusion matrix (Figure 6.4) further illustrates differences in BERT-RF's and TFIDF-RF's perception of the deparadoxification strategies. Both models experienced *temporal* and *social* to be the most overlapping classes. This likely comes from the models getting confused by paragraphs where a speaker creates some sense of urgency for a social actor to take responsibility for a specific situation. For TFIDF-RF it was particularly extreme: The model only made a single prediction of *temporal*, which happened to be for a paragraph labeled *social*. The fact that TFIDF-RF only predicted one data point as temporal while BERT-RF did it 14 times, six of them successfully, suggest that *temporal* is more dependent on context than the other strategies. Interestingly, there are only two classes which BERT-RF never conflicted, namely *not-relevant* and *temporal*. This implies that *not-relevant* is closer to *factual* and *social* than *temporal*.

In conclusion of RQ2, it is clear that contextual embeddings outperform non-contextual embeddings in this particular case. However, the difference in performance was not large enough to conclude that non-contextual models are obsolete when it comes to deparadoxification. TFIDF-RF preformed surprisingly well compared to BERT-RF. It is apparent that the ratio between the importance of context versus the individual words is not as large as the authors originally assumed. Yet, it is clear that the context does play a role in deparadoxification. With a larger training dataset, the difference between the contextual model and non-contextual model will likely become more evident, as contextual embeddings are proven to outperform non-contextual embeddings when it comes to language containing complex structure (Arora et al., 2020). In addition, using contextual embeddings increase the need for model complexity, which in turn increase the need for more data to improve performance (Ying, 2019). This fact is plausibly why TFIDF-RF performed surprisingly well: Because its embeddings are so simple compared to those of BERT's, that having a small data sample plays to TF-IDF's advantage. In terms of model tuning, only BERT-RF preferred a change of hyperparameters, imposing constraints on itself through regularization by lowering *max_ depth*. In addition, the tuning led to increasing the number of trees, hence reducing variance. These two factors indicate that BERT-RF was more prone to overfitting, as TFIDF-RF did not prefer any regularization. TFIDF-RF kept its baseline parameters, preferring a simpler model in terms of trees, but with higher variance in terms of maximum tree depth. Considering

that BERT contains contextual knowledge of millions of sentences (due to pre-training) and TF-IDF only has knowledge of the paragraphs of which it was build, TF-IDF has a far simpler space for possible combinations to be found, which might explain why TFIDF-RF preferred the simpler random forest model.

7.1.3 Deparadoxification Distribution and Underlying Causes

RQ3: Is there a uniform or otherwise distribution of deparadoxification strategies, and what could be the underlying causes for this distribution?

Both the labeled distribution (Figure 6.3) and the predicted distribution (Figure 6.5) shows that the classes does not follow a uniform distribution. There is a clear majority class, *not-relevant*, in both distributions. The predicted distribution will be the focus of RQ3. For this distribution, *not-relevant* (52%) is followed by *social* (28%), *factual* (18%), and lastly *temporal* (2%). The distribution suggests that approximately half of the communication that takes place in the UNSC meetings is ordinary communication and not decisions. This imply that half the communication is communicating decisions that has already taken place (Schoeneborn, 2011). The authors partially attribute this to the UNSC's natural use of meeting etiquette, formalities, and bureaucratic language, as stated by (Wodak and Krzyżanowski, 2008), and experienced by the oracles in their labeling of *not-relevant*. As seen in Figure 6.10, many of the most frequent words, both for the predicted and labeled distribution, are commonplace meeting etiquette, such as "president", "secretary general", "member", and "representative".

Social had the highest frequency of the three deparadoxification strategies, representing 28% of the total distribution and 58% of the actual strategies. This means that the majority of the non-ordinary communication, as defined by (Schoeneborn, 2011), is making it seem like decisions has already been made and only the formalization is left (Andersen, 2003). Hence, the tension of "us" and "them", which constructs the social space, appears quite frequently in the meeting minutes. It also implies that assigning traits or powers to central social actors is a large part of the communication that takes place. This is likely due to that which the UNSC function in large part by delegating

responsibility to either governments in the form of e.g., ceasefire directives or military action, or regional organizations. In addition, the UNSC has no direct obligation to take responsibility themselves over every security crisis (Bellamy and Dunne, 2016). As discussed previously, there are very few empirical studies on deparadoxification. To the authors' knowledge, the largest one conducted was done by Ask et al. (2007), where the researchers interviewed 25 large Swedish organizations, investigating deparadoxification within IT governance. The study found that *social* was the most common strategy, by a large margin, which is consistent with the UNSC deparadoxification distribution. Even though our study uses a very different case, the domain remains the same. Considering the lack of empirical studies, it is worth to compare our results with this single study. The fact that *social* was also the largest strategy for Ask et al. (2007), might imply that *social* is in general a more frequent strategy in large organizations than the other two. However, more empirical research within this field is necessary to make any such generalizations.

Factual deparadoxification was the second most frequent strategy, representing 18% of the total distribution, and 37% of the strategies. According to Luhmann (1995), factual deparadoxification is the most straightforward one out of the three strategies. This makes it somewhat surprising to see that it is 21 percentage points behind *factual* when only looking at the strategy classes. This might be due to the UNSC being more concerned about the responsibility of social actors, as apposed to different alternatives of action. This implies that the member states are e.g., more likely to point out other member states as responsible and demand that they take action. Or, it might be the case that they prefer to take on responsibility themselves, as opposed to presenting alternatives to the other members. One might assume that the UNSC meetings mainly consists of discussing course of action, but based on the distribution, that does not seem to be the case. Another plausible explanation is that there are studies suggesting that most of the decisions actually take place during informal meetings, where alternatives are presented Eckhard et al. (2021). This will be elaborated on in RQ4.

