Lead User Identification in Online Forums with Social Media Analytics

Master's Thesis

written by

Daniel Seifert

Anton Schloesser

Handed in: 1st of June 2016

Number of pages / characters: 96 / 171.501

Supervisor: Daniel Hardt, Department of IT Management

Study programs:

Daniel Seifert: Management of Innovation and Business Development - Cand. merc. (EBA)

Anton Schloesser: Business Administration and Information Systems - Cand. merc. (it) (IM)

Abstract

The aim of this thesis is to show how social media analytics facilitates the identification of lead users among members of an online forum.

Our research summarizes and discusses literature from the areas of lead user research and social media analytics and combines both streams of research to analyze user behavior on social media platforms.
We describe different computer-aided methods to identify individuals that possess the specific lead user characteristics ‘ahead of trend’, ‘dissatisfaction’, ‘involvement’, ‘opinion leadership’ and ‘product-related knowledge and expertise’.

In triangulating our findings with the result of a mass screening search we could enhance the validity of our approach.

Table of content

Abbreviations

Acronym	Explanation
API	*Application programming interface*
App	*Software Application*
CRM	*Customer relationship management*
CSS	*Cascading Style Sheets*
DOM	*Document Object Model*
FLUID	*Fast Lead User Identification*
HTML	*HyperText Markup Language*
IDE	*Integrated development environment*
iOS	*mobile operating system by Apple*
JAVA	*Programming Language*
Jsoup	*Java library of methods designed to extract and manipulate data stored in HTML documents.*
NLTK	*Natural Language Toolkit*
R	*R (programming language)*
ROM	*Read only memory*
RSS	*Really Simple Syndication*
SMA	*Social Media Analytics*
SQL	*Structured Query Language*
XDA	*O2 Xda brand was a range of Windows Mobile PDA phones, marketed by O2*
XSS	*Cross-site scripting*

Table of figures

1 Introduction

The IT industry, in particular the online world, is currently experiencing the most massive technological change in the history of computer science. Mobile devices are dominating the market and major developments and innovations are no longer happening on desktop computers and mainframes, but on smartphones and tablet devices. Google and Apple, with their mobile operating systems Android and iOS, are leading the field with a cumulative market share of over 90% (IDC Research 2015).

Since the earliest days of Android, it has been the users that played a crucial role in the development of new features, improving system performance and fixing bugs. One of the major contributors is Steve Kondik who is famous for CyanogenMod, the biggest independent and user-maintained open-source Android version. During an interview with androidcentral.com, he highlighted that major developments in the Android field are user initiated. Furthermore, he explained that a considerable amount of features initially developed for CyanogenMod were adapted by Google for their latest Android release 'Marshmallow'. The past years have clearly shown that experienced users motivated by the possibility of own benefit are greatly involved in the development of new functionalities for the Android ecosystem (Dobie 2015).

In general, user innovation effort is mostly performed by a group of people called lead users. These individuals regularly outperform organizational manufacturers in terms of innovativeness and product success. Due to their unique characteristics, lead users have also become a major research area in the field of innovation management (von Hippel 1986; Schreier, Oberhauser, and Prügl 2007).

Lead users are among the first who perceive high benefits from adopting new products and services and potentially start to innovate if the available offerings do not meet their demands and expectations. Furthermore, they show considerable product knowledge and their individual needs have high potential to becoming general market trends. In addressing their so-called "leading edge needs", lead users apply existing products and services in new ways or create completely novel prototypes and solutions. Researchers have particularly paid attention to ways on precisely characterize these individuals and how to separate them from the majority of market participants. In doing so, several academic papers describe five specific characteristics lead user possess:

ahead of trend, dissatisfaction, product-related knowledge and expertise, involvement and opinion leadership (von Hippel 1986; Schreier, Oberhauser, and Prügl 2007; Morrison, Roberts, and Midgley 2004; Lüthje 2004).

Since the evolvement of the lead user theory, not only academic researchers have been interested in studying lead userness, but also organizations aim to early integrate individuals into corporate development processes. As lead user innovations account for a considerable number of commercially highly successful products and service concepts, it is a promising field of study for corporate enterprises looking for new innovative ideas. The integration of lead users into corporate innovation reduces costs and effort of design and product development. It is further a way to judge new product and service concepts in terms of general market fit, as lead users a said to foresee future market trends (Schreier and Prügl 2008).

In order to identify lead users, several methods for lead user detection have been developed. The most basic and widely discussed approach is mass screening. This rather quantitative method is based on parallel scanning of huge data sets for distinctive user characteristics. Other intensively studied approaches are pyramiding and broadcast search. These methods aim at tapping into knowledge pools residing outside of a pre-defined search domain. The ultimate goal of all lead user search techniques is to identify individuals showing the distinctive lead user characteristics (von Hippel, Franke, and Prügl 2009).

Reviewing the most recent studies and literature, one can see that the attention on theory and practical guidelines for lead user discovery is again significantly growing. This trend can be traced back to the latest developments on the internet.
Throughout the last decade, social media has been growing exponentially as a form of communication, where users are invited to actively engage in creating, sharing or rating of online content. Due to its massive reach, speed and ease of use, social media has been changing how society communicates publicly. Furthermore, it has been setting new trends in a variety of topics, ranging from fashion, politics to entertainment and technology. Social media sets the ground for novel approaches in analyzing human communication and interaction. Not only science is interested in the new insights gained by exploring user generated content, also firms started to use it as a new way for communicating with their clients. Today, companies are not only actively

engaging in online discussions but also use social media analytics to better understand their customers and new market or product trends (Stieglitz et al. 2014).

Also in the area of lead user research, studies show that academics increasingly consider social media as a valuable field for investigating lead user behavior. As user generated content represents interests, opinions, knowledge and other forms of human expressions, it is a promising field in the search for lead user characteristics. The latest publications show that it is the beginning of new qualitative and quantitative approaches for lead user discovery. One stream of new lead user literature focuses on machine learning to analyze text and user behavior in online communities. Researchers extracted data from Twitter and ran several machine learning and other computer aided algorithms in order to identify innovative users (Pajo et al. 2013; Pajo et al. 2015).
Other studies concentrate on more qualitative methods and brought tools from social science to the lead user research (Belz and Baumbach 2010).

Throughout this paper, we will describe and discuss a novel computer aided approach for lead user detection in online communities. Our goal is to transfer already developed knowledge and findings to new circumstances as well as to enrich this field of science with new ideas and procedures. Traditional lead user research has often been described and criticized as being too costly and time consuming, as the techniques for identifying these individuals require massive manual input and tasks to complete. Our approach and developed tools should provide academic and organizational researchers with new ideas and insights in simplifying lead user detection. Furthermore, it aims at speeding up the process as well as reduces the amount of manual work

1.1 Motivation

Our thesis is motivated by two distinct streams of research that have both gained notable attention. First, lead user theory and second, social media analytics.
While the origin of the lead user theory leads back to the mid-80s, social media analytics is a relatively new academic field. The incredible growth of social media throughout the recent years demanded a way on how to process the significant amount of data published on these platforms. Social media analytics aims at solving this issue by combining, extending, and adapting methods for exploring social media data (Stieglitz et al. 2014).

Not only academics started to research social media. There is an increasing number of businesses adapting methods and tools to gain new customer insights as well as to take an active part in the user conversation. Having the knowledge that user innovations are highly attractive, companies also started to engage in social media in order to to receive new ideas and inspirations. Social media platforms like NineSigma[1] allow businesses to post problems and invite users to solve them.

Even though business and academic research has already recognized the use of social media in the field of lead user study, we noticed that there have only been few attempts to use social media analytics for detecting innovative users. Pajo et al. (2013, 2015) were the first academics to introduce machine learning and other automated computer-based techniques to lead user research. However, the application of their approach has been limited to Twitter.

Our aim and motivation is to extend research on lead user theory as well as social media analytics. We describe how computer-aided methods can be successfully applied to online forums. Belz and Baumbach (2010) have already found evidence for lead userness on these social media platforms, but they made use of a manual netnography approach for detecting those individuals.

We extend their research on lead users in online forums by introducing a novel computer-based approach for detecting these valuable individuals. We aim at showing the successful application of different social media analytics tools to lead user research by combining the well-studied methods in unique ways.

1.2 Research question

This thesis wants to understand how lead users can be identified in online forums by making use of different techniques from the field of social media analytics. Our goal is to merge the two

[1] NineSigma helps organizations in the public, private and nonprofit sectors 'connect with the world' to find new solutions, knowledge and partners to accelerate their innovation cycle

main concepts of lead user theory and social media analytics in a newly developed approach. Therefore, our thesis was initially guided by following question:

How do methods of social media analytics facilitate the identification of lead users in online forums?

Throughout the course of our research, we recognized that there are especially five distinctive attributes characterizing lead users. By identifying these characteristics among the selected user population, lead users can be separated from the rest of the market. Thus, we decided to further narrow down our research question:

How do methods of social media analytics facilitate the identification of lead users in online forums by analyzing their specific characteristics?

1.3 Delimitation

In order to receive highly valuable and reproducible results, we set clear boundaries for our study and research framework.

First, we only focused on the area of Android development as we observed that users in this field possessed a high affinity towards online forums and social media in general. Furthermore, we recognized that major Android developments were initiated by users.
Second, we narrowed down our research domain to one particular forum, namely the xda-developers community. This ensured the accessibility to a high volume of valuable recent social data. Besides that, even though our analytical tools were compatible with other online forums, a broader research domain would have been too big for the scope of a Master's Thesis.
Furthermore, we did not process large data sets categorized as 'Big Data', as we were limited by the hard disk space and calculation power of a personal computer.

Regarding our literature review, we did not aim to draw an exhaustive picture of the multiple existing concepts, methods and tools that are available in the information systems domain. We rather focused on presenting chosen analysis methods and machine learning applications that were applied by our tool.

1.4 Structure of thesis

This thesis is divided into six chapters to provide a solid understanding of the presented topic as well as precisely explain on how we approached and conducted our analysis.

We started showing the relevance of the selected research field and described our motivation towards the topic. Furthermore, this chapter explained the delimitation of our study.

Chapter two addresses the methodological approaches and choices applied for conducting the research as well as describe the conducted data collection.

Chapter three gives a theoretical introduction to the field of lead user research and social media analytics. As lead userness and methods of lead user detection play a crucial role for our analysis, we focus on theory that discusses the different characteristics lead user show as well as describe different approaches of identifying these individuals. In the context of social media analytics, our explanations particularly portray theory in the area of machine learning methods.

In chapter four we give more context about our research domain, Android development and the Android online community xda-developers.

The fifth chapter contains the central part of this thesis. In order to identify lead users amongst the xda-developers community, we present a novel computer aided approach for unveiling lead users. Furthermore, we show how our findings could be verified by applying a traditional lead user search technique.

Chapter six revisits major findings in an analytical discussion. We draw conclusions on theory, practical findings and the used methodology. Furthermore, limitations of the investigation are addressed and further research is suggested. Finally, implication for management are provided tailored to the case study of this thesis.

2 Methodology

In order to lead through the applied methodology for this thesis, we make use of the 'research onion' diagram introduced by Saunders, Lewis, and Thornhill (2015). We describe the underlying philosophy, research approach, strategy, time horizon, and finally the applied techniques that enabled us to conduct our analysis.

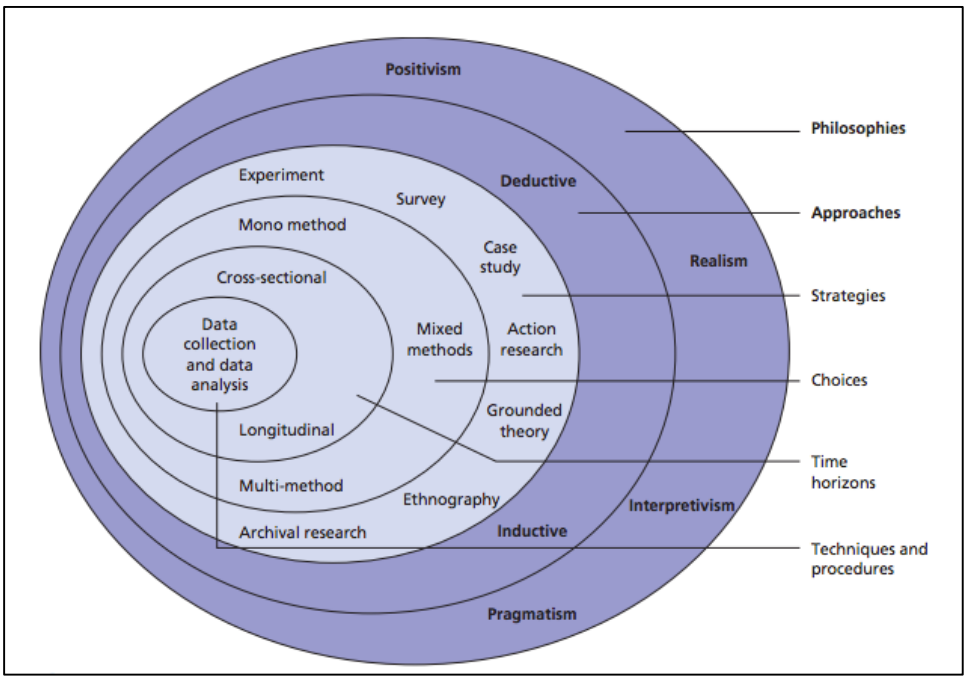


Figure 1: The research onion (Saunders, Lewis, and Thornhill 2015)

2.1 Philosophy

Philosophical paradigms generally reflect a researcher's own understanding of the nature of the conducted study. Lincoln and Guba (1985) argue that logical and objective explanations for a chosen research philosophy can hardly be given, as the selected paradigm purely relies on the researcher's presumption of reality. This is why academics should describe a paradigm that fits best to their conducted research, without aiming at finding logical verification for it.

In the case of this thesis, we argue that our philosophical paradigm mainly falls within the stance of positivism. Positivism assumes that reality can be measured by applying quantitative methods, such as experiments, surveys and verification of hypotheses (Rana Sobh and Chad Perry 2006).

In fact, our thesis relies on different quantitative approaches to measure various types of data that describe reality.

However, we will also explain how we triangulated the results in order to enhance validity of our study. Triangulation stems from realism that argues that all theory can be revised and reality is only imperfect.

2.2 Approach

This thesis follows a deductive research approach. The goal of deductive research is to test theoretical concepts using new empirical data. It states a question to be solved and applies analytical methods to receive results and findings. Finally, deductive research aims at testing the newly developed insights (Bhattacherjee 2012).
Following this perspective, the aim of our study is described by the defined research question. In reviewing relevant literature, we will establish solid theoretical knowledge for our analysis. Afterwards, the defined research question will be tested by applying different quantitative models. Lastly, the results of a verification study will examine the validity of our approach and findings.

2.3 Strategy

For applying the deductive analysis approach, this thesis follows a case study research strategy. A case study can be defined as "a strategy for doing research which involves an empirical investigation of a particular contemporary phenomenon within its real life context using multiple sources of evidence" (Robson and McCartan 2016).
Our research investigates the phenomenon of lead userness in an Android development online forum using multiple analytical methods.

For verifying our results, we follow a survey strategy by sending out questionnaires to a selected user population. In general, a survey strategy allows the collection of quantitative data that can be analyzed using descriptive and inferential statistics (Saunders, Lewis, and Thornhill 2015).

2.4 Choices

The forth layer of the onion framework describes whether a study is using qualitative or quantitative data. Our research can be characterized as a multi-method quantitative study, as it comprises two quantitative analysis types (Saunders, Lewis, and Thornhill 2015). First, the processing of user data with several computer-aided techniques and secondly, the evaluation of a questionnaire sent out to enhance the validity of the primary study.

2.5 Time horizons

In this category, one should decide about the time frame of the study and whether it is cross-sectional or longitudinal (Saunders, Lewis, and Thornhill 2015).

This study is a mix of longitudinal and cross-sectional research. We analyzed user behavior within a long period by accessing historical social media data (longitudinal). For enhancing validity of our approach, we conducted a survey over 10 days (cross-sectional). This means we did a 'snap-shot' of the status quo of that time.

2.6 Techniques and procedures

The last aspect of the research onion diagram regards techniques and procedures for data collection and data analysis (Saunders, Lewis, and Thornhill 2015).

In total, we accessed two different data sources. For our novel approach of lead user identification, we extracted data directly from the online community. In order to do so, we conducted web scrapping of social media data in structured and unstructured form.

Since this information was publicly available, we did not see any legal or ethical issues in downloading, using and processing the data.

Our second data source consisted of provided feedback collected through a survey sent out to 1000 forum members. This survey contained 15 questions that could be answered with values from 1 to 5 on a likert-type scale.

Our data analysis is mainly based on two frameworks, lead user theory and social media analytics. We show how we gathered information to analyze and identify attributes from the lead user theory by applying methods originating from social media analytics.

The description of the chosen research context is based on Porter's (2014) 'Five Ps of Virtual Communities'.

In order to precisely guide through the conducted analysis, this thesis makes use of the 'Social Media Analytics Framework' designed by Stieglitz et al. (2014) as the underlying theoretical concept. Its steps of tracking, preparing and analyzing social media data lead us to the desired results.

The analysis part this thesis shows how we applied machine learning and other computer-aided methods to the gathered and pre-processed data.

In order to enhance validity of our findings we further describe the application of a traditional screening search that stems from lead user theory. The results of the screening questionnaire were triangulated to the outcomes of our main analysis.

3 Theoretical review

This chapter introduces relevant theory in the field of lead user research, social media analytics and machine learning.

3.1 Lead user theory

The lead user theory portrays the central theme of this thesis. It is important to pay significant attention to the variety of literature dealing with lead users and lead userness in order to give a full picture of the topic and to establish a solid basis for further explanations. It is not only essential to understand the specific characteristics lead users show and how literature dealing with lead userness has been evolving, but also to discuss the different methods researchers describe in order to identify these individuals.

Therefore, the following chapter reviews relevant academic research on the lead user concept. It starts with an introduction to lead user theory to provide a profound understanding of the discussed topic. Evidence of the importance of user-centric innovation is found in a variety of diverse research themes that is also introduced. Lead userness is defined by several distinct characteristics that are presented in the following. In order to successfully integrate lead users into product development activities, literature suggests a four-step process called the 'lead user method'. As this paper concentrates on ways how to identify lead userness in the marketplace, it particularly pays attention to the identification part of the process when reviewing the lead user method. Traditional methods of lead user identification and their specific characteristics and potential shortcomings are discussed subsequently. During the most recent years, a new stream of literature has been evolving, dealing with new methods of lead user identification. With these mostly computerized and data-driven approaches, we finalize this chapter.

3.1.1 Introduction to the lead user theory

The emergence of the lead user concept leads back to the research into different sources of innovation. Several studies show that users rather than manufacturers are often the main initiators in the development of commercially successful new products and technologies (Burke and Enos 1963; Freeman 1968; Lüthje 2004; von Hippel 1986). User innovation can be observed in several product categories, from industrial goods (Herstatt and von Hippel 1992), over

software (Urban and Von Hippel 1988; Franke and Hippel 2003), up to consumer goods (Lüthje 2004). Researchers proof that user initiated innovations in these industries regularly outperform classic new product development processes and that the integration of users displays a promising strategy for any kind of organization (Lüthje and Herstatt 2004).

Empirical research indicates that these user-centric innovations are mostly performed by a group of individuals or firms named 'lead users'. According to von Hippel (2005, 1986) lead users are members of a user population that are "at a leading edge of an important market trend(s)" (von Hippel 2005, 47), i.e. they currently feel needs that the majority of the market will experience later. As diffusion theory indicates, general market needs are dynamic, develop over time and are determined by underlying trends (Rogers and Shoemaker 1971). Thus, von Hippel (2005) argues that there exist users whose individual needs foretell important market trends as their needs diffuse over time and at some point impact all members of a society. Furthermore, lead users highly benefit from the solution to their needs and thus might become innovators themselves if existing products or services do not meet their expectations (von Hippel 1986). Von Hippel (1986) argues that the higher the benefit a user obtains from a solution to his individual needs, the greater the innovation effort.

Research provides us with evidence that lead users are regarded as a valuable source for innovative and commercially highly attractive product and service concepts (Franke, von Hippel, and Schreier 2006; Franke and Hippel 2003; Lilien et al. 2002; Morrison, Roberts, and Midgley 2004). Due to their high innovativeness and their leading edge status, various authors suggest that lead users should be integrated into corporate new product development processes by applying the 'lead user method' (Lüthje and Herstatt 2004; Urban and Von Hippel 1988; von Hippel 1986). The lead user method describes approaches how firms learn from users' needs they face at the leading edge of a particular market. It further portrays their individual solutions and shows how to translate those into valid and promising new technologies and products. The ultimate goal of the lead user method is to let the identified individuals participate in workshops where they are given the opportunity to express the needs and solutions they are dealing with (Schreier and Prügl 2008).

Literature extending the lead user theory focuses not only on the developer perspective of lead userness, i.e. users that create products and service themselves, but also highlight other factors that distinguish lead users from other individuals. Schreier & Prügel (2008) demonstrate that lead users not only show innovative behavior in terms of generating novel product ideas but also by adopting new products and services much faster and in greater proportions compared to the majority of the market. This is due to the fact that they are ahead of what later becomes general market trends, thus earlier perceive leading-edge needs and show higher benefits from the adoption of new solutions (Schreier and Prügl 2008; Lüthje 2004). Furthermore, it is shown that lead users speed up the diffusion of new innovations and serve as opinion leaders for different product categories (Morrison, Roberts, and Midgley 2004; Urban and Von Hippel 1988).

The fact that user innovations account for a considerable proportion of newly developed products and services (Lüthje 2004; von Hippel 1986; Freeman 1968; Burke and Enos 1963) leads to the question which factors influence this behavior and how to separate lead users from other groups of individuals. In order to do so, several distinctive characteristics have to be considered. The following explanations precisely describe different lead user characteristics and show how research gradually developed this perspective.

3.1.2 Lead user characteristics

Von Hippel (1988, 1986), who pioneered the area of lead user research describes lead users according to two specific characteristics.

Firstly, lead users are said to be ahead of what later become general market trends. They face product needs considerably earlier than the majority of market participants. These so called leading edge needs might consequently fuel the innovativeness of lead users as they try to find solutions to their individual circumstances (Schreier and Prügl 2008). Furthermore, evidence is found that the position on the leading edge of a market and the resulting innovativeness is positively linked to an outcome of attractive products. As lead users develop solutions that solve their leading edge needs, these products later potentially appeal to the entire market (von Hippel 2005). In general, lead users are regarded as a valuable source for commercially attractive product and service concepts (Franke, von Hippel, and Schreier 2006; Franke and Hippel 2003; Lilien et al. 2002; Morrison, Roberts, and Midgley 2004).

Secondly, lead users are unsatisfied with existing products and services and thus significantly benefit from a solution matching their individual needs (von Hippel 1986). According to that, von Hippel (1986) argues that the higher the level of dissatisfaction and the greater the benefit from a novel solution matching the users' needs, the higher the innovation effort. In short, the dissatisfaction characteristic considerably fuels the innovation activities of lead users. This can be directly linked to an earlier stream of literature, saying that the investment in innovation efforts is dependent on its expected benefits (Mansfield 1968).
Literature extending the lead user theory adds further relevant attributes that precisely describe lead users. Schreier and Prügl (2008) who studied several innovation activities in extreme sport communities show that lead users also tend to possess more knowledge and use experience in a certain product area than the majority of the market. It is found that product expertise and use experience is positively linked to a user's innovativeness and thus lead userness. They conclude that besides the ahead of trend and the dissatisfaction aspect, the presence of product related knowledge and use experience play a crucial role in the identification of lead user behavior. Also Gatignon and Robertson (1985) state that "the key to diffusion of technological innovation may be in building the consumer knowledge and experience base for this type of technology" (Gatignon and Robertson 1985, 863).

Besides the 'ahead of trend', 'dissatisfaction' and 'product related knowledge' aspects, lead users are often characterized as opinion leaders (Morrison, Roberts, and Midgley 2004; Urban and Von Hippel 1988). Opinion leaders are "[…] individuals who exert an unequal amount of influence on the decisions of others" (Rogers and Cartano 1962, 435). Opinion leaders act as a contact point for others looking for advice and information before making a decision, i.e. they exert personal influence on others in a variety of situations (Rogers and Cartano 1962). In contrary to opinion leadership, opinion seekers are individuals that actively search for new information and act as recipients for opinion leaders' advice (Flynn, Goldsmith, and Eastman 1996). According to Urban and von Hippel (1988) lead users fulfill a crucial role as opinion leaders especially in the post-launch phase of a new innovation, as they are said to fuel the diffusion of technology. Also Schreier, Oberhauser, and Prügl (2007) clearly demonstrate that users situated on the leading edge of a market show considerable opinion leadership characteristics, as they "[…] serve other users as role models, as they are ahead of the mass who seeks to follow their lead" (Schreier, Oberhauser, and Prügl 2007, 19). Their study of tech-divers

proves that lead userness is negatively related to opinion seeking and that leading-edge tech divers serve as opinion leaders in their field. Following this line of argumentation, it is obvious that lead users are not only seen as the initiators of new product development but also fuel the diffusion of newly introduced products by acting as opinion leaders.

Lüthje (2004) who studied the innovation activities of users in the field of consumer products portrays six characteristics describing lead userness in consumer markets. In addition to the already discussed 'ahead of trend', 'dissatisfaction', 'opinion leadership' and 'product-related knowledge' characteristics, he introduced the concept of 'involvement' to the lead user research. This thesis bases its definition of involvement on research conducted in the field of social media. According to Amaro and Duarte (2015), involvement in social media can be analyzed on a behavioral perspective. Thus, involvement can be defined by time spent and intensity shown in online communities. One can say that highly involved individuals show high engagement in terms of time spent online and the creation of new content with significant length.
Lüthje (2004) demonstrates that a high level of user involvement is positively linked to lead userness. His explanation is based on innovation costs in sport-related consumer goods. Lüthje (2004) argues that these costs are lower for users that show high commitment, i.e. involvement in their product field. These users have fun finding new ideas and solutions and thus become innovators. It is further shown that especially the combination of high-expected benefit and high involvement lead to user innovations in sport-related consumer goods.

3.1.3 The lead user method

In order to profit from the high innovativeness of lead users and to learn from their needs and ideas, researchers particularly pay attention on ways how to include those individuals into new product development processes (Lüthje and Herstatt 2004; Urban and Von Hippel 1988; von Hippel 1986). Especially in the field of consumer goods, innovative users tend to not use their knowledge for commercial reasons, but "[…] primarily try to realise their ideas for private purposes only" (Lüthje 2004, 693). It is obvious that concepts and processes are required to let manufacturers identify lead user behavior and to integrate these individuals into their corporate product development activities (Lüthje 2004).

In order to do so, von Hippel (1986) developed a four-step process that describes how firms learn from users' needs they face at the leading edge of a market. The so-called 'lead user method' directly evolved from the research on the behavior of innovative users. Since that, von Hippel (1986) has been the basis for a variety of studies dealing with ways how to translate lead user solutions into commercially successful products (Urban and Von Hippel 1988; Lüthje and Herstatt 2004; von Hippel 1986).

The ultimate goal of the lead user method is to let the identified individuals participate in workshops where they are given the opportunity to express the needs and solutions they are dealing with. Corporate manufacturers can learn from lead user behavior and have the chance to collaborate with these individuals in order to develop commercial products that show the potential to appeal to an entire market (Schreier and Prügl 2008).

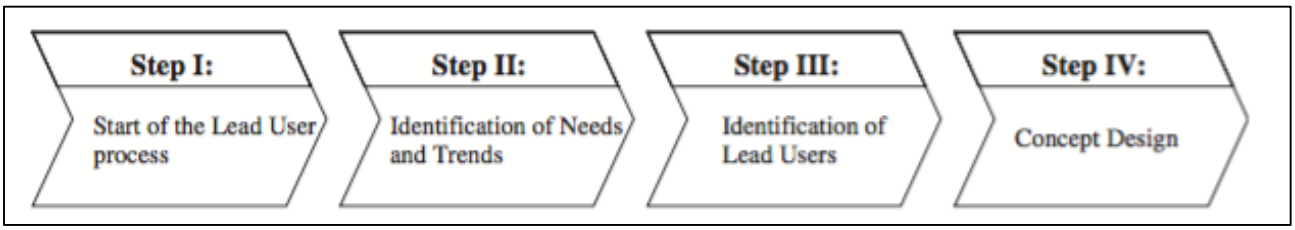


Figure 2: The lead user method (Schreier, Oberhauser, Prügl 2007)

The lead user method described by von Hippel (1986) and portrayed in figure 2 starts with a definition of the search field. It can be a market, a product or service area in which new concepts and product ideas should be generated (Lüthje and Herstatt 2004). This step should also include a statement of the ultimate goal of the lead user method as well as an evaluation of internal and external factors that might influence its application (von Hippel, Thomke, and Sonnack 1999).

Once the search field is defined, the lead user method continuous with the identification of needs and trends in the market. In order to successfully identify lead users in a particular product area, it is essential to describe the underlying trend on which these individuals might show a leading edge position (von Hippel 1986). Conducting interviews with market experts and scanning literature and industry databases represent promising strategies in the identification and selection of the most attractive trends (Lüthje and Herstatt 2004).

Trend identification is followed by the actual process of detecting lead users. The identification step of the lead user method aims at finding individuals that possess the distinct characteristics of lead userness. Researchers have developed a variety of quantitative and qualitative methods that should enable firms to discover lead user behavior (von Hippel 1986; Lüthje and Herstatt 2004). The most basic and best described methods are the screening and pyramiding approach (Lüthje and Herstatt 2004; Stockstrom et al. 2016; von Hippel, Franke, and Prügl 2009; Lilien et al. 2002). While screening basically scans a huge amount of data for lead user characteristics in parallel, pyramiding utilizes a network approach that is applied to groups of people in series (von Hippel, Franke, and Prügl 2009). More novel approaches include broadcast search (Poetz and Prügl 2010), netnography (Belz and Baumbach 2010) and automated methods based on the application of data mining techniques (Pajo et al. 2013).