Temporal deparadoxification is the least frequent strategy in both distributions (excluding *mixed*). For the predicted one, *temporal* represents only 2% of the whole distribution,

and 5% of the strategies. While this might seem surprising giving how the UNSC often deals with seemingly urgent humanitarian crisis in areas heavily affected by conflict, the duration of these conflicts and the time span it takes to reach decisions (voting on resolutions) makes it plausible that temporal deparadoxification is not occurring as frequent as social or factual deparadoxification. However, by looking at the third quartile, we see that 75% of the topics (e.g., "The situation in the Middle East") contains six or less meetings. Based on this, it seems plausible that a sense of urgency should be commonplace in the meetings. However, when taking the outliers into account, e.g., looking at the 30 most frequent topics out of the total 541 topics (derived from *meta_meetings*, see Chapter 5.2.3), they contain 51% of the total meetings, resulting in an average of 98 meetings per topic. Hence the UNSC spends half their meetings on long-lasting issues, which helps explain the low frequency of *temporal*. It is however important to note that *temporal* was the class BERT-RF struggled the most with, as discussed in RQ2. This likely had a significant impact on *temporal's* predicted frequency, which explains why the class is more frequent in the labeled distribution.

Regarding *Mixed*, which was not included in training BERT-RF, grew in proportion to the other classes during the active learning iterations (except for the 4^{th} iteration, where it dipped slightly (Figure (6.2). This is implies that while BERT-RF queried the paragraphs it was the most uncertain about, using LC sampling, the oracles naturally became more uncertain about how they should label them. The fact that for 3^{rd} and 4^{th} iteration, *mixed* was more frequent than *temporal* and *not-relevant* is a clear sign of the frequency of overlap between the deparadoxification strategies within the paragraphs. This fits well with the previously mentioned works of Constantinou et al. (2016) which argue that diplomats tend to use words that are purposefully elastic, in order to suggest alternative meanings to please multiple stakeholders at once. For instance, both suggesting and not suggesting responsibility for e.g., a humanitarian disaster at the same time. This observation carries implications for how difficult it is to train a classifier to predict these intricate and sometimes intertwining patterns of deparadoxification, especially given the lack of sizable training data.

In conclusion of RQ3, it is clear that the deparadoxification strategies does not follow a uniform distribution. This is also true when looking at the ML classes (taking *not-relevant* into consideration). Both the predicted distribution (Figure 6.5) and the labeled distribution (Figure 6.3) are heavily imbalanced. The authors suggests a number of reasons for the distribution. Firstly, that the high frequency of *not-relevant* can be partially explained by the UNSC's natural use of meeting etiquette and bureaucratic language. Secondly, *social* being the most frequent of the strategies can be caused by the extensive use of delegation of responsibility to governments and regional organizations. The authors goes on to compare the frequency with the works of Bellamy & Dunne (2016), which also found *social* to have the highest frequency. Thirdly, *factual* might have lower frequency due to alternatives and course of action being discussed more often during informal meetings, as opposed to the public ones. RQ4 will elaborate on this. Fourthly, the authors suggest that the low frequency of *temporal* is due to the UNSC using half their meetings on long lasting conflicts, which does not tend to create a sense of urgency. Lastly, the surprisingly large frequency of *mixed* in the labeled distribution is plausibly due to a mix of the inherent intricacies in detecting the strategies and the ambiguousness of political language. To iterate an important remark, these observations are based on the predictions of BERT-RF, which entails that the uncertainties of the model makes it impossible to make any decisive conclusions about the class distribution.

7.1.4 Effect of Strategies on Voting Outcome

RQ4: Does the use of any of the three strategies affect the voting outcome of resolutions?

$$\begin{aligned} H_0 &: \sum_{n=1}^{4} |\beta_n| = 0 \\ H_1 &: \sum_{n=1}^{4} |\beta_n| \neq 0 \end{aligned} \tag{7.1}$$

To investigate whether any of the deparadoxification strategies affect voting outcome, we use the F-test of overall significance in regression with an *alpha* value of 0.10. The null hypothesis (H_0) states that the regression coefficients of all independent variables (the deparadoxification strategies) are equal to zero. This entails that any unit change in any

of the strategies does not affect the dependent variable (voting outcome). The alternative hypothesis (H_1) states that at least one of the coefficients are not equal to zero, implying that at least one independent variable explain some of the variation the dependent variable. If the p value of the F-statistic is below *alpha*, we reject H_0, suggesting that the deparadoxification strategies explain some of the variation in voting outcome.

There are five metrics that are especially relevant for assessing this case; r-squared, adj. r-squared, prob (f-statistic), the coefficients, and ($P >| t |$). These can be found in Table 6.7 and 6.8. R-squared, which indicates what percentage of the dependent variable's variance that is explained by the independent variables, is 0.017 indicates that the independent variables (occurrences of different deparadoxification strategies) explain 0.17% of the variation in voting outcome. Adj. r-squared, which adjusts for the number of variables by penalization, had a value of -0.024, implying that the residual sum of squares is close to the total sum of squares. This further substantiate how limited the independent variables are in explaining the variation in the dependent variable. The regression coefficients for all independent variables, except sum_T are 0. This means that for an one-unit shift in either sum_NN, sum_F, or sum_S, the mean of *points* will not change (given that the other independent variables are constant). Sum_T had a coefficient of 0.0006, which implies a minuscule positive expected increase in *points* when sum_T increase. The partial regression plot 6.6 visualize these minuscule or non-existent coefficients by drawing almost perfectly straight lines for each independent variable. One of the most important metrics is the p values from the T-tests. Using an alpha value of 0.10, we fail to reject H_0 of the T-test for each independent variable. This suggest that the results are insignificant, meaning that the individual independent variables explain no variation in the dependent variable.

Finally, by assessing the the prob (f-statistic) (the p value from the F-test) we can asses the joint effects of all the independent variables together, hence performing the F-test of overall significance in regression (Equation 7.1). With an alpha value of 0.10, we fail to reject H_0, given the prob (f-statistic) value of 0.80. This implies that we retain the null hypothesis, which states that sum_NN, sum_F, sum_S, and sum_T does not explain

any of the variation in *points*. Hence, given how the deparadoxification strategies are represented as the independent variables and how the voting outcome is represented as the dependent variable, we conclude that deparadoxification does not affect voting outcome based on the regression analysis.

One of the possible explanations for not finding significant results is the lack of voting outcome spread. As described in the methodology, *points* had a mean of 0.98 and a standard deviation of 0.05, resulting in a CV of 5.1%. Out of the 99 instances of voting in the regression dataset, 79 of them were unanimous (15-0-0). While 10 of them (the second most common result) had 1 vote deviating from voting in favor of the resolution. Intuitively, having an organization that agree most of the time makes it difficult to detect the affects of deparadoxification. However, it could have been the case that on those rare occasions where the outcome was not unanimous, there was a noticeable increase or decline in any of the strategies. However, this turned out not to be the case, based on the statistically insignificant results.

Another possible explanation is that the UNSC wait with initiating a vote until it is clear that all or most stakeholders are satisfied with the contents of the draft resolution. Alternatively, it might be the case that drafted resolutions are in general in the best interest of all stakeholders when it comes to maintaining global peace and security. It is also possible that due to the nature of diplomacy, the representatives already know which side they should take in any conflict of interest, hence the meetings does not necessarily affect how they choose to vote. It might rather be external factors that dictate the voting outcome, as opposed to deparadoxification that takes place within the meetings. As stated by Eckhard et al. (2021), there are multiple researchers arguing that decisions does not actually take place within the UNSC meetings, but rather during informal meetings. Based on this view, the UNSC is simply a talking shop. If that is the case, then the organization is as a means of legitimizing already made decisions, as opposed to making them. Arguably, from a high-level perspective, most of the communication in the UNSC is social deparadoxification according to Andersen's (2003) definition, because most decisions appears as they have already been made and the only thing left to do is formalize the decision. This could explain why approximately 80% of the sampled

decisions are unanimous.

In conclusion of RQ4, with an *alpha* value of 0.10, we fail to reject H_0 of the F-test of overall significance in regression, implying that all the regression coefficients are zero, which suggests that the deparadoxification strategies does not explain any of the variation in the voting outcome. Based on this, we conclude that the strategies, in the form of their occurrences, independent of time and each other, does not affect voting, given the delimitations of this paper and how the strategies and voting outcome are represented in the variables. The lack of statistical significant results might be attributed to a number of reasons, including the lack of voting outcome dispersion, the organization might wait with initiating voting until all or most parties agree, most resolutions might be in the best interest of all parties, or finally, the representatives might have already made up their mind, independent of the actual meetings. If the latter is true, one can argue that all meetings consists of social deparadoxification, as their decisions appear as if they have already been made and formalization is the only thing that is left to do. Even though the F-test resulted in failing to reject H_0, it is important to reiterate that this test was constructed based on the delimitation and set of assumptions of this paper. The outcome of the test only *suggest* that deparadoxification does not affect voting outcome. Not taking the assumptions into consideration would paint a false image of how deparadoxification affects decision making in the UNSC. Not only is it assumed that the independent variables adequately captures the usage of deparadoxification strategies and that the dependent variable adequately captures voting outcome, but in addition it is assumed that BERT-RF adequately classified the paragraphs used for the regression analysis. As discussed in RQ1 and RQ2, BERT-RF did in fact perform rather poorly, with an micro F1-score of 0.53 from cross validation and 0.55 on the test set. By alleviating some of these assumptions, there are numerous ways, yet to be discovered, that deparadoxification can affect voting outcome. This will be elaborated on in Chapter 7.3.

7.2 Limitations

The authors have identified two limitation categories of the paper which should be taken into consideration when evaluating the findings: data limitations and oracle labeling limitations. Attempting to alleviate some of these limitations in future research projects

might yield improved results.

Data Limitations The oracles labeled 1,520 paragraphs in total, which were all cross-checked. While this is a substantial amount given the time intensive labor of labeling paragraphs as deparadoxification strategies, the authors suggest that the classification models' performance can be significantly improved if larger amounts of training data is provided. It is the authors' perspective that the size of the labeled dataset was insufficient to satisfactory train any supervised machine learning model for this specific task. Having twenty or thirty times as much data would expectedly result in a significant increase in performance. Increasing the data size would have allowed the model to establish a firmer understanding of the underlying patterns before querying the paragraphs it was the most uncertain about. Furthermore, the level of noise likely have a significant negative impact on the models' performance, as empirical studies have shown (Gupta & Gupta, 2019). This further increased the negative impact of the small training data size.

For the regression model, there are also noteworthy data limitations. Firstly, the ration between resolutions and paragraphs. Because the dataset that had been used up until the regression analysis had been acquired through random sampling and active learning, it could not be used for regression because it did not contain all relevant paragraphs for each resolution vote. When adjusting the sampling method to what was appropriate for the regression analysis, as explained in Chapter 5.7.6, evaluating 57,067 paragraphs resulted in only 99 resolution data points which could be used in the regression analysis (Table 5.7). This sparsity and lack of resolutions puts heavy constraints on the regression analysis (the UNSC adapts on average around 35 resolutions each year (UN, 2022)). While it is possible to evaluate all the paragraphs in the dataset, it would require running BERT for approximately 100 hours to retrieve the embeddings (based on the runtime of running on 57,067 examples, see Appendix A3). While it could potentially increase the number of resolutions to 1,418 (based on the episodes dataframe), the regression analysis would still be limited by the inconvenience (from a machine learning perspective) that the UNSC tends to agree during voting, as elaborated on in RQ4. This fact limits the possibilities of finding any significant regression coefficients. In addition, it is crucial

to note that, as discussed in RQ4, the data representing the independent variables in the regression analysis were in fact obtained by using BERT-RF, a classifier which only obtained a micro F1-score of 0.53 during cross validation. Hence, the regression dataset is not necessarily a fair representation of the occurrences of the strategies leading up to a vote. In addition, the independent variables only captures the occurrences (as classified by BERT-RF) and takes no other factors into consideration. As a result, it is impossible to decisively conclude that deparadoxification does not affect voting outcome in any way.

Oracle Labeling Limitations While the size of the training data is a limitation in itself, there are also a set of limitations associated with how the data was obtained, namely through oracle labeling. The first one address the fact that the oracles also happen to be the authors of the paper, entailing that both created the labeling instructions and conducted the labeling. This puts constraints on how easy it is to replicate the study as the authors might have obtained an unspoken understanding of the categories (Krippendorff (2004). Such an understanding is not included in the labeling instructions, which can limit their applicability in future studies. However, the authors did apply measures in effort to partly alleviate this limitation, as discussed in Chapter (5.3).