The lead user method ends with the development of new product concepts. Therefore, the identified lead users have to be involved in corporate innovation activities. Conducting lead user workshops is one example of how firms can learn from needs, innovative ideas and solutions at the leading edge of a market (Lüthje and Herstatt 2004).

The challenge of user involvement lies in the users' potential unwillingness to openly reveal his or her ideas to manufacturers. This can be overcome by setting the right expectations and show users how they benefit from cooperating. Special rewards, like exclusive information or the opportunity to get early access to the newly developed product, might increase the readiness to participate in corporate innovation processes (Brockhoff 2003).

This thesis pays particular attention to the third step (identification) of the lead user method. It discusses and develops new quantitative approaches in lead user detection. In order to distinguish these computer-aided methods from traditional techniques, it makes sense to review relevant literature on classic lead user identification techniques in more detail. We discuss 'mass screening', 'pyramiding', 'broadcast search' and 'netnography'. In the past years, new literature has evolved that focuses on new ways of identifying lead userness. These computer-aided approaches promise new possibilities for lead user research and are introduced in the following.

3.1.3.1 Mass screening

Mass screening is one of the most basic methods of lead user detection and it is applied in a great number of empirical studies (von Hippel, Franke, and Prügl 2009; Urban and Von Hippel 1988; Herstatt and von Hippel 1992; Morrison, Roberts, and Midgley 2004; Schreier and Prügl 2008; Lüthje and Herstatt 2004; Stockstrom et al. 2016).

This rather standardized and quantitative technique is based on parallel scanning for lead user characteristics in a large entity of potential relevant product users. In practice, mass screening is often conducted with surveys in form of written questionnaires or telephone interviews. Participants are invited to answer questions about their own innovation activities and lead user behavior. The goal is to identify those individuals that show a high score regarding the desired lead user characteristics (Belz and Baumbach 2010).

As mass screening relies on parallel search in a pre-defined user population, in can be a very effective method if the search field contains a manageable number of potential users or customers (Lüthje and Herstatt 2004). A well-defined search field with clear boundaries, the possibility to reach the entire user population and valid self-assessments lead researchers to those individuals with the highest score regarding the desired attributes (Stockstrom et al. 2016).
However, as mass screening is based on self-assessments of targeted individuals as well as the fact that it stays in-between well-defined search boundaries, literature often criticizes the technique as being less efficient compared to other methods of lead user identification (von Hippel, Franke, and Prügl 2009).

3.1.3.2 Pyramiding

In contrary to mass screening, von Hippel, Franke, and Prügl (2009) argue that the pyramiding method requires less effort in finding lead user behavior. Furthermore, Poetz and Prügl (2010) state that it overcomes local search bias by providing the chance to tap into knowledge pools that are far away from the initially pre-defined search field.

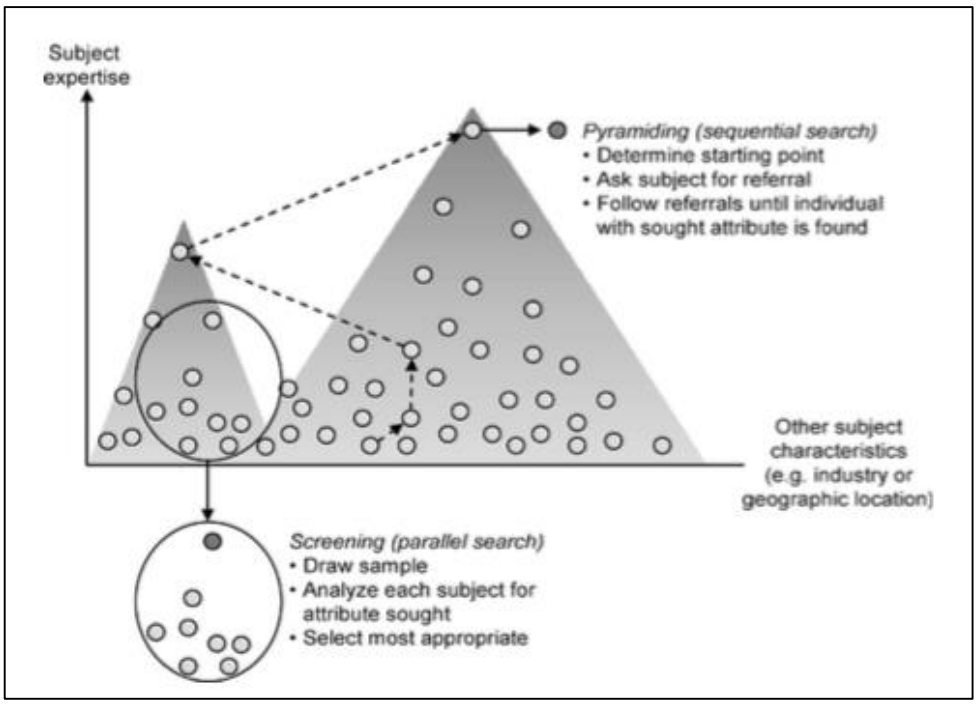


Figure 3: Screening vs. pyramiding (von Hippel et al. 2009)

In general, pyramiding is a lead user search technique that is based on the idea that users showing a high score on a given attribute, e.g. product-related knowledge and expertise, potentially know other individuals that possess an even higher level of this particular characteristic. In the process of lead user search, people are asked to name other persons that, in their opinion, know more about the search field or possess better information about other experts in the product area. The identified individuals are further asked to name people that even show a higher level of the desired attribute compared to themselves. The pyramiding process continuous in this way until individuals are identified that show the researcher's desired score of a particular attribute, e.g. product-related knowledge and expertise. These users are said to reside on the very top of the search pyramid (Poetz and Prügl 2010; von Hippel, Franke, and Prügl 2009).
Research provides us with evidence that pyramiding is a highly efficient search method for identifying individuals that show very specific and rare characteristics, e.g. lead userness, within large and poorly mapped search areas (von Hippel, Franke, and Prügl 2009).

In addition to the efficiency aspect, Poetz and Prügl (2010) argue that pyramiding holds two method-specific advantages that outperform mass screening techniques. First, it allows 'learning on the fly', i.e. every time a user is contacted and asked for a referral, the researchers can adjust

the approach and learn from the person's answers. In doing so, the learnings and feedback can be utilized to improve the lead user search. Secondly, pyramiding is not limited by a pre-defined search field. In the course of its application, it invites users to refer to individuals that might potentially be outside of the defined population and in so-called analogues domains.
This allows to identify lead users not only in the actual target market, but also searches for individuals with relevant ideas and solutions in more distant markets where similar needs and expectations exist (Lilien et al. 2002).

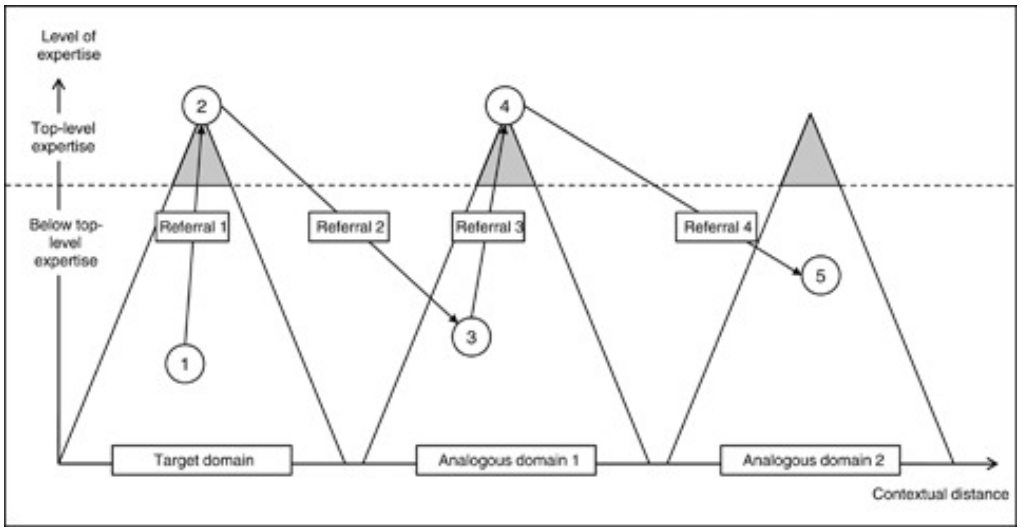


Figure 4: Pyramiding for lead user detection (Poetz and Prügl 2010)

Figure 4 illustrates the application of pyramiding search, first in the actual target domain and later in analogues markets.

The validity of the pyramiding search method and its ability to cross domain-specific boundaries is proven in a variety of lead user studies (Lilien et al. 2002; von Hippel, Franke, and Prügl 2009; Stockstrom et al. 2016; Poetz and Prügl 2010).

3.1.3.3 Broadcast search

Besides mass screening and pyramiding, broadcast search represents another promising strategy in the identification of lead users. The underlying idea of this approach is to post a problem or question into an online community and receive ideas and solutions from problem solvers (Lakhani 2006).

In contrast to other lead user search approaches, broadcast search first starts with a clear description of the unresolved problem or the needed improvement to an existing product or service. This is followed by posting the defined issue to online communities where it appeals to a diverse set of potential problem solvers. These individuals post their solutions as answers into the virtual communities and researchers can start communicating with the identified problem solvers. It is proven in a variety of studies that innovative users show a high degree of intrinsic motivation to submit solutions to the presented issue (Hienerth, Poetz, and von Hippel 2007).

Jeppesen and Frederiksen (2006) further provide us with evidence that most of the presented answers and solutions are in fact submitted by users showing strong lead user characteristics.

In the application of broadcast search, besides being intrinsically motivated, users are often incentivized by the company that is looking for solutions through the broadcast method. This is done in the form of an idea competitions where selected winners are later invited to participate in lead user workshops (Hienerth, Poetz, and von Hippel 2007). Conducting idea competitions as a form of broadcast search is said to increase the number and quality of the submitted ideas and solutions (Piller and Walcher 2006).

Lakhani (2006) describes broadcast search as an economically efficient research method, both for solution seekers as well as problem solvers. Compared to mass screening and pyramiding, it involves fewer costs for researchers, as users self-select depending on their individual solutions for the posted problem. Furthermore, broadcast search, like pyramiding, has the ability to overcome local search bias. Lakhani's (2006) findings show that the probability of a posted problem being solved is even higher if the solution comes from users from diverse target domains.

3.1.3.4 Netnography

'Netnography' is a relatively new approach for lead user detection. It was first utilized in the field of lead user research by Belz and Baumbach (2010) who used the technique to identify lead user characteristics in online communities. Netnography compares posts of the most active users to a set of lead user characteristics.

In general, netnography has arisen with the growth of the Internet and it is used to systematically analyze consumer insights in online communities. It has evolved from ethnographic research and is a hybrid term made up of Internet and ethnography. Ethnography is an anthropological method that has its roots in social science. It is defined as a qualitative and open-ended research technique used to precisely describe "distinctive meanings, practices and artifacts of particular social groups" (Kozinets 2002, 3). Its flexibility, rich qualitative content as well as its open-endedness makes ethnography applicable in a variety of circumstances (Kozinets 2002).

Netnography is based on ethnographical research approaches and uses their methods for studying cultures and groups that emerge online. It helps researchers to understand, identify and assess consumer needs, decisions, trends, behavior as well as influences in online communities. Netnography is faster, easier and more cost efficient than classic ethnographic research and it provides insights in symbolism, meanings, and consumption patterns of users online (Kozinets 2002).

Empirical studies show that lead users tend to actively participate in online communities to discuss problems and solutions about existing products and services (Sawhney, Verona, and Prandelli 2005; Jeppesen and Laursen 2009). This is the reason why Belz and Baumbach (2010) introduced the application of netnography to the area of lead user research. Nethnographic research, like ethnography, usually consists of four steps: 'entré', 'data collection and analysis', 'interpretation' as well as 'research ethics and member checks'.

After the identification and selection of the suitable online community (entré), Belz and Baumbach (2010) collected posts from users discussing sustainable food consumption and related products. Afterwards, they conducted a qualitative analysis by manually coding and interpreting the posts according to the five lead user characteristics: 'dissatisfaction', 'ahead of

trend', 'product-related knowledge and expertise', 'opinion leadership' and 'involvement'. In order to fulfill ethical guidelines and member checks, the researchers informed all users of the analyzed online community about the activities conducted. The authors conclude that approximately 22.5% of active online users they have analyzed possess lead user characteristics. Furthermore, they compared the results with a traditional mass screening approach and could prove that netnography, although showing some limitations, is indeed a valid technique in the identification of lead user characteristics in online communities.

3.1.3.5 Fast lead user identification (FLUID)

During the most recent years, another stream of literature has evolved, dealing with new and mainly automated ways of lead user identification. These techniques aim at overcoming disadvantages of traditional lead user detection methods by applying computer-aided techniques and algorithms. Pajo et al. (2013, 2015) argue that these new approaches generally outperform mass screening, pyramiding, broadcast search and netnography in terms of efficiency, time needed and overall costs involved in conducting the analysis. It is possible to analyze and evaluate large quantities of user data "[…] without relaying on interactive human interference" (Pajo et al. 2013, 507). The researchers propose a 'Fast Lead User Identification' (FLUID) method based on rich user data that is extracted from the social network Twitter.

The underlying idea of the FLUID approach comes from netnography research (Pajo et al. 2013), as it has shown that user needs, motivations, desires or attitudes can be detected in online communities (Kozinets 2002).

FLUID is described as a systematic and automated way to find and identify lead users. It relies on data mining methods to extract user generated content from social media and to gain insights about user behavior and characteristics. The approach is aimed at separating individuals with distinctive lead user characteristics from non-lead users (Pajo et al. 2013).

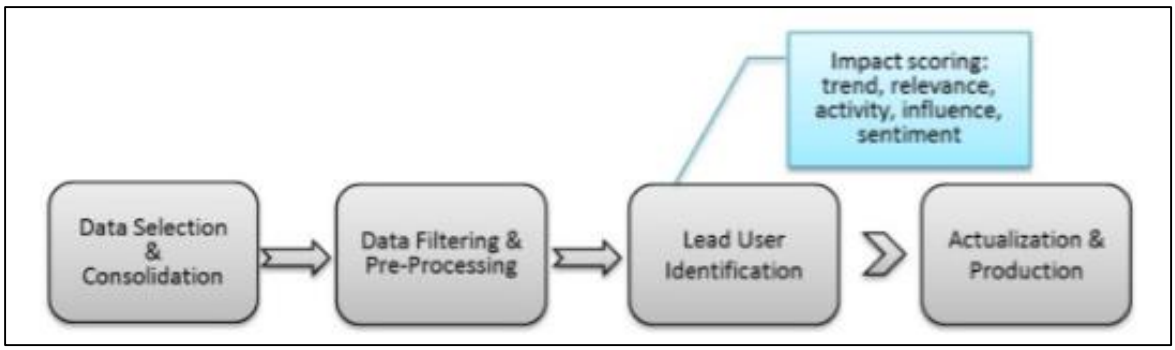


Figure 5: The FLUID system (Pajo et al. 2013)

Figure 5 shows the typical FLUID system. In the process of its application, data is obtained form the selected online source. This step is followed by data-filtering and pre-processing in order to erase irrelevant content and spam. Thereafter, the actual analysis is conducted by applying data mining methods that detect activity, trend, influence, relevance as well as sentiment of the user content. Individuals that rank high on these criteria are identified as potential lead users (Pajo et al. 2013).