Due to the oracles being human, there is a natural limitation in how they interpret information differently and inconsistently. As Kahneman et al. (2016) states, factors external to the study, such as the oracle's current mood, cognitive abilities, and ability to concentrate over time, may have a significant impact on their assessment of the information that is presented to them. This chance of variability in information assessment is termed as noise (Kahneman et al., 2016). Accordingly, noise is a limitation of the oracle data labeling, which influences the classifier and hence the results. While this kind of noise is difficult to spot, the authors made the design choice of having the oracles cross-check every labeled coding unit in an effort to alleviate the noise. However, the authors find it unlikely that this design choice alone ensured a noise free dataset.

The last limitations are related to bias. Arguably, by including the labelling category mixed, the authors added bias by simplifying a describable phenomenon (Krippendorff,

2004). This implies that the author might have uncertainties about the phenomena of deparadoxification. However, it is also arguable that including mixed as category was necessary because a mixed deparadoxification strategies actually occur within the paragraphs. Hence, the inclusion of mixed does not necessitate an unclear understanding of deparadoxification. Furthermore, while the labeling conducted by the oracles are prone to e.g., noise and other errors, it is important to note the labeling are based on the assumption that the labeling instructions adequately captures Luhmann's concept of deparadoxification. Hence, the authors understanding and ability to formulate the different strategies are in itself a limitation, as the authors are also prone to human error in how they interpret information differently.

7.3 Future Research

The authors of this paper wish to highlight five potential bases for future research that they believe can result in valuable findings, further bridging the gap between data science and deparadoxification. These bases consider data labeling, variations of BERT, unit of analysis, and dynamic Bayesian networks (DBNs).

Data labeling As the results and discussion has shown, while both BERT-RF and TFIDF-RF outperformed the benchmark models by a large margin, active learning and hyperparameter tuning alone was not enough to sufficiently train the classifiers. The authors believe that the micro F1-score of BERT-RF (0.53) can be greatly improved with a substantially larger labeled dataset. To do so, the labeling can be outsourced using crowdsourcing platforms such as Amazon Mechanical Turk to achieve the desired volume. Furthermore, the authors suggest doing multiple iterations of random sampling before using an active learning technique to see how the model responds. In addition, while the LC technique is one of the most popular query strategies due to its computational efficiency and simplicity (Konyushkova et al., 2017), the authors suggest trying other query strategies such as query-by-committee (Seung et al., 1992) or multiple-instance learning (Settles et al., 2008) to investigate whether changing the query strategy significantly impacts the results.

Variations of BERT The pooling strategy chosen in relation to BERT was concatenating the last four hidden layers, which is the strategy that yielded the best

results in the original BERT paper (Devlin et al., 2019). However, as Devlin et al. 2019 demonstrates, there are numerous different strategies to choose from, which is likely to yield different results based on the task. We therefore recommend to experiment with different pooling strategies, such as second-to-last hidden or weighted sum all 12 layers, to investigate how it might change the results. Furthermore, while this paper has utilized the feature-based approach with BERT, there is improvement potential in using the fine-tuning approach instead. It is however noteworthy that this approach would require substantially more computational power (Devlin et al., 2019), especially in regards to active learning and hyperparameter tuning (all the embeddings have to be calculated for each iteration). Lastly, after BERT was released, multiple BERT inspired models have shown promising results such as RoBERTa, StructBERT (Wang et al., 2019) and DeBERTa (He et al., 2020). We encourage experimenting with these models to see whether they can yield improved results to further bridge the gap between machine learning and deparadoxification.

Unit of Analysis The unit of analysis, termed the coding units for labeling, of this paper was the paragraphs. As argued in Chapter , using a paragraph as the unit of analysis is a suitable trade-off between having enough textual context to find deparadoxification, meaning validity, and performing feasible data labeling with oracles, meaning reliability. Paragraphs as the unit of analysis seem to match the trade-off of validity and reliability and do not prefer either validity or reliability over the other. However, as previously stated, the strategies are sometimes intertwined within single paragraphs, hence the mixed category. For this reason, the authors suggest that experimenting with a different unit of analysis might yield interesting results. Furthermore, multi-label classification (as opposed to multi-class) might improve the results found in this paper, given that enough training data is provided. With this approach, the mixed category would be unnecessary, as the oracles would be able to label a single paragraph as multiple strategies.

Dynamic Bayesian Networks The authors suggest using DBNs as an approach for further analyzing deparadoxification within the UNSC after improved classification results have been achieved. DBNs are Bayesian networks which relates different variables to each other over time. DBNs have been applied to a wide span of research fields, including voting behaviour (Costa et al., 2021), but we have not been able to locate a

paper analyzing how UNSC resolutions are connected and influence each other over time. The different paragraphs that makes up the meetings often refer to former resolutions, which are shaped in earlier meetings which also refer to other resolutions. We propose mapping out all the resolutions and meetings in a DBN where each node represents either a resolution or meeting and the edges between them represents the casual relationship established by one node referring to another. The meetings can be assigned values based on the deparadoxification that takes place within each meeting using a classification model, as demonstrated in this paper. This would allow the investigation of conditional deparadoxification dependencies, how deparadoxification impact decisions over time, and how those decisions create and impact future decisions.

8 Conclusion

This paper aimed to contribute to the call for empirical studies on deparadoxification and help bridge the gap between organizational decision-making theory and ML. While deparadoxification has been considered mostly theoretical, the authors sought to investigate how it manifests itself in the real world by training a supervised ML model to recognize the different strategies and measure their impact on decision making. The non-existence of labeled data or earlier studies on this intersection specifically addressing deparadoxification and ML prompted the authors to pick a case study suitable for manual labeling of the strategies, namely the UNSC. By using the BERT model for extracting the embeddings and random forest as the classifier, the authors iteratively labeled 1,610 UNSC meeting minutes paragraphs through active learning with LC sampling. By investigating the use of active learning, the researchers help uncover possible paths to the merging deparadoxificaiton and ML, which suffers from the lack of labeled data.

BERT-RF responded to active learning with a decline in performance from a micro F1-score of 0.60 to 0.49. It is argued that the immediate decline is due to early overfitting being mitigated by increasing the sample size. Yet, the authors hypothesized that the LC sampling technique did confuse the model by feeding it outliers before it had established a firm understanding of the underlying patterns of deparadoxification, hindering BERT-RF in improving its performance. To investigate whether contextual embeddings outperform non-contextual embeddings, BERT-RF was compared with TFIDF-RF. While BERT-RF evidently performed better than TFIDF-RF, TFIDF-RF performed surprisingly similar to BERT-RF, implying that either deparadoxification is not as context dependent as expected, or that BERT-RF did not have enough training data needed for contextual models to adequately learn the contextual patterns. In terms of hyperparameter tuning, TFIDF-RF kept its default values, while BERT-RF preferred imposing constraints on itself through regularization in addition to increasing the number of trees, which indicates that BERT-RF was more prone to overfitting than TFIDF-RF, which is natural given the high complexity of BERT embeddings. Two additional benchmark model were used, ZeroR and UDC, which performed significantly worse than BERT-RF and TFIDF-RF, implying that both BERT-RF and TFIDF-RF did manage to learn the underlying patterns to a certain degree.

The deparadoxification strategies did not follow a uniform distribution. On the contrary, both the labeled and predicted distribution were heavily imbalanced, with the differences being especially prevalent in the predicted one: Not-relevant (51.7%), factual (17.6%), social (28.1%), and temporal (2.4%). The authors suggest a number of reasons for this distribution, including the UNSC's use of meeting etiquette, extensive delegation of responsibility, long lasting conflicts and ambiguous diplomatic language. To investigate whether the strategies affect the voting outcome, the F-test of overall significance in regression was used with an alpha value of 0.10. The test resulted in failing to reject the null hypothesis, suggesting that the independent variables do not explain any of the variation in the dependent variable. This suggests that the occurrences of deparadoxification strategies, independent of time and each other and in the way they are classified by BERT-RF, does not affect voting outcome. The authors suggest that this is due to most voting results being unanimous. It is also hypothesized that the UNSC meetings are in effect, simply a medium for communicating decisions essentially pre-determined, as opposed to making them. This could, to some degree, help explain the lack of significant results in the regression analysis.

The authors suggest that if future studies manage to make up for the significant uncertainties of BERT-RF by training a model on a substantially larger training set, unknown patterns of deparadoxification and their impacts might be discovered. Specifically, the authors recommend increasing the training set size, preferably without the LC sampling technique. Furthermore, trying other variations of BERT, such as RoBERTA (Liu et al., 2019), experiment with other units of analysis, and lastly, using DBNs to investigate how conditional deparadoxification dependencies over time impact decision making. The fact that BERT-RF performed substantially better than ZeroR and UDC proves that it is possible to detect and measure deparadoxification through machine learning, assuming that the labeling instructions and the way in which the oracles execute them, adheres to Luhmann's descriptions. The paper has successfully demonstrated an approach for bringing the otherwise separated fields together, ML and deparadoxification, hence fulfilling its purpose by contributing to answering the call for empirical research on deparadoxification.

References

Aggarwal, U., Popescu, A., and Hudelot, C. (2020). Active Learning for Imbalanced Datasets. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1417–1426, Snowmass Village, CO, USA. IEEE.

Agrawal, A., Tripathi, S., and Vardhan, M. (2021). Active learning approach using a modified least confidence sampling strategy for named entity recognition. *Progress in Artificial Intelligence*, 10(2):113–128.

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Andersen, N. (2013). *Managing Intensity and Play at Work*. Edward Elgar Publishing.

Andersen, N. (2003). The Undecidability of Decision. *In T. Hernes & T. Bakken (Eds.), Autopoietic organization theory: Drawing on Niklas Luhmann's social systems perspective*, (12):235–258.

Arora, S., May, A., Zhang, J., and Ré, C. (2020). Contextual Embeddings: When Are They Worth It? *arXiv:2005.09117 [cs]*. arXiv: 2005.09117.

Ask, U., Bjornsson, H., Johansson, M., Magnusson, J., and Nilsson, A. (2007). IT Governance in the Light of Paradox–A Social Systems Theory Perspective. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 234a–234a. ISSN: 1530-1605.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*. arXiv: 1409.0473.

Basu, R. (2004). *The United Nations: Structure & Functions Of An International Organisation*. Sterling Publishers Pvt. Ltd. Google-Books-ID: IjWMX9nCa0sC.

Bellamy, A. and Dunne, T. (2016). *The Oxford Handbook of the Responsibility to Protect*. Oxford University Press. Google-Books-ID: McaSDAAAQBAJ.

Bergstra, J. and Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. page 25.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*.

Blaschke, M. (2019). The UN Security Council Meeting Records. Type: dataset.

Braathen, P. (2016). Paradox in organizations seen as social complex systems. *Emergence: Complexity & Organization*, 18(2):1–14.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. *arXiv:1309.0238 [cs]*. arXiv: 1309.0238.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]*. arXiv: 1406.1078.

Claesen, M. and De Moor, B. (2015). Hyperparameter Search in Machine Learning. *arXiv:1502.02127 [cs, stat]*. arXiv: 1502.02127.

Cohen, M. D., March, J. G., and Olsen, J. P. (1972). A Garbage Can Model of Organizational Choice. *Administrative Science Quarterly*, 17(1):1–25. Publisher: [Sage Publications, Inc., Johnson Graduate School of Management, Cornell University].

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active Learning with Statistical Models. *Journal of Artificial Intelligence Research*, 4:129–145.

Constantinou, C. M., Kerr, P., and Sharp, P. (2016). *The SAGE Handbook of Diplomacy*. SAGE. Google-Books-ID: PLfeDAAAQBAJ.

Costa, P., Nogueira, A., and Gama, J. (2021). Modelling Voting Behaviour During a General Election Campaign Using Dynamic Bayesian Networks | SpringerLink.

Cunha, M. P. e. and Putnam, L. L. (2019). Paradox theory and the paradox of success. *Strategic organization*, 17(1):95–106. ISBN: 1476-1270 Publisher: SAGE Publications Sage UK: London, England.