This thesis is motivated by the research conducted and methods developed in the field of netnography and FLUID. Pajo et al. (2015, 2013) show how data mining is used to automate and speed up lead user detection processes. In contrast to the FLUID approach introduced above, our research field is a social community in form of an online forum, where users discuss new developments in the area of Android hard- and software. As the setting of this thesis differs from the FLUID research, new computer-aided techniques to lead user identification have to be introduced. Others, like sentiment analysis, can be transferred to the new research theme.

The following chapters provide a solid introduction to the field of social media analytics as well as discuss different machine learning techniques that we applied to detect lead user characteristics in the online community. Computer-aided techniques, such as cluster analysis, sentiment detection as well as trend analysis using part-of-speech tagging are introduced and discussed.

3.2 Social media analytics

The second major concept for this thesis is 'social media analytics' (SMA). Social media analytics is an emerging interdisciplinary research field that aims at combining, extending, and adapting methods for exploring social media data (Stieglitz et al. 2014).

It is applied for monitoring, analyzing, measuring and interpreting digital interactions and relationships of people, topics, ideas and content. Human relations can take place in external-facing communities as well internal ones managed by organizations. Social media analytics includes forms of machine learning, like sentiment analysis and natural-language processing as well as social networking analysis (influencer identification, profiling and scoring). Furthermore, it contains advanced techniques such as text analysis, predictive modeling and recommendations, automated identification and classification of subject/topic, people or content (Gartner 2012).

As social media analytics comprises a huge variety of concepts and methods with its boundaries not clearly set, we will only highlight those parts that are relevant for this thesis. Therefore, the next chapters will give a good understanding of social media in general. We will further focus on online communities and their position in social media. The introduction to Stieglitz et al.'s (2014) 'Social Media Analytics Framework' will finally lead us to 'Big Social Data', a recently emerged term dealing with the gathering, processing and analyzing of high volume of information from social media.

3.2.1 Social media

In the last decade, social media networks have been enormously growing in their user base and in adoption. Now there are more than 1.5 billion Facebook members and Twitter has more than 320 million monthly active users (Statista 2016).

Obar and Wildman (2015) identified the following commonalities among current social media services:

1) Social media services are (currently) Web 2.0 based
2) User-generated content is key in social media
3) Users generate profiles for a site or app maintained by a social media service
4) Social media services enable the development of social networks by connecting a profile with those of other users

Zeng et al. (2010) highlight the following categories of social media services: weblogs, microblogs, social network sites, location-based social networks, discussion forums, wikis, podcast networks, picture and video sharing platforms, ratings and review communities, social bookmarking sites, and avatar based virtual reality spaces.

In reviewing these categories and comparing it to the current online landscape, one should consider adding a category for messenger services. This is because four out of nine leading social network services deal with direct and group messaging services (WhatsApp, Facebook messenger, QQ and WeChat) (Statista 2016).

Nevertheless, Zeng et al.'s (2010) definition of social media can still be applied. They state that social media refers to "[…] a conversational, distributed mode of content generation, dissemination, and communication among communities" (Zeng et al. 2010, 13).

Montalvo (2011) sees social media as "[…] fundamentally scalable communications technologies that turn Internet-based communications, (i.e., smart phones, PCs, tablet computers, portable media players, etc.) into an interactive dialogue platform. Social media platforms […] all exist as a result of Web 2.0" (Montalvo 2011, 1).

The benefits from this scalability are expressed in 'Metcalfe's Law'. "Metcalfe said the usefulness of a network improves by the square of the number of nodes on the network" (Karlgaard 2005, 33). In other words, the more people are involved in social media, the better it is for every single user and the overall community.

3.2.2 Virtual communities

Virtual communities can be seen as platforms where individuals with common interests interact with each other. These interactions follow specific norms and protocols and are at least partly supported by technology (Porter 2004).

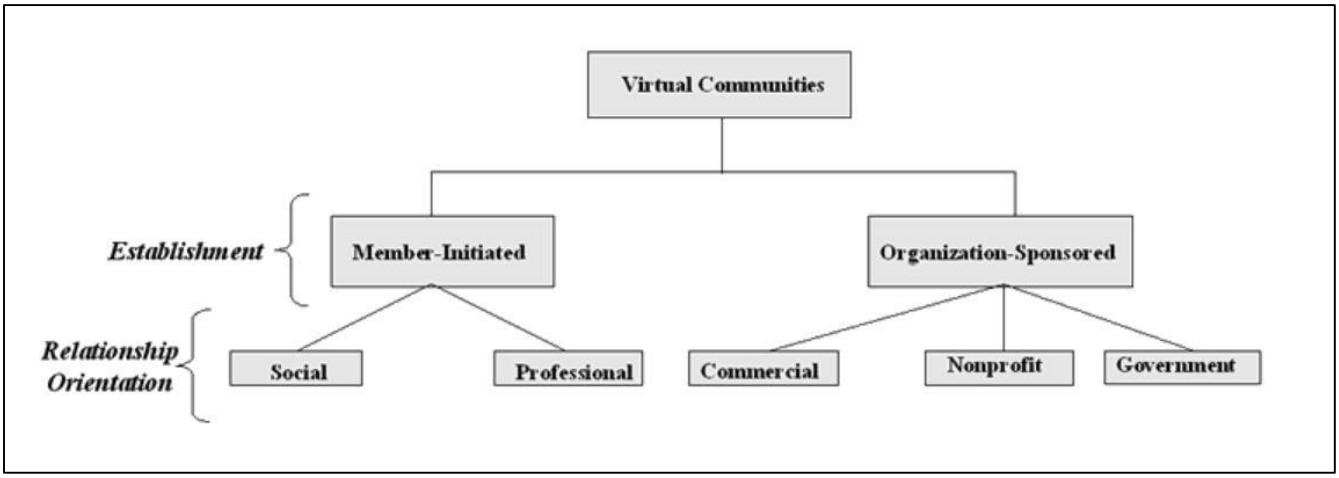


Figure 6: Typology of virtual communities (Porter 2004)

Figure 6 shows Porter's (2004) proposed typology of virtual communities. On the first level, she distinguishes between 'Member-Initiated' and 'Organization-Sponsored' communities. If the community was established by, and remains managed by members it is seen as 'Member-Initiated'. Commercial or noncommercial (e.g. government, non-profit) communities are described as 'Organization-Sponsored'.

On the second level, Porter (2004) states the different relationship characteristics among the members of the community. Member-initiated communities facilitate social networks among individuals, while organization-sponsored communities include the organization as a crucial part of the conversation.

Furthermore, Porter (2004) created the 'Five Ps of Virtual Communities' in order to classify the key attributes of a virtual community. These are: 'Purpose, Place, Platform, Population and Profit Model'.

'Purpose' describes the specific content of the interaction online. It primarily focuses on the communication among community member. 'Place' is an attribute that defines where exactly the interaction happens, either completely online or at least partially virtual. 'Platform' refers to the design of interaction, i.e. the technical aspect of communication. It enables synchronous, asynchronous or both forms of communication. 'Population' is an attribute that focuses on group structure (small or large) as well as the intensity of relationships (strong or weak). Lastly, Porter (2004) describes 'Profit Model' as a characteristic that specifies whether a community is revenue generating.

We later use this classification system as the foundation for a structured analysis of an online forum.
In the course of the analysis, we particularly focus on online forums or web forums as specific types of online communities.

Online forums can be defined as "[…] a website or section of a website that allows visitor to communicate with each other by posting messages" (Christensson 2011, 1). They exist for a wide range of subjects. Whereas many online forums deal with IT topics, they are not limited to this domain. Examples for topics are babies, cars, fitness, health, houses, parenting and teaching. Forums can be very general, like a gaming forum and others focus on very specific topics, such as a forum for a particular computer game. Web or online forums are also known as Internet forums, online bulletin boards or discussion boards. In general, they allow visitors to view the different postings. In order to create a new topic called thread or to comment on another subject, the user has to create an account (Christensson 2011).

3.2.3 Social media analytics framework

This part aims at gathering further definitions of social media analytics, stating its goal and purpose as well as defining its position in the research domain.

Zeng et al. (2010) state that social media analytics deals with developing and evaluating informatics tools and frameworks to collect, monitor, analyze, summarize, and visualize social media data. Vatrapu (2013) refines this definition by describing SMA as "[…] the collection, storage, analysis, and reporting of social data emanating from social media engagement of and social media conversations" (Vatrapu 2013, 152).

SMA can be used for facilitating conversations and interactions between online communities and for extracting useful patterns and intelligence to serve entities that include, but are not limited to, active contributors in ongoing dialogues (Zeng et al. 2010).
Furthermore, the research purpose is to develop and evaluate scientific methods as well as technical frameworks and software tools for tracking, modeling, analyzing, and mining large-scale social media data for various purposes (Stieglitz et al. 2014).

The research agenda is multidisciplinary in nature and has drawn attention from academic communities of all major disciplines. From an information technology standpoint, social media research has primarily focused on social media analytics and, more recently, on social media intelligence (Zeng et al. 2010).

In a business setting, SMA might be considered as a subset of business intelligence that is concerned with methodologies, processes, architectures, and technologies that transform raw data from social media into meaningful and useful information for business purposes (Stieglitz et al. 2014).

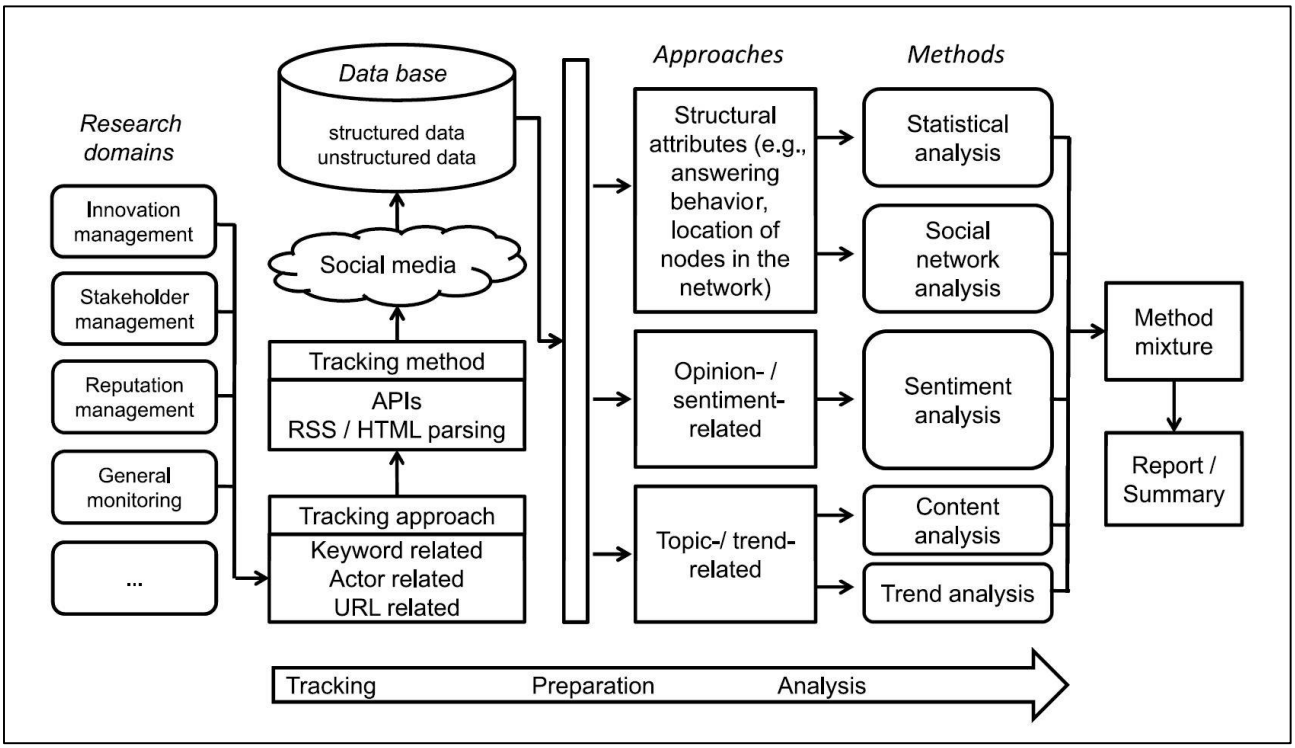


Figure 7: Social Media Analytics Framework (Stieglitz et al. 2014)

Figure 7 displays Stieglitz et al.'s (2014) 'Social Media Analytics Framework'.

Since SMA is usually driven by specific requirements from a target application, it shows that SMA can be applied to solve problems for the following research domains: innovation management, stakeholder management, reputation management, and many more (Zeng et al. 2010).

The framework further visualizes the underlying process in listing the approaches and methods for tracking, preparation and analysis of data. These steps differ slightly from the process steps

found in other literature, which is collection, storage, analysis, and reporting (Zeng et al. 2010; Vatrapu 2013).

The interdisciplinary nature of social media analytics is characterized by the systematic employment of a mixture of analysis methods from the computer science, mathematics/statistics, network analysis and linguistics (Stieglitz et al. 2014).

The steps of Stieglitz et al.'s (2014) framework are:

1) Identify data that fit to the given research aim and using of the keyword-, actor-, or URL related approaches.

2) Download the data via API's, RSS or HTML parsing. This is highly dependent on the particular social media platform.

3) The process step of storing the data is differentiated by structured data (links, network nodes and edges) or unstructured data (e.g., text, code, symbols).

4) Develop or modify a tool, which gathers and prepares the necessary data for the preprocessing step (e.g. by removing spam or stop words manually, or based on filters).

5) Dependent on the research goal, apply appropriate analysis approaches (e.g. the identification of structural attributes, sentiments, or topic- and trend-related patterns), methods (e.g. statistical analysis such as regression analysis, social network analysis, sentiment analysis, content analysis, or trend analysis), and analysis tools (e.g., Gephi, SentiStrength, Condor, NLTK). Distinguish between static or dynamic data analysis. Static data analysis might be useful to identify the co-occurrence of specific words in a data set. On the other hand, dynamic data analysis might be useful to better understand how issues are evolving over time.

3.2.4 Big social data

The increasing number of users and their growing engagement in social media, lead to a rapid growth of user-generated content. There are 500 million new tweets per day on Twitter (Internet live stats 2016) and 300 hours of new video material is uploaded to YouTube every single minute (DMR Stats 2016).

This enormous adoption and use of social media is generating large volumes of unstructured information. This huge amount of data is termed 'Big Social Data' (Vatrapu 2013).

Stieglitz et al. (2014) highlight that not only the most popular social media platforms, such as Twitter, Facebook, and LinkedIn generate high volumes of data, but also other platforms that facilitate mass collaboration and self-organization. Weblogs, wikis, and user tagging systems also grow massively.