Dai, Z. and Callan, J. (2019). Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988.

Dang, N. C., Moreno-García, M. N., and De la Prieta, F. (2020). Sentiment Analysis Based on Deep Learning: A Comparative Study. *Electronics*, 9(3):483. arXiv: 2006.03541.

Daumeé, H. (2017). *A Course in Machine Learning*.

Derrida, J. (1992). Force of Law: The "Mystical Foundation og Authority". In *Deconstructing and The Possibility of Justice*, pages 3–67. Routledge, New York:.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. arXiv: 1810.04805 version: 2.

Eckhard, S., Patz, R., Schönfeld, M., and Meegdenburg, H. v. (2021). International bureaucrats in the UN Security Council debates: A speaker-topic network analysis. *Journal of European Public Policy*, pages 1–20.

Elgeldawi, E., Sayed, A., Galal, A. R., and Zaki, A. M. (2021). Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics*, 8(4):79. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

Face, H. (2022). Hugging Face – The AI community building the future.

Ghojogh, B. and Crowley, M. (2019). The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial. *arXiv:1905.12787 [cs, stat]*. arXiv: 1905.12787.

Grefenstette, G. (1999). Tokenization. In van Halteren, H., editor, *Syntactic Wordclass Tagging*, pages 117–133. Springer Netherlands, Dordrecht.

Géron, A. (2017). HML-Hands-On Machine Learning with Scikit-Learn - TensorFlow.pdf.

Hapke, H. and Nelson, C. (2020). *Building Machine Learning Pipelines*. "O'Reilly Media, Inc.". Google-Books-ID: H6_wDwAAQBAJ.

Hastie, T., Tibshirani, R., and Friedman, J. (2008). *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*.

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328. ISSN: 2161-4407.

He, P., Liu, X., Gao, J., and Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention.

IBM (2021). What is Overfitting?

Jungblut, M. (2017). Between sealed borders and welcome culture: Analyzing mediated public diplomacy during the European migrant crisis. *Journal of Communication Management*, 21(4):384–398. Publisher: Emerald Publishing Limited.

Kahneman, D., Rosenfield, A. M., Gandhi, L., and Blaser, T. (2016). Noise: How to Overcome the High, Hidden Cost of Inconsistent Decision Making. *Harvard Business Review*. Section: Decision making and problem solving.

Knudsen, M. (2006). Displacing the Paradox of Decision Making: The Management of contingency in the modernization of a Danish county. *Niklas Luhmann and Organization Studies*, pages 107–126. Publisher: Liber.

Konyushkova, K., Sznitman, R., and Fua, P. (2017). Learning Active Learning from Data. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Koroteev, M. V. (2021). BERT: A Review of Applications in Natural Language Processing and Understanding. *arXiv:2103.11943 [cs]*. arXiv: 2103.11943.

Krippendorff, K. (2004). *Content Analysis*.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer, New York, 1st ed. 2013, corr. 2nd printing 2018 edition edition.

Kumar, A., Makhija, P., and Gupta, A. (2020). Noisy Text Data: Achilles' Heel of BERT. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 16–21, Online. Association for Computational Linguistics.

Kurbalija, J. and Slavik, H. (2001). *Language and Diplomacy*. Diplo Foundation. Google-Books-ID: yKcHHU2DaPoC.

Li, H. B., Wang, W., Ding, H. W., and Dong, J. (2010). Trees Weighting Random Forest Method for Classifying High-Dimensional Noisy Data. In *2010 IEEE 7th International Conference on E-Business Engineering*, pages 160–163.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. arXiv: 1907.11692.

Luck, E. C. (2006). *UN Security Council: Practice and Promise*. Psychology Press. Google-Books-ID: zyvyR8BsHH4C.

Luhmann, N. (1995). *Social Systems*. Stanford University Press, Stanford.

Luhmann, N. (2006). The Paradox of Decision Making. In *Niklas Luhmann and Organization Studies*. CBS Press and the authors 2006.

Lutz, M. (2010). *Programming Python: Powerful Object-Oriented Programming*. "O'Reilly Media, Inc.". Google-Books-ID: qtdkAgAAQBAJ.

M, H. and M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):01–11.

Malone, D. and Malone, R. D. M. (2004). *The UN Security Council: From the Cold War to the 21st Century*. Lynne Rienner Publishers. Google-Books-ID: iww8h3E8MBMC.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Orallo, J. H., Kull, M., Lachiche, N., Quintana, M. J. R., and Flach, P. A. (2019). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*. ISBN: 1041-4347 Publisher: IEEE.

Nassehi, A. (2005). Organizations as Decision Machines: Niklas Luhmann's Theory of Organized Social Systems. *The Sociological Review*, 53(1_suppl):178–191. Publisher: SAGE Publications Ltd.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv:1912.01703 [cs, stat]*. arXiv: 1912.01703.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Phillips, D. (2010). *Python 3 Object Oriented Programming*. Packt Publishing Ltd. Google-Books-ID: mAy_CffZSDgC.

Polyzotis, N., Roy, S., Whang, S. E., and Zinkevich, M. (2017). Data Management Challenges in Production Machine Learning. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1723–1726, Chicago Illinois USA. ACM.

Press, O. U. (2022). diplomacy.

Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3):e1301. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1301.

Report, S. C. (2019). The UN Security Council Handbook. *Security Council Report*, page 120.

Roelofs, R., Shankar, V., Recht, B., Fridovich-Keil, S., Hardt, M., Miller, J., and Schmidt, L. (2019). A Meta-Analysis of Overfitting in Machine Learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Rosenfeld, R. (2000). Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278.

Schoeneborn, D. (2011). Organization as Communication: A Luhmannian Perspective. *Management Communication Quarterly*, 25(4):663–689. Publisher: SAGE Publications Inc.

Schönfeld, M., Eckhard, S., Patz, R., Meegdenburg, H. v., and Pires, A. (2021). The UN Security Council Debates. Publisher: Harvard Dataverse Type: dataset.

Segal, M. R. (2004). Machine Learning Benchmarks and Random Forest Regression.

Seidl, D. and Becker, K. H. (2006a). *Niklas Luhmann and Organization Studies*. Samfundslitteratur, Frederiksberg, DENMARK.