Analysts have facilitated access to the large-scale empirical datasets but are still facing difficulties in storing and handling the data. Technical limitations, complexity and higher costs are only examples of the determining factors. 'Big Social Data' analysis evolves from this need and combines disciplines such as social network analysis, multimedia management, social media analytics, trend discovery and opinion mining (Cambria, Wang, and White 2014).
Another need for a structured data analysis approach stems from the fact that large data sets from Internet sources are often unreliable because of their potential incompleteness and inconsistency, in particular when multiple data sets are used together (e.g. social media content and location-based data). Regardless of their size, data sets are always subject to limitations and biases (Boyed et al. 2012; Stieglitz et al. 2014).
'Big Social Data' analysis methods can help to create understanding of those biases and limitations, since data analysis is most effective when researchers take account of the complex methodological processes that underlie the analysis of that data.

As crucial as understanding the methodological part, is the knowledge about the principles of machine learning techniques that can be applied. This is necessary in order to be able to choose the right analysis method.

3.3 Machine learning methods

Machine learning is a field of computer science and engineering that studies mathematical functions and applications for learning systems (Sugiyama and Kawanabe 2012).
It is designed to emulate human intelligence by learning from current context and the surrounding environment. Machine learning establishes the basis for a new era of applications, especially for dealing with big data (Naqa, Li, and Murphy 2015).
Arthur Samuel, who pioneered the area of machine learning, defined it as "the field of study that gives computers the ability to learn without being explicitly programmed" (Samuel 1959). Machine learning has already been successfully applied in a variety of areas ranging from

medical science, biology, finance, entertainment, computer engineering to pattern recognition and industrial engineering (Naqa, Li, and Murphy 2015).

In order to successfully apply machine learning methods, the input samples as well as the desired output have to be defined properly. Secondly, the appropriate machine learning model has to be chosen. In addition, data quality has to be considered as an important requirement as even a well working model or algorithm cannot substitute bad data. Thirdly, it is important to keep the chosen model as simple as possible to avoid any mismatches. Finally, it is important to accept that machine learning has some major limitations in terms of providing a intuitive interpretation of the learned process and the model outcomes (Naqa, Li, and Murphy 2015).

In general, machine learning can be categorized into three different types, depending on the method of learning: 'supervised learning', 'unsupervised learning' and 'reinforcement learning'.

Supervised learning relies on training data to establish an input-output relation. In contrary to that, unsupervised machine learning algorithms are not built upon given training samples. Their goal is to extract useful information behind data without prior learning. Like supervised learning, reinforcement algorithms rely on input data to establish an input-output relation. However, as the output samples cannot be observed directly, reinforcement methods aim at acquiring a policy function that needs to be trained without supervisors. They rely on reward information as training data and the policy function is learned in a way that the sum of rewards is maximized (Sugiyama and Kawanabe 2012).

This thesis focuses on supervised as well as unsupervised machine learning algorithms for identifying and analyzing lead user characteristics in online forums. The following chapters introduce the specific types of both approaches that were used for our analysis.

3.3.1 Supervised machine learning methods

Supervised machine learning, or classification, describes methods to build and train an algorithm by using a set of pre-labeled and well-known training data. Each example in the training data consists of a pair of an input object and the defined output label. In supervised learning, the learning algorithm analyzes the training data and builds a classification function for later

application. In other words, it learns how to analyze future new example input on basis of the trained pre-labeled data (Kotsiantis 2007).

This thesis focuses on supervised machine learning techniques to build classifiers for text categorization. We make use of sentiment analysis for text polarity identification as well as use tokenization methods for trend detection. Following explanations give the theoretical introduction to both research areas.

3.3.1.1 Sentiment analysis

For several years, sentiment analysis has been a widely discussed and researched supervised machine learning approach in the field of social media analysis and artificial intelligence (Pang, Lee, and Vaithyanathan 2002; Pak and Paroubek 2016; Go, Bhayani, and Huang 2009). Especially with the rise of online services such as Facebook, LinkedIn or Twitter every internet user is now invited to actively engage in the creation, sharing and rating of content online. The so published user messages in form of expressed interests, opinions, reviews or other forms of content prepares the ground for new procedures in in-depth analysis of human emotions and opinions. Researchers have started to systematically develop processes and applications used to gain and analyze users' opinions posted in social media networks. The most widely researched and used practice describes ways how to identify the specific sentiments of a given text, i.e. how the user expresses feelings regarding a certain topic. Sentiment analysis let researchers and companies tap into a completely novel pool of user data and information.

This chapter gives an introduction to sentiment analysis. We show the required steps for content preparation as well as highlight different approaches to text and sentiment classification.

The central purpose of sentiment analysis is to identify how sentiments are expressed in texts and whether they refer to a positive, negative or neutral attitude towards a certain subject. Therefore, sentiment analysis extracts written text and categorizes it according to the polarity of the given data (Pang, Lee, and Vaithyanathan 2002).

In accordance with Pang et al. (2002), Pak & Paroubek (2010) and Go et al. (2009) our descriptions follow a common sentiment analysis approach that is based on machine learning algorithms using 'Naive Bayes Classification', 'Maximum Entropy Classification' or 'Support Vector Machines' for text categorization. These machine-learning techniques are based on supervised learning that uses a pre-labeled text corpus (e.g. text sentiments labeled positive,

negative or neutral) to train a classifier. Later, the trained model is fed with new unseen data and executes the categorization task (Bird, Klein, and Loper 2015).
The following figure shows a typical process of text classification in sentiment analysis.

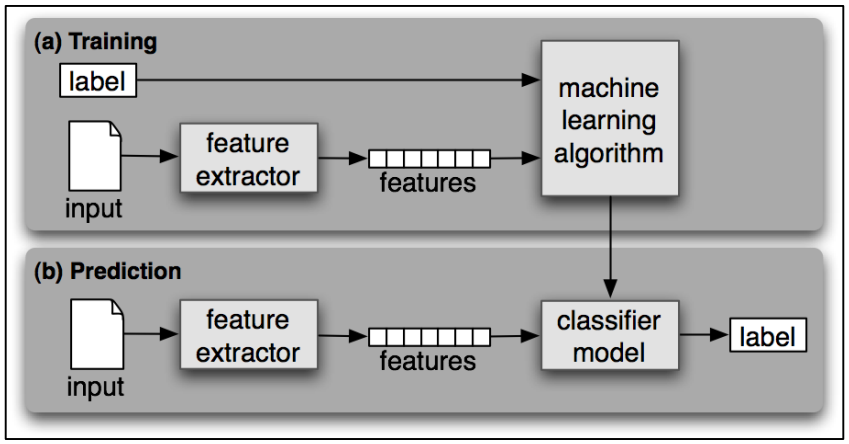


Figure 8: The process of text classification (Bird, Klein, and Loper 2016)

During the training phase a feature extractor converts pre-labeled training data into a feature set consisting of text fragments labeled as positive, negative or neutral (Bird, Klein, and Loper 2015). Pang and Lee (2008) highlight the importance of converting text into a feature vector as a major step in data-driven text processing as the output contains all the polarity information needed for training the classifier. The feature vector consists of either unigrams, i.e. every word is a single feature, bigrams or even n-grams. The question whether to choose unigrams or better higher level n-grams for sentiment analysis, appears to be a matter of discussion. While Pang, (Lee, and Vaithyanathan 2002) argue that unigrams regularly outperform bigrams in analyzing sentiment polarity in extracted movie reviews, Dave, Lawrence, and Pennock (2003) successfully applied bigrams and trigrams in their research of product-review polarity text classification.

After text pre-processing and feature extraction, the created feature vector is then transferred to the machine-learning algorithm that generates the classifier model for later prediction. The most common and widely applied supervised methods for sentiment analysis are ‘Naive Bayes Classification’, ‘Maximum Entropy Classification’ and ‘Support Vector Machine’ (Bird, Klein, and Loper 2015).

During the prediction phase the same feature extractor is used to convert new unseen pieces of text to feature sets. Lastly, the trained classifier model calculates the polarity of the new input data and labels it according to the trained algorithm. By applying this text classification process one can automatically analyze the sentiment of a huge set of input data (Bird, Klein, and Loper 2015).

Following explanations will give a brief theoretical introduction to the three supervised machine learning approaches 'Naive Bayes Classification', 'Maximum Entropy Classification' and 'Support Vector Machines'. All three methods are characterized as statistical learning algorithms, that have a clear underlying probability model, "which provides a probability that an instance belongs in each class, rather than simply a classification" (Kotsiantis 2007, 257).

Naive Bayes Classification

'Naive Bayesian' text classifiers are said to be fast, accurate, simple, and very easy to implement. Sentiment analysis using 'Naive Bayes Classifiers' begins with calculating the prior probability of each label of the training data. The estimation and decision process is based on how often each label is present in the training corpus and how every single feature contributes to the labeling result. In other words, the machine-learning system computes the probability "that an input will have a particular label given that it has a particular set of features" (Bird, Klein, and Loper 2015). When a new set of unseen data arrives, the 'Naive Bayes Classifier' calculates the probability for each label to be assigned to the new set of features by estimating the maximum likelihood of each label (Bird, Klein, and Loper 2015).

'Naive Bayes Classification' methods follow the assumption that all features are completely independent of one another given the specific label. This reduces complexity and simplifies processing of the data (Nedelcu 2012).

Maximum Entropy Classifiers

The 'Maximum Entropy' (MaxEnt) method is very similar to 'Naive Bayes Classification'. Instead of using prior likelihoods to estimate the appropriate data labeling, it uses an iterative scaling process to boost the performance of the model. In particular, it searches for the set of features that maximizes the total likelihood. Pre-labeled training data is used to find a set of features. This training set defines the label-specific expectations for the conditional distribution.

On basis of the labeled training data, the expected value of these features is calculated. Afterwards, random values are presented to the model and afterwards refined by an iterative process in order to find the optimal solution (Bird, Klein, and Loper 2015).
The greatest benefit of the 'Maximum Entropy' approach is that in contrast to 'Naive Bayes Classifiers' no independence assumption about the relationship between features is made. In other words, even features that consist of bigrams or n-grams can easily be added to MaxEnt without having any problems with feature overlapping (Go, Bhayani, and Huang 2009).
Therefore, 'Maximum Entropy' theoretically achieves better results than Naive Bayes. In fact, in two out of three datasets' Maximum Entropy' outperforms 'Naive Bayes' (Nigam 1999).

Support Vector Machines

'Support Vector Machines' (SVM) are regarded as being highly effective for traditional text classification. In fact, this technique is able to outperform 'Naive Bayes' and MaxEnt in most cases. This might stem from the fact that this approach is based on a large margin classifier and not on a probabilistic classifier like MaxEnt and 'Naive Bayes'. Furthermore, SVM can be used for classification and regression of information (Pang, Lee, and Vaithyanathan 2002).
In a sample where only two categories exist the basic idea of this concept is to find a vector, a so-called hyperplane, which separates the objects in two categories (Pang, Lee, and Vaithyanathan 2002). The main advantage of this method is that no parameter fine-tuning is necessary because SVM automatically finds the best parameter for the classification. Also in high dimensional feature spaces a manual feature selection is not needed, because it is done automatically by SVM (Kotsiantis 2007).

3.3.1.2 Part-of-speech tagging

Part-of-speech tagging, or POS tagging, is the process of assigning a part of speech to a given word out of a corpus. The various parts of a speech are classified with descriptors, or tags to input tokens. They are useful because they provide meta information to words and its neighbors (Jurafsky and Martin 2014).
Part of speech taggers are programs that use different methods and types of information like dictionaries or lexicons, pre-tagged sample documents and rules in order to conduct this process (Voutilainen 2005).

One differentiates between two types of taggers. Firstly, rule-based taggers use a dictionary or lexicon and hand-written rules in order to distinguish between ambiguous words like ‘light’. These taggers have been developed since the late 1950s. One of the first part-of-speech taggers was part of the parser in Zellig Harris’s ‘Transformations and Discourse Analysis Project’ (TDAP), implemented at the University of Pennsylvania (Jurafsky and Martin 2014).

Secondly, from the mid-80s on, stochastic taggers have been developed. The most important to name are taggers using ‘Hidden Markov Models’ (HMM) and cue-based models using ‘Maximum Entropy’ methods or decision tree models in order to combine probabilistic features. Taggers with HMM choose a tag sequence that maximizes the product of word likelihood and tag sequence probability. They also require a lexicon, but only untagged text for training the tagger. The problem of disambiguation is solved in using the most probable tag. In the example ‘Janet will back the bill’, the word ‘back’ will be tagged as a verb (VB) rather than an adverb (RB), adjective (JJ) or noun (NN). This is because the probability of a verb is much higher than the other options in this context (Jurafsky and Martin 2014).
In adding a variety of learning algorithms (e.g. with the Baum-Welch (EM) algorithm), HMM taggers have been further improved (Brill 1995; Cutting et al. 1992; Church 1988).

For training and testing part-of-speech taggers, there are three most used corpora for the English language. The ‘Brown Corpus’, published in the United States in 1961, consists of one million sample words extracted from 500 written texts of different genres. The ‘WSJ Corpus’ contains one million words published in the ‘Wall Street Journal’ in 1989. Moreover, the ‘Switchboard Corpus’ consists of two million words of telephone conversations collected from 1990 to 1991. These text corpora were created by running an automatic POS tagger and hand correcting the errors (Jurafsky and Martin 2014).
For the English language, there are several tagsets that differ in their level of detail. The ‘Brown Corpus’ has 87 different tags, whereas the most commonly used one is the ‘Penn Treebank’ tagset with 45 tags (small), 61 tags (medium) or with 146 tags (large) for the British national corpus:

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	*$*
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	*#*
PDT	Predeterminer	*all, both*	"	Left quote	(' or ")
POS	Possessive ending	*'s*	"	Right quote	(' or ")
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

Figure 9: Penn Treebank part-of-speech taste (Jurafsky and Martin 2014)

Figure 9 displays the small version of the 'Penn Treebank' part-of-speech tagset. Most modern language processing in English uses the 45-tag 'Penn Treebank' tagset. The tags are usually placed behind each word or punctuation. They are separated by a slash. One example for a tagged sentence would be the following:

> ***The**/DT **grand**/JJ **jury**/NN **commented**/VBD **on**/IN **a**/DT **number**/NN of/IN **other**/JJ **topics**/NNS **.**/.*

POS taggers are useful for information retrieval in general, text to speech applications and word sense disambiguation in particular. Also abstract levels of analysis profit from consistent low-level information like parts of speech. Therefore, a good tagger could serve as a preprocessor for parsing or other further text mining (Jurafsky and Martin 2014).

Furthermore, information technology applications make usage of POS taggers for text indexing and topic retrieval. This implies that nouns and adjectives are more suitable to be used as index terms than adverbs, verbs or pronouns. (Voutilainen 2005)

3.3.2 Unsupervised machine learning methods

In this chapter, we explore the use of unsupervised machine learning, also known as cluster analysis. In contrary to supervised classification methods, unsupervised machine learning does not rely on pre-labeled data sets for training the algorithm. This absence of category or label information clearly distinguishes clustering from supervised machine learning and makes it an even more challenging task (Jain 2010). A second significant difference is that even though "[…] most clustering algorithms are phrased in terms of an optimality criterion there is typically to guarantee that the globally optimal solution has been obtained" (Gentleman and Carey 2008, 137).

In general, cluster analysis aims at separating data into meaningful subgroups. The number of these subgroups and their specific composition is usually unknown in advance (Fraley and Raftery 1998). Data in the same subgroup should be homogenous, while patterns in different clusters are not (Xu and Wunsch 2005). Clustering usually follows either a heuristic or more formal approach based on statistical models (Fraley and Raftery 1998).

Literature dealing with unsupervised learning describes two basic forms of cluster analysis: 'hierarchical clustering' and 'partitioning' (Gentleman and Carey 2008).

Hierarchical methods organize data into hierarchical structures, mostly in forms of a binary tree or dendogram. This clustering method can be further divided into agglomerative and divisive approaches. Agglomerative clustering starts with *N* clusters containing one single object each. After a series of merge operations, where in every step the two most similar clusters are combined, a cluster hierarchy is formed. In contrary to that, divisive approaches start with all data situated in a single cluster. During its application the most heterogeneous objects are divided until every single pattern has its own subgroup (Xu and Wunsch 2005).

Compared to hierarchical clustering, partitional algorithms do not impose a hierarchical structure. Besides an *n x n* similarity matrix used by hierarchical methods, partitional clustering "[…] can use either an *n x d*-dimensional feature space, or an n x n similarity matrix" (Jain 2010, 653). In order to apply partitioning algorithms, the required number of clusters has to be

determined in advance. In a series of iterations, patterns are moved from one subgroup to another, starting from an initial partition. This process is repeated until a optimal partition is attained or a specific number of pre-defined iterations is made (Gentleman and Carey 2008).

The following chapter pays further attention to partitioning algorithms. In particular, this thesis focuses on the application of k-means methods for clustering data. As parts of our analysis and the search for lead user characteristics rely on this unsupervised machine learning approach, we give a solid introduction to k-means clustering.

3.3.2.1 K-means clustering

K-means clustering is the most commonly used and known partitional algorithm in unsupervised machine learning. This is because its simplicity, efficiency, ease of implementation as well as its empirical success convince in various research cases (Jain 2010). The ultimate goal of k-means clustering is to continuously relocate objects into different clusters in a way that the distance of objects with-in the clusters is minimized. In other words, an iterative algorithm is used to partition sample data into k groups such that the sum of squared errors or distances between the samples and the empirical means or centers of a cluster is at a minimum (Gentleman and Carey 2008).

Jain (2010) shows how to mathematically describe k-means clustering using following equations. $X = \{x_i\}, i = 1, \dots, n$ is the set of n d-dimensional patterns that should be clustered into k clusters, $C = \{c_k, k = 1, \dots., K\}$. As described above, k-means clustering aims at computing the minimum distance between each point in the cluster and the empirical center. Defining μ_k as the center of cluster c_k, the squared error between μ_k and every pattern in the cluster is calculated:

$$J(c_k) = \sum_{x_i \in c_k} \|x_i - \mu_k\|^2$$

Finally, the algorithm aims at minimizing the sum of the squared error over all *k* clusters:

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \|x_i - \mu_k\|^2$$

In the course of its application, k-means clustering demands three user-defined parameters: the number of clusters K, cluster initialization, and distance metric. Jain (2010) sees the number of clusters K as the most critical criterion, as k-means runs independently for different input values. As a consequence, it is up to the user to define a meaningful number of clusters K before initiating the algorithm. Second, different cluster initialization can result into different final clusters. This is due to the fact, that k-means only approaches local minima. The selection of different partitions as starting points for running the algorithm and selecting the result with the smallest squared error is one approach how to overcome the local minima. Finally, the user-selected distance metric significantly influences cluster analysis. Researchers have applied different distance metrics in a variety of circumstances that result in very diverse findings. Typically, k-means clustering uses the Euclidean metric for calculating the distance between the samples and the cluster centers (Jain 2010).

After having defined the the prerequisites for k-means clustering, the actual application can be initiated. Therefor, Jain and Dubes (1988) describe an approach for iterative partitional clustering using k-means. The main steps of their approach are:

1) An initial partition with *K* clusters has to be defined
2) Each object to be assigned to its closest center, generation of new partition
3) Calculation of new cluster centers; repeat step 2 and 3 until cluster association stabilizes

Figure 10 and 11 illustrate the typical process of k-means clustering with two-dimensional input data and three clusters.

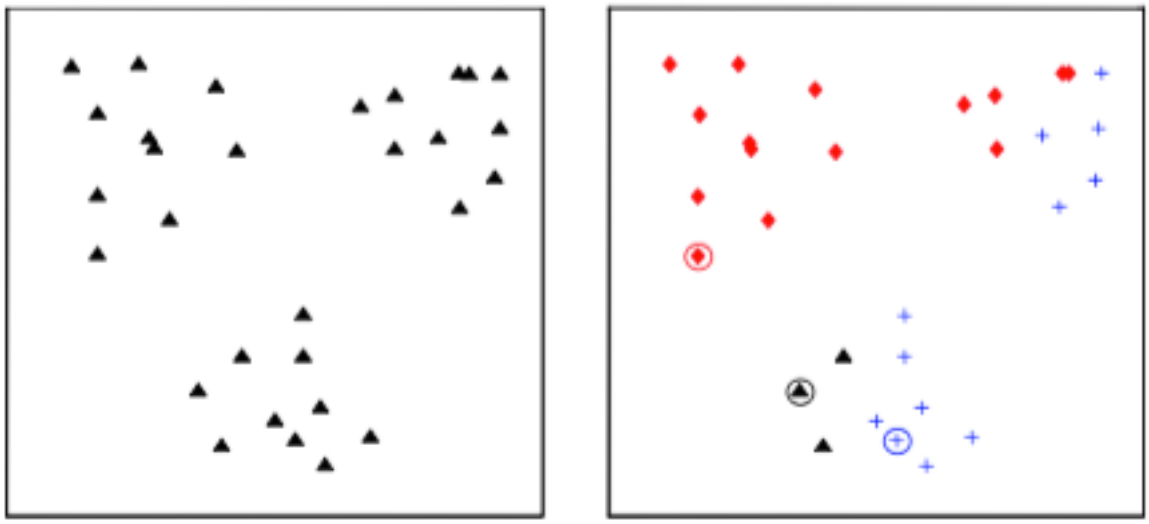


Figure 10: K-means clustering process step 1-2 (Jain 2010)

In order to apply the k-means algorithm, a number of clusters *K* has to be defined first. The initial partition contains of *K* seed points that are either pre-defined or selected randomly from the matrix. After the selection of the cluster centroids, the algorithm runs the initial partition or clustering by allocating each data object to the nearest seed point. The centers of the resulting clusters represent the initial cluster centroids (Jain and Dubes 1988).

The initial assignment is illustrated in figure 10, where three seed points are selected as cluster centers and data points are allocated to clusters.

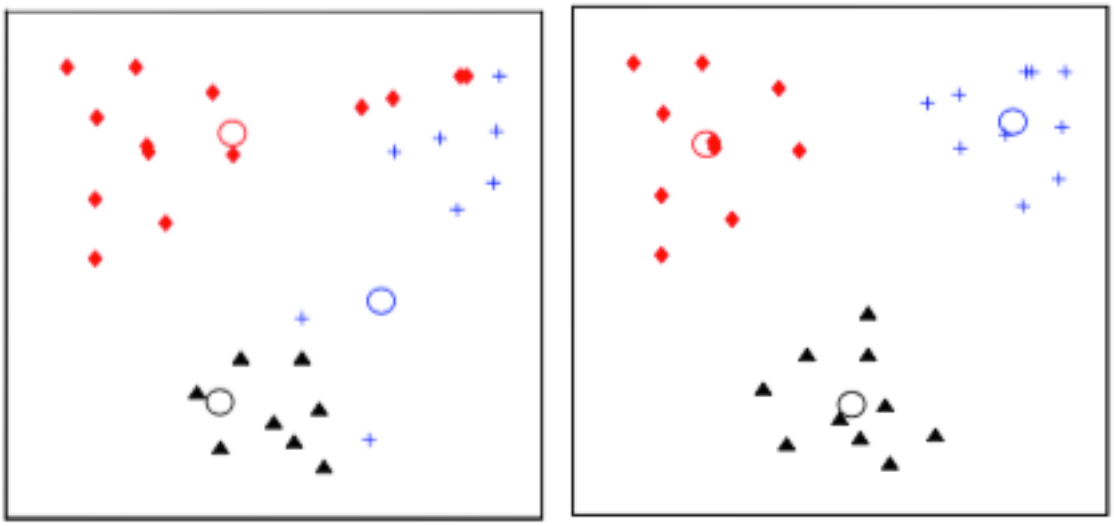


Figure 11: K-means clustering process step 3-4 (Jain 2010)

During the following steps the initial partitions are updated by reassigning the objects to clusters in order to minimize the square-error. By using the Euclidean metric, the distance between patterns is calculated and new cluster centers are defined after the reassignment of all patterns. The points in the matrix are only reassigned if the sum of distances is reduced (Martínez-Álvarez et al. 2007).

The algorithm stops when it reaches convergence, i.e. when the criterion function cannot be improved by further clustering. The results are k groups of clusters where the distance of each object with-in a certain cluster is at a minimum. The ultimate goal of k-means clustering is reached and each cluster represents objects that show similarity according to pre-specified criteria (Jain and Dubes 1988).

4 Context

This chapter presents the context of this work and describes the underlying examples for this study. By providing details about Android software development in general and the specifics of the chosen online community ‘xda-developers’, we help the reader to understand the application of the concepts and methods used in the later parts of this work.

4.1 Android software development

The following explanations introduce the mobile platform ‘Android’. We particularly focus on common facts and its historical development. Afterwards, we show how users started to engage in Android software development.

4.1.1 Android platform

Android is a software stack including a Linux-based operating system, middleware and key applications (apps). It is primarily designed for mobile devices and is developed by Google as an open source project (Gandhewar and Sheikh 2010).

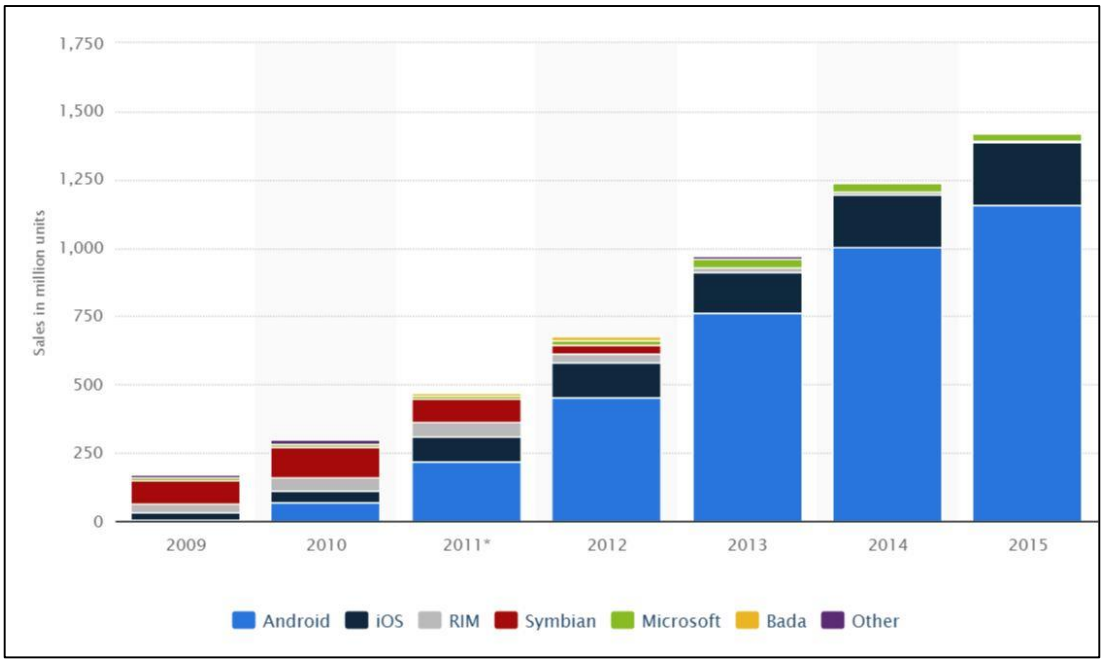


Figure 12: Global smartphone sales by operating system (Statista 2016)

Figure 12 shows that Android is the market leader for mobile operating systems since 2011. In 2015 it had a market share of about 80% (IDC Research 2015).

4.1.2 The history of Android

Initially, Android was developed by Android Inc., a company that was acquired by Google in 2005 (Owen 2010).

In 2007, Google presented 'Android OS' as an open source project and launched the 'Open Handset Alliance'. This group of hardware, software and telecommunication companies joined forces in the development of new standards for mobile devices (Open Handset Alliance 2007).

Today, Android is used as the main mobile operating system by most of the hardware manufacturers. In the recent years, it was further developed for TVs (Android TV), cars (Android Auto) and wearable devices (Android Wear) (Android.com 2016).

One major factor that boosted the impressive success of Android had been its setup as an open source project. The underlying source code was made publicly and third party developers were able to fully analyze and understand it. That permitted feature comprehension, bug fixing, further improvements regarding new functionalities and lastly, porting the system to new hardware. Furthermore, its Linux kernel-based architecture model allowes taking advantage of the knowledge and features offered by Linux (Gandhewar and Sheikh 2010).

However, for securing further control of the platform, Google started to convert open source Android apps back to Google specific closed source apps. "While you can't kill an open source app, you can turn it into abandon ware by moving all continuing development to a closed source model" (Amadeo 2013, 1). Most of the major Android open source functions like messaging, search, camera and music were converted into Google services and rebranded as Google Hangouts, Google Search, Google Camera and Google Play Music.

4.1.3 Android software development and hacking

Soon after the first release of Android smartphones, a large community of developers and enthusiasts evolved and used the open-source code as a basis for community-driven projects.

They added new features, improved performance or ported Android to devices that were initially shipped with different operating systems.
The community created the so-called 'Custom-ROMs', a modified version of the standard Android operating system (Stock ROM) (Russakovskii 2010).
The main advantages of 'Custom-ROMs' lie in the higher update frequency and faster upgrade to the latest Android version, since new features and improvements are directly pushed from the developers to the end user.

In the context of innovation theory, it is highly relevant to look into Android development because of the following reasons: Android and the smartphone industry in general are facing a fast product lifecycle that demands for continuous product improvements. Furthermore, there is a high involvement of users since Android was developed as an open source project. Lastly, there is proof that functions and improvements that were initially developed by users and distributed in online forums were later implemented into the standard version of Android (Dobie 2015; Finley 2013).

4.2 The xda-developers online forum

For the successful application of SMA, it is highly relevant to select appropriate data, as the results of the analysis can only be as good as the quality of the source (Boyed et al. 2012).

As there exist an enormous diversity of social data on the Internet, it was crucial for the success of our project to carefully select data sources and a fitting subject. This is why we decided to focus on online forums.
Online forums contain content of a specific domain or topic. They usually have moderators that guide conversations and delete off-topic comments or posts. Therefore, one can assume that the quality of content is higher than on other social media platforms. Furthermore, information is shared in a structured way and generally accessible for everyone, since the majority of online forums is of public nature (Christensson 2011).

For our study, we chose the xda-developers forum, which is available under following internet address: http://forum.xda-developers.com/.