Seidl, D. and Becker, K. H. (2006b). Organizations as Distinction Generating and Processing Systems: Niklas Luhmann's Contribution to Organization Studies. *Organization*, 13(1):9–35. Publisher: SAGE Publications Ltd.

Seidl, D., Lê, J., and Jarzabkowski, P. (2021). The Generative Potential of Luhmann's Theorizing for Paradox Research: Decision Paradox and Deparadoxization. In Bednarek, R., Pina e Cunha, M., Schad, J., and K. Smith, W., editors, *Interdisciplinary Dialogues on Organizational Paradox: Investigating Social Structures and Human Expression, Part B*, volume 73b of *Research in the Sociology of Organizations*, pages 49–64. Emerald Publishing Limited.

Settles, B., Craven, M., and Ray, S. (2008). Multiple-Instance Active Learning.

Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 287–294, New York, NY, USA. Association for Computing Machinery.

Shearer, C. (2000). The CRISP-DM model: the new blueprint for data mining. *Journal of data warehousing*, 5:13–22.

Sievers, L. and Daws, S. (2014). *The Procedure of the UN Security Council*. OUP Oxford. Google-Books-ID: BstLBAAAQBAJ.

Skansi, S. (2018). *Introduction to Deep Learning - From Logical Calculus to Artificial Intelligence*.

Sklearn (2022). 3.3. Metrics and scoring: quantifying the quality of predictions.

Smith, W. K. and Lewis, M. W. (2011). Toward a Theory of Paradox: A Dynamic equilibrium Model of Organizing. *Academy of Management Review*, 36(2):381–403. Publisher: Academy of Management.

Sohn, Y. (2021). Four pillars of Luhmann's analytical apparatus: Applications for

communication research. *Studies in Communication Sciences*, 21(2):207–224. Number: 2.

Song, X., Salcianu, A., Song, Y., Dopson, D., and Zhou, D. (2021). Fast WordPiece Tokenization. *arXiv:2012.15524 [cs]*. arXiv: 2012.15524.

Sun, D., Wen, H., Wang, D., and Xu, J. (2020). A random forest model of landslide susceptibility mapping based on hyperparameter optimization using Bayes algorithm. *Geomorphology*, 362:107201.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *arXiv:1409.3215 [cs]*. arXiv: 1409.3215.

Taivalsaari, A. (1996). On the notion of inheritance. *ACM Computing Surveys*, 28(3):438–479.

Takahashi, K., Yamamoto, K., Kuchiba, A., and Koyama, T. (2022). Confidence interval for micro-averaged F1 and macro-averaged F1 scores. *Applied Intelligence*, 52(5):4961–4972.

Tian, Y. and Zhang, Y. (2022). A comprehensive survey on regularization strategies in machine learning. *Information Fusion*, 80:146–166.

Turian, J., Ratinov, L.-A., and Bengio, Y. (2010). Word Representations: A Simple and General Method for Semi-Supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden. Association for Computational Linguistics.

UN (1983). Provisional Rules of Procedure (S/96/Rev.7) | United Nations Security Council.

University, P. (2022). WordNet | A Lexical Database for English.

Vanderplas, J. (2017). *Python Data Science Handbook*. O'Reilly.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need. *arXiv:1706.03762 [cs]*. arXiv: 1706.03762.

Wang, L. (2014). Active Learning via Query Synthesis and Nearest Neighbour Search. page 10.

Wang, W., Bi, B., Yan, M., Wu, C., Bao, Z., Xia, J., Peng, L., and Si, L. (2019). StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. *arXiv:1908.04577 [cs]*. arXiv: 1908.04577.

Wetzel, L. (2018). Types and Tokens. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2018 edition.

Whitchurch, G. G. and Constantine, L. L. (1993). Systems Theory. In Boss, P., Doherty, W. J., LaRossa, R., Schumm, W. R., and Steinmetz, S. K., editors, *Sourcebook of Family Theories and Methods: A Contextual Approach*, pages 325–355. Springer US, Boston, MA.

Wodak, R. and Krzyżanowski, M. (2008). *Qualitative Discourse Analysis in the Social Sciences*. Macmillan International Higher Education. Google-Books-ID: PQAdBQAAQBAJ.

Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, 1168:022022.

Zhu, D., Cai, C., Yang, T., and Zhou, X. (2018). A Machine Learning Approach for Air Quality Prediction: Model Regularization and Optimization. *Big Data and Cognitive Computing*, 2(1):5. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

Appendix

A1 Data Descriptions

Table A1.1: Columns in meta_speeches

Column	Column Description
Country	The speaker's nation
Speaker	The speaker's name
Participanttype	The speaker's type of participant
Role_in_un	The speaker's role in the UN
SPV	Meeting number
Basename	ID for the meeting
Topic	Overall topic for the speech
Date	Date format D Month, Yr
Filename	Equal to doc_id
Types	No. of types in speech
Tokens	No. of tokens in speech
Sentences	No. of sentences in speech
Topic 2	The second topic for the speech
Subtopic	Subtopic of the speech
Agenda_item1	First agenda item
Agenda_item2	Second agenda item
Agenda_item3	Third agenda item
Decision	Applicable when a decision has been made

Table A1.2: Columns in meta_meetings

Column	Column Description
Basename	ID for the meeting
Date	Date format D Month, Yr
No. of speeches	No. of speeches for each meeting
Topic	Overall topic for the speech
pressrelease	Link to pressrelease, if available
Outcome	Link to meeting outcome
Year	YYYY format
Month	MM format
Day	DD format

Table A1.3: Columns in episodes

Column	Column Description
Meetingrecord	Id created by the UNSC
date	YYYY-MM-DD
Topic	Overall topic for the meeting
Resolution	Id created by the UNSC, only present when voting
Vote	Voting outcome, e.g., 15-0-0
Text	Id for the meeting
Speaker	No. of speakers per meeting
Duration	No. of minutes

A2 Labeling instructions

	Operational Definitions: Temporal deparadoxification
Definition	"Temporal deparadoxification concerns the definition of decisions as a reaction to the gravity of the moment. In order for a decision to be decided, it has to be necessary, not able to be postponed (Derrida 1992: 26). The moment of decision is always a finally pressing and abrupt moment no matter how much time the decision may allow itself. Within the time dimension, deparadoxification is about the creation of this moment of decision which cannot be deferred even a moment. Colloquially, we are familiar with the problem from phrases like "the moment has come", "the moment is ripe" or the reverse, "the moment is not yet ripe for this decision" (On "the right moment" see also Kirkeby 2000). The decision must be synchronised with "the times"" (Andersen, 2003). "Temporization can be seen as a detachment of the contradiction in time, and hence repackaging the contradiction into a narrative." (Ask et al., 2007).
Our Assumption	If a previous event constructs some kind of urgency in the present, or future, then it is temporal deparadoxification. Always referring to a past or future event. But simply referring to time in itself is not temporal. Creating a sense of urgency by referring to an urgent crisis such as famine, is considered temporal
Word/Phrase List	Words and phrases associated with hurry, such as urgent, need, immediate, hurry, stress, action. Additionally, words and phrases that are referring to time, e.g., wake up, as soon as possible, in a couple of days, present stage, months, years, days etc.
Observered examples from research	"The closer we get to the finishing line the less clear the papers become – in order for people to connect to them. It is a well-known situation; it is often like that. It doesn't make it any easier for the rest of us when we have to follow up on the decisions with the unions – for what exactly have they agreed to?" (Knudsen 2006,p.121-122).
Examples from dataset	"War has not touched it, and only political will is needed to have it operational in a couple of days. " (UNSC, Resolution 3356) "The additional deployments are especially necessary and urgent since the situation in the demilitarized zone and in the north-west of the country remains precarious." (UNSC, Resolution 3326).

Figure A2.1: Temporal Labeling Instructions

	Operational Definitions: Social deparadoxification
Definition	"Finally, social deparadoxification is about making decisions look as if they had in fact already been made so that their formalisation is the only thing left." (Andersen, 2003). "Social deparadoxification can happen through "political analyses" or "interest analyses" of the decision-making situation. By pointing out central players in the environment and attributing them with authority, preferences, and strategies, the decision eventually takes the shape of social imperative." (Andersen, 2003). "The isolation and description of "them" defines "us" as decision-makers. After such "analyses" the decision gradually becomes nothing more than the resolve to either accept or be at the foreground of the inevitable decision. It seems obvious that "we" must reach precisely this decision now while we can still be at the foreground and both be recognised and influence the decision." (Andersen, 2003).
Our Assumption	As if the decision has already been done by pointing to an actor.Placing responsibility on a social actor. Simply expressing gratitude to a social actor is not in itself social deparadoxification. Assigning negative responsibility on an actor is also social deparadoxification as well as a positive and neutral one.
Word/Phrase List	Words and phrases associated with responsibility and social actors, such as responsible, authorities, involved, appeal, efforts, gratitude, the UN, the Government, referring to pronouns, liable, trust, reliable.
Observered examples from research	"The closer we get to the finishing line the less clear the papers become – in order for people to connect to them. It is a well-known situation; it is often like that. It doesn't make it any easier for the rest of us when we have to follow up on the decisions with the unions – for what exactly have they agreed to?" (Knudsen 2006,p.121-122).
Examples from dataset	"Nevertheless, it is important to point out, as the Secretary-General does in his report and as has been pointed out by the Security Council, it is the parties themselves — the Tajik parties — who must fully assume their responsibilities and adopt all necessary measures to strengthen the peace process" (UNSC, Resolution 3382) "With its adoption, all the important recommendations in the report (S/1994/1363) of the Secretary-General will have been acted upon, and the stage will be set to create the necessary conditions to achieve national reconciliation and a political solution to the crisis in Tajikistan." (UNSC, Resolution 3382).

Figure A2.2: Social Labeling Instructions

	Operational Definitions: Not relevant
Definition	"A set of categories that lacks exhaustiveness may be rendered exhaustive through the addition of a new category that represents all units not describable by the existing ones. " (Krippendorff, 2004, p. 132).
Our Assumption	Not all communication is decision making and deparadoxification strategies. Some communication is just regular communication. What is not deparadoxification, belongs in the class of not relevant.
Word/Phrase List	Words and phrases that is not related to any of the other classes.

Figure A2.3: Not-relevant Labeling Instructions

	Operational Definition: Mixed
Definition	Units applicable for two or more deparadoxification classes (Krippendorff, 2004, p. 132)belong to the class mixed.
Our Assumption	"No recording unit may fall between two categories or be represented by two distinct data points." (Krippendorff, 2004, p. 132). Several deparadoxification strategies may occur at the same time.
Word/Phrase List	Words and phrases from temporal, social and factual deparadoxification classes.

Figure A2.4: Mixed Labeling Instructions

A3 Runtimes & Specifications

Table A3.1: Runtimes & Specifications

Activity	Description & Runtime	
	Description	Runtime
Extracting BERT embeddings for BERT-RF	Random sample: 50,000 paragraphs	450m 20s
Extracting TF-IDF embeddings for TFIDF-RF	1,520 paragraphs (labeled)	0m 0.1s
Extracting BERT embeddings for regression model	Selected sample: 57,067 paragraphs	530m 46s
Randomized Search BERT-RF	321m 10s	
Randomized Search TFIDF-RF	218m 32s	
Grid Search 1 BERT-RF	286m 07s	
Grid Search 1 TFIDF-RF	205m 59s	
Grid Search 2 BERT-RF	61m 12s	
Grid Search 2 TFIDF-RF	29m 17s	
Training BERT-RF Model	3m 2s	
Training TFIDF-RF Model	0m 01s	

Note. Processor: Intel(R) Core(TM) i7-10610U CPU @ 1.80GHz 2.30 GHz. Installed RAM: 32,0 GB (31,7 GB usable).

A4 Additional OLS Regression Results

Table A4.1: Additional OLS Regression Results

Omnibus:	139.240	Durbin-Watson:	2.082
Prob(Omnibus):	0.000	Jarque-Bera (JB):	4387.182
Skew:	-4.988	Prob(JB):	0.000
Kurtosis:	34.049	Condition No.:	605