We selected the the xda-developers forum for following reasons.

First, there exist millions of 'Bulletin Board Systems' (BBS) and online forums on the Internet. In order to make our scrapping tool compatible to the majority of these communities, we chose a forum based on the 'vBulletin Board' software, as it is one of the most popular systems for online forums (vBulletin 2016).

Second, with its 7,265,731 members (in May 2016) the xda-developers community is the 5th biggest English speaking online forum and the biggest when it comes to discussions about technology and products in the mobile phone category. The site had an average of 1,650 new threads and 40,000 posts per day in 2014 (XDA Changelog 2014).

For that reasons we assumed that topics in the forum were highly up to date and there was sufficient data to analyze even in very specific sections.

Porter (2004) proposed to classify virtual communities according the 'Five Ps of Virtual Communities' that are Purpose, Place, Platform, Population and Profit Model.

The purpose of the forum is to provide developers and users of mobile devices a communication platform and a room for discussions and information exchange. These interactions take primarily place online. However, the provider of the forum organizes yearly conventions. As platform, the forum uses a bulletin board system that provides an asynchronous communication experience. The population grew massively in the last years and consist of users from all over the world. They share the common interest of mobile devices. The profit model of the forum is based on online advertisements in form of banner ads.

The xda-developers forum was created by two Dutch developers in 2002. Initially, the forum focused on hacking smartphones of the o2 xda brand. Later they started to offer sections for other models, especially for devices running on Microsoft Windows mobile operating systems. In 2006, they shifted to the vBulletin system and re-launched the forum (Finley 2013).

In 2010, xda-developers was bought by JB Online Media LLC, an American company owned by Joshua Solan. Solan himself was initially a user of the forum. Before buying the forum, his app 'Themer' (it allows phone users to switch their phone theme very easily) became very popular amongst the user base (Finley 2013). Having Porters Typology of virtual communities (Porter 2004) in mind, one can state as conclusion that the forum initially was 'member-initiated' with a social relationship orientation since it was first started by two developers as a hobby. However, we argue that today one can classify this forum as 'organization-sponsored' since it generates revenue.

4.3 Data selection

In this section, we illustrate what data can be retrieved from online forums. The following figure displays an example of the first post in a thread.

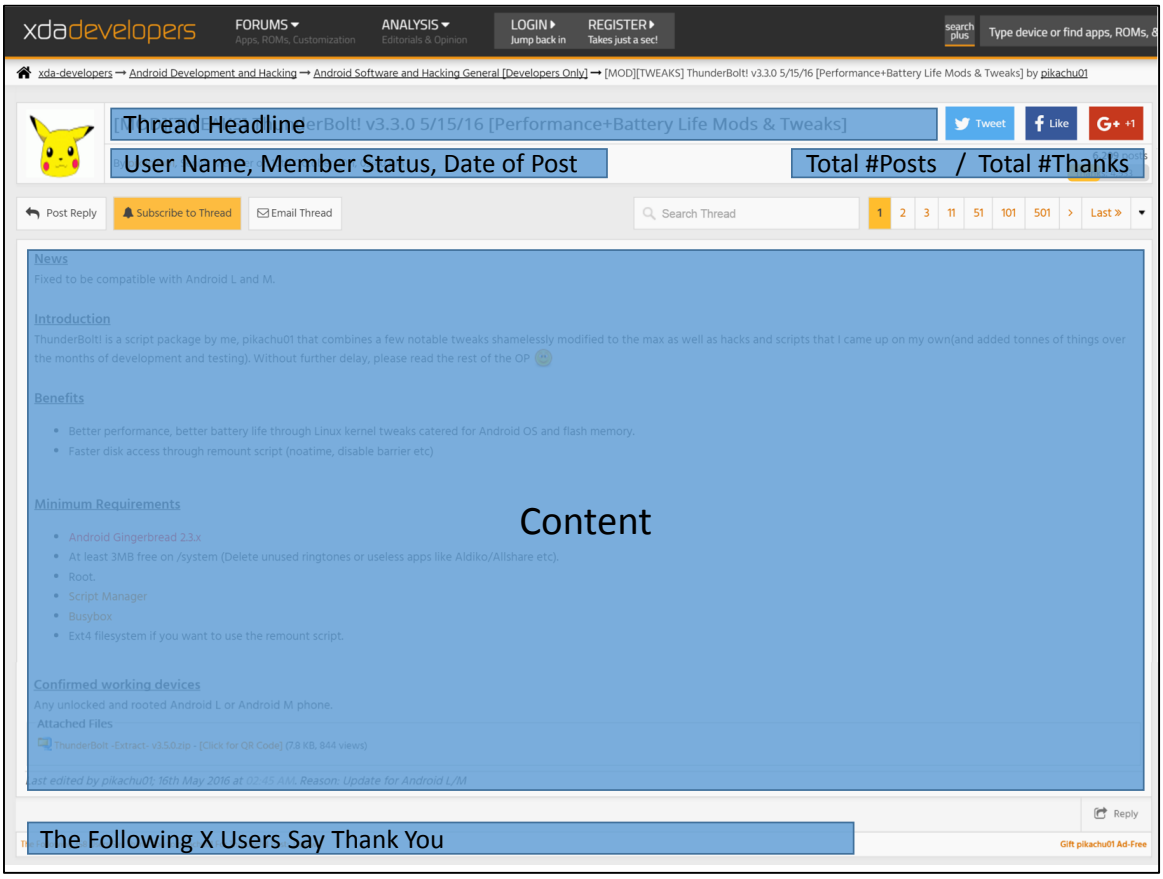


Figure 13: Screenshot of the xda-developers forum with highlighted areas

We divided the available data between user specific and post specific information. The user specific dataset contains the username, the member status, the forum join date, user location (if provided), the mobile phone operator (if provided), the total number of posts and the total number of thanks received from other users.

The post specific dataset contains the content of the post. It is HTML formatted and can include text, code, images, gif animations and file attachments. However, for our study we solely accessed the plain text of the post, the time and date of the post as well as the number and all names of users that said 'Thank You' to this specific post. The post usually contains quotes of posts from other users. We further accessed the name of the quoted user and the content.

5 Analysis

This chapter applies the described theoretical approaches to the presented context and proposes a solution for identifying lead users in an online forum.

We start our explanations with describing the general system set-up, including the overall process and the technical requirements. After having gathered the necessary data from the xda-developers community, we started exploring the dataset for each of the introduced lead user characteristics. This chapter introduces tools and procedures to analyze 'involvement', 'opinion leadership', 'dissatisfaction', 'ahead of trend' and 'product related knowledge'. Afterwards, the result of each analysis step is aggregated and we present the identified lead users. Lastly, we describe how we conducted a mass screening search among the selected user population in order to enhance the validity of our study.

5.1 Identification of lead user characteristics with social media analytics

In order to run our analysis and to identify lead users among the online forum members, we adopted Stieglitz et al.'s (2014) 'Social Media Analytics Framework'. Based on their approach we followed a process model that is divided into four major steps.

We started with the gathering of data from the web, followed by storing this information into a suitable database. Afterwards, the actual analysis was conducted by applying suitable calculation engines. Finally, we generated an output in form of a report containing a list of users that possess the investigated characteristics.

In the following, we present our tool set-up. In accordance to Stieglitz et al. (2014), we designed several tools that serve the different tasks.

Figure 14 illustrates the overall process (on the left side) and the corresponding setup and usage of the tools and methods for each process step (in the middle / right side).

During the collection phase, we parsed data from the xda-developers forum using Java and the jsoup library. In the following step, the information was stored in a relational database (MySQL). We processed structured and unstructured data from the online forum. The analysis phase applied the chosen approaches and methods for identifying lead userness among the forum members. We used Python tools for sentiment analysis and part-of-speech tagging as well as R for cluster analysis and text mining.

Each of our developed tools connected to the database in order to get the required variables. Afterwards, the information was processed and the results were stored back into the database.
In the last step, the results of each approach were combined in order to get an aggregated list of identified lead users.

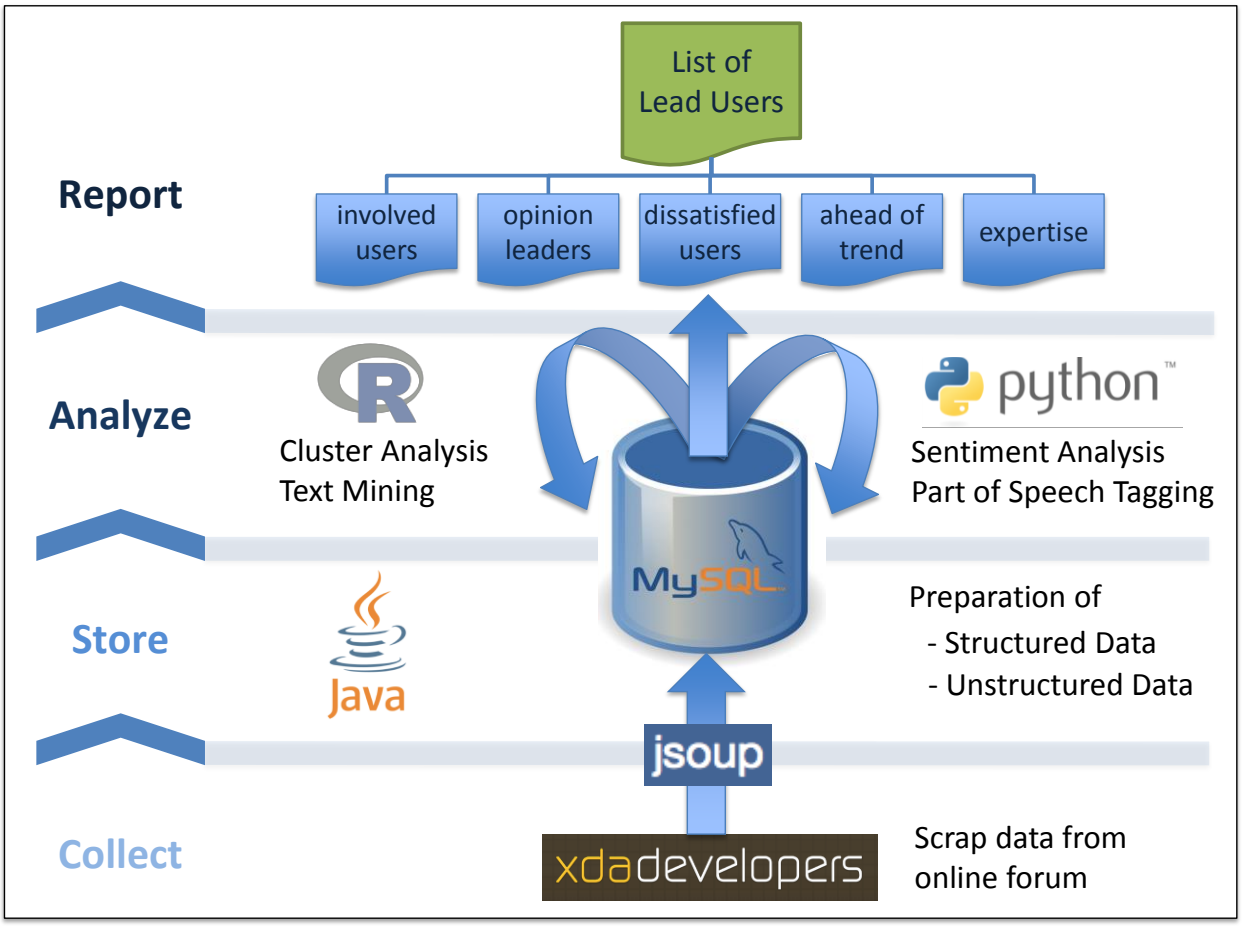


Figure 14: Process overview and applied methods

In the following, we present the technical preconditions as well as the detailed approach of data collection and data preparation.

5.1.1 Project setup, data collection and storage

The analysis was conducted with the use of a Windows 8.1 (64bit) machine running an Intel Core i7 processor and 8GB of RAM. We further used 'WAMP Server' in order to host a local MySQL database.

For our Python applications we used a Python (2.7) installation containing the following packages: setuptools (7.0), pip (1.5.6), Mysql-connector-python (2.1.3), pattern (2.6), scipy (0.17.0), numpy (1.11.0), nltk (3.2.1) with the 'Brown Corpus' and 'Punkt Corpus'. As a precondition for applying Python on Windows, we needed 'Visual Studio 2015' with 'Python Tools' and the 'Visual Studio C++' compiler for Python.

For our Java Programs we used 'Eclipse IDE for Java Developers', 'Version Mars.2 Release' (4.5.2) with the following additional libraries: mysql-connector-java (5.1.3) and jsoup (1.8.1)

For clustering and text mining, we used R version 3.3.0. We imported the following libraries into R: tm, qdap, qdapDictionaries, dplyr, RColorBrewer, ggplot2, scales, Rgraphviz, wordcloud and RMySQL.

All the tools and libraries that were applied are publicly available as open source code.

The following figure displays the package explorer of 'Java Eclipse'. In total, we implemented seven small Java programs for the various process steps of data collection (A1 and A2), conversion of data (B1-3), analysis of relative data (C1) and preparing data for text mining (D1). In addition, we implemented two further public classes, datetime.java for conversion of dates and DB.java for the connection to the MySQL database. In total, we wrote 1.144 lines of code.

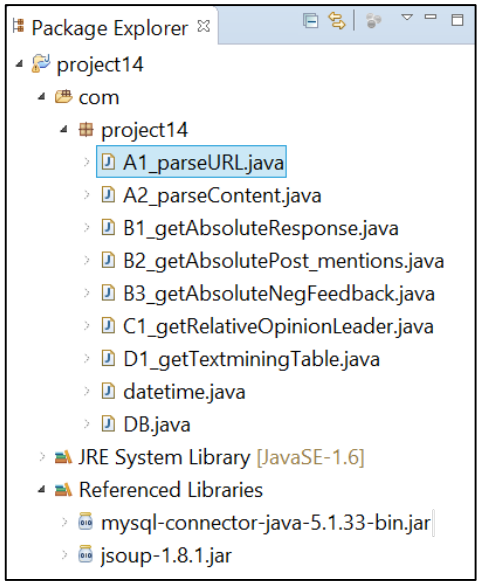


Figure 15: Package explorer of Java Eclipse

In the following, we present parts of our Java, Python and R code. The full source code can be found in the appendix of this thesis.

For collecting the desired data, Stieglitz et al. (2014) propose 'keyword-related', 'actor-related' and 'URL-related' tracking approaches. These approaches can be applied using APIs, RSS feeds or HTML parsing as downloading method.

Social media services are often accessible through native APIs. Platforms like Facebook, Instagram or Twitter provide APIs that allow users to directly load data. Accessing APIs is usually done using keyword-related approaches (Stieglitz et al. 2014). In order to call an API, a distinct search term / keyword is needed. Some APIs also allow actor-related approaches to access data. In this case, the API returns all conversations and contacts of one distinct user.
However, data loading is mostly limited, e.g. Twitter allows 60 API calls per hour and Instagram allows 5000 API calls per hour. There is a general trend towards intensifying the restrictions and limiting free access (Twitter 2016).

Blogs can be accessed via RSS. RSS stands for Rich Site Summary. It is often utilized to publish frequent updated information. Most of the news sites or other streaming services provide RSS

feeds. RSS feeds can be accessed using ‘keyword-related’ and ‘URL-related’ approaches (Liu, Ramasubramanian, and Sirer 2005).

Websites that do not provide APIs or RSS feeds can be accessed with web parsing or scrapping. Web scrappers usually offer the same functionalities as search engine web crawlers that browse the Internet for available content. They are not dependent on API rate limits and are completely free of charge. However, web scrappers often require high implementation effort. In addition to that, changes of the website layout or structure can lead to failures. Web parsing can be conducted with each of the three collecting approaches. However, it is most commonly combined with URL-related methods (Malik and Rizvi 2011).

Since we did not have direct access to the servers of the forum and neither suitable APIs were available, we made use of a URL-related tracking approach using the HTML parsing method.
Firstly, we accessed the xda-developers forum and downloaded the most active threads of the categories ‘Android General’, ‘Android Themes’, ‘Android Software Development’ and ‘Android Software and Hacking General [Developers Only]’. We selected this group of topics because we observed that discussions in these categories covered content dealing with new Android features. Besides that, the majority of other categories was mainly dealing with discussions around single mobile devices instead of comprehensive Android topics.

In order to collect data from the forum, our Java program applied an open source HTML parser called jsoup. Jsoup acts like an automated browser. It can open websites and download pre-defined datasets (Hedley 2016).

We conducted the HTML parsing in two major steps. Firstly, we accessed the unique URL of each post and stored it to the database (program A1_parseURL.java). Secondly, the gathered URLs were used by the program ‘A2_ParseContent.java’ to access the actual content and to download it.

Run Java Program: A1_parseURL.java
Gathered Parameter: Unique post URLs of the xda-developers threads

```
//Jsoup connection
Document doc = Jsoup.connect(html +"&page="+ pnum)
        .userAgent("Mozilla").timeout(20000).get();

//Define Locations in HTML

// For-Loop for Post-entries
for (int i = 0; i <= 9; i++){
    // 10 Entries per page. After 10th (i=9) post Page-ID (pnum) raises one
    if (i == 9) {pnum=pnum+1;}

    //Postinfo
    Elements postcount = doc.select("a[class=postCount]");
    Elements postbit = doc.select("a[class=postbit-anchor]");
    //Get Data from HTML
    Element printpostcount = postcount.get(i);
    Element printpostbit = postbit.get(i);
```

This Java program checked the total number of posts and pages of a thread. Usually, there were 10 posts on each page. Each post was described by a unique post ID that we gathered and stored into the database. Afterwards, a new URL was generated by concatenating the root URL of the xda-developers forum with the unique post ID.

For the process of gathering and storing the newly generated post URLs, we made use of the jConnector library. This library is provided by Oracle and allows Java code to connect to a MySQL database and to run SQL statements (Oracle 2016).

The following code shows the part of the program that initialized the database connection.

```
// Connection to MySQL DB
public static void connection(){
    try {
        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("driver ok");
    } catch (ClassNotFoundException e){
        e.printStackTrace();
    }
}
public static void ConnectionToMySql (String ID, int postcountint, String printpostcount){
    connection();
    // Connection parameters
    String host = "jdbc:mysql://localhost:3306/masterthese";
    String username = "root";
    String password = "CBS";
    // connect
    try {
        Connection connect = DriverManager.getConnection(host, username, password);
        System.out.println("connected to mySQL db");
        String sql = "INSERT INTO URLtable(ID, PID, URL)VALUES (?,?,?)";

        PreparedStatement statement = (PreparedStatement) connect.prepareStatement(sql);

        statement.setString (1,ID);
        statement.setInt    (2,postcountint);
        statement.setString (3,printpostcount);
        statement.executeUpdate();

        statement.close();
        connect.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
```

Afterwards, the program added one new line for each unique post URL to the MySQL table 'URLTABLE'.

```
    //Write to mysql db

    ConnectionToMySql (ID, postcountint, "http://forum.xda-developers.com/"+printpostcount.attr("href"));
```

After having stored the unique URLs of each post, we started accessing the actual forum content. Therefore, various kinds of structured and unstructured data were available. We only focused on text and disregarded images, attachments or pieces of presented code.

During the scrapping process, we gathered the following information from the website:

Datafield	Description
post_id	Unique ID of every post. This key was needed for identification and analysis purposes
thread_id	Unique ID of every thread
post_timestamp	Date and time of a post
post_post	The actual content of the post
post_length	During the process of downloading, the length of the text was measured and stored in this data field. The length of made quotes was deducted and only the text length of new content was measured.
post_thanks	The number of 'thanks' replies from other users
member_name	The name of the editor of the post
member_type	The forum has its own classification of its members. Since we did not know the conditions of this ranking, we did not consider this information for our analysis.
member_thanks	The number of total 'thanks' replies the user received from other users
member_posts	Total number of posts of a user
member_joindate	Date the user joined the community
post_quotes	Number of quotes the member made in his posts
post_quote	The original post that got quoted
post_quote_name	The name of the quoted user

In order to obtain the dataset shown above, we ran a second Java program.

Run Java Program: A2_ParseContent.java

This program represents the core element of the data acquisition part. It can process all stored threads from the previously filled 'URLTABLE'. As input parameter, this program requires the thread_ID of the corresponding thread.
The following code shows the jsoup statements used for obtaining the data from the web.

```
//Postinfo
Elements postcount = doc.select("a[class=postCount]");
Elements dates = doc.select("span[class=time]");
Elements posts = doc.select("div[id^=post_message_]");
Elements thanks = doc.select("div[id^=post_thanks_box_]");

//Userinfo
Elements posternames = doc.select("a.bigfusername");
Elements memberinfo = doc.select("div[class=postbit-userinfo-cell]"); //for member-status
Elements memberthanks0 = doc.select("span[class=userThanksText]");
Elements memberposts0 = doc.select("div[class=pbuser user-posts]");
Elements memberjoin0 = doc.select("div[class=pbuser user-joindate]");
//Element countryimg = doc.select("div[class=moreUserInfo] img").first();
```

After downloading the posts, some modification and preprocessing was made. In the following, we shortly illustrate the cleaning and reformatting of a post's timestamp.

SQL queries can only calculate times if they were in the correct 'datetime' format. However, the format that we downloaded looked like this: 'posted on 15th May 2016, 06:35 PM'. For further processing, however, SQL requires a format like this: 'YYYY-MM-DD HH:MM:SS'.

Following code demonstrates the conversion of the variable 'datetime' into the correct time format.

```
// Convert Date to SQL readable format
String datestring = printdates.text();
datestring = datestring.replaceAll("([0-9]{1,2})(st|nd|rd|th)(.*)", "$1$3");
System.out.println("Timestamp reduced: "+ datestring);

String inputformat ="dd MMMMM yyyy, hh:mm a";
String format1 ="yyyy-MM-dd HH:mm:ss";
//format needed for SQL: YYYY-MM-DD HH:MM:SS
SimpleDateFormat sdf = new SimpleDateFormat(inputformat, Locale.ENGLISH);
SimpleDateFormat sdf1 = new SimpleDateFormat(format1);

try {
    Date date = sdf.parse(datestring);
    System.out.println(date);
    System.out.println(sdf1.format(date));
    sqldate = (sdf1.format(date));
} catch (ParseException ex) {
    ex.printStackTrace();
}
```

Afterwards, further prepossessing of the datasets was conducted. Our developed Java tool processed the quotes, extracted usernames, modified the 'Thank You' results, processed the member information and modified the member join dates.

Finally, the program inserted the information gathered from each post into the database. Following code shows shows the beginning of the statement.

```
//Write to mysql db
ConnectionToMySql (IDx, IDint, postcountint, threadid, ""+sqldate+"", ""+printposts.text(), post_topics, post_pol, post_sub, length,
```

In the week from the 7th to the 14th of May 2016 we downloaded 100.000 posts in total with a download speed of 1000 posts per hour. The number of unique users was 14901. The timestamp of the first post was '2009-01-06 20:36:00' and of the last post '2016-05-11 23:29:00'.

As mentioned before, the Java program already preprocessed and modified the downloaded information. We further ran several SQL statements in order to clean and prepare the data for further processing.

Having a prepared and cleaned dataset consisting of 100.000 posts, we could start the actual analysis by applying several computer-aided methods for identifying lead user characteristics and ultimately lead users in the xda-developers community.

5.1.2 Involvement

As stated in the literature review, several empirical studies (Sawhney, Verona, and Prandelli 2005; Jeppesen and Laursen 2009) show that lead users tend to actively participate in online communities. Furthermore, they show high involvement in their specific field of interest (Lüthje 2004). Therefore, this chapter shows how we identified forum members that were highly involved in the ongoing online discussion.

Amaro and Duarte (2015) proposes to measure involvement based on the total time spent in the online community and by the amount and frequency of content creation. Based on this approach, we separated active users from passive users by defining following own criteria for 'involvement'.

Firstly, we measured the user's length of membership in the community (length of membership criteria). We filtered out all newly registered users. i.e. members with a join date within the past three months. Secondly, we deleted members who wrote less than five posts within the last 10

months (frequency criteria). Lastly, we deleted all members that were never quoted by other users, assuming that these individuals do not post relevant content (relevancy criteria).

In order to filter the user data, we applied several SQL statements. First, all members and their counted sum of posts were copied from our main table (mastertable) into a new table (analysis).

```
INSERT INTO analysis (member_name, countofposts)
SELECT member_name, COUNT( * )
FROM masterdata
GROUP BY member_name
ORDER BY COUNT( * ) DESC;
```

In order to calculate the length of membership, the member join date and the date of the user's last post were detected.

```
UPDATE analysis a INNER JOIN (
SELECT member_name, TIMESTAMPDIFF (MONTH, member_joindate, member_lastactivity)
time_diff FROM analysis)
b ON a.member_name = b.member_name
SET a.time_diff = b.time_diff;
```

Having calculated the length of membership, we were able to compute the relative frequency score. The frequency score sets the count of user posts in relation to the length of membership. This is represented by the following SQL statement. We received the variable 'posts per 10 months'. The value for one month would have been too small to further processing.

```
UPDATE analysis a INNER JOIN (
SELECT member_name, countofposts/time_diff_months*10 freq FROM analysis)
b ON a.member_name = b.member_name
SET a.frequency = b.freq;
```

As stated above, we defined certain criteria in order to determine whether a user is actively involved on a regular basis. Users that did not fulfill the defined criteria were deleted in the following steps:

1) DELETE from analysis where member_joindate > '2016-02-01 00:00:00'

→ 30 Users were filtered out.

2) DELETE from analysis where response = '0'

→ 5413 Users were filtered out.

3) DELETE from analysis where frequency < '5'

→ 6287 Users were filtered out.

We identified 3,171 community members as fulfilling the 'involvement' criteria for lead userness. In total, 14,901 users were deleted. As we required active user involvement in order to successfully conduct the next analysis, we only considered the resulting 3,171 members for all further steps.

5.1.3 Opinion leadership

As described in the literature review, opinion leadership is another distinctive characteristic that lead users possess. Opinion leaders act as contact point for others in the search for advice and exert personal influence on other persons (Rogers and Cartano 1962). In lead user research, opinion leadership is particularly significant in the post-launch phase of a new innovation, as lead users are said to fuel then diffusion of new developments (Urban and Von Hippel 1988) and serve others as role models in the adoption of products and services (Schreier, Oberhauser, and Prügl 2007).

Previous work in the field of identifying opinion leaders recommended two traditional techniques: the 'self-designation method' and the 'sociometry method'. The self-designation technique asks participants to assess whether others regard them as influential. In contrary to that, the sociometric method makes use of a network approach in which every member of the network is asked who he or she consults for advice (Rogers 2010).

As both methods are very manual and time consuming, we did not regard them as valuable for our own lead user identification approach.

More recent work in the area of identifying opinion leadership focuses on social network analysis (Bodendorf and Kaiser 2010) or the user's interest space (Zhai, Xu, and Jia 2008).

Social network analysis not only considers the single user's opinion but also analyzes the communication relationship between users. The user's interest space method focuses on knowledge about the user's membership in different online communities with diverse topics. This approach is useful if the research domain is not clearly defined and user activity in several communities wants to be regarded.

Our own application is partly inspired by both research techniques. In the case of the xda-developers community, we do not need to know about the users memberships in other forums, as the search field is precisely narrowed down. In order to unveil the opinion leadership characteristics amongst the online community, we followed a k-means clustering approach motivated by Hudli, Hudli, and Hudli (2012).

This method recognizes the content and the activity of the community members and allows the creation of an individual online profile of each user. The so-created user profiles can be analyzed using k-means algorithms in order to identify homogeneous groups of observations. The ultimate goal is to receive a cluster that contains community members that act as opinion leaders in the xda-developers forum.

Hudli, Hudli, and Hudli (2012) intensively studied the dynamics of Internet based discussion boards and recognized that certain user behavior clearly characterizes opinion leadership. According to their research, opinion leaders spend a significant amount of time in online discussion forums. Secondly, those users tend to show a high degree of engagement as they regularly post new messages and comments. Furthermore, they or their content is often quoted or referenced and other users regularly respond to their posts. Opinion leaders' messages are frequently met with positive feedback from other individuals, with very limited negative response. Lastly, they tend to write messages that are more detailed and they get involved in the discussion in a significant way.
These observations allowed us to define an individual online profile of each user. In accordance with Hudli, Hudli, and Hudli (2012) we chose a set of attributes that should describe each user regarding the different observations we made and data we gathered.

We selected following attributes that enabled us to determine opinion leadership behavior amongst the xda-developers online forum members:

- **Involvement**, i.e. the degree a user is involved in the discussion, measured by the amount of messages in the threads we examined
- **Frequency**, i.e. the number of posts within a given period of time
- **Degree of response**, i.e. the degree to which the messages of a user are responded by others
- **Number of references**, i.e. the degree to which a user is referred to in messages of other users
- **Positive feedback**, i.e. the degree of 'Thank You' responses a user receives
- **Negative feedback**, i.e. the number of feedback with negative polarity
- **Average size of messages of a user**

Some of the defined attributes could be collected through evaluating the usage pattern we downloaded for each community member. We ran several SQL queries to gather the needed data and to define each of the variables. For analyzing negative feedback given by other users, however, we made use of the sentiment analyzer initially built for identifying the dissatisfaction characteristic of lead user behavior. In the following, we portray how the variables are defined.

The involvement attribute is defined as the aggregated number of posts a user submitted in the threads we selected for our analysis. This slightly differs from the definition of involvement we used in chapter 5.1.2.
The frequency variable set the involvement score in relation to a certain period, i.e. it represents the number of posts a user wrote within a given time. We selected a time period of 10 months, as a lower value would had been too small for further processing.
In order to calculate the positive feedback variable, we extracted the 'Thank You' responses a user received. The xda-developers forum allows thanking a user for a submitted post by clicking a button. These 'Thank You' responses were crawled and stored in the analysis table. Afterwards, a relative score of these attributes was generated from the total number of positive feedback and the sum of posts of a particular forum member. This last step allows the

comparison between users, as not every member shows the same degree of involvement (posts submitted) and thus does not receive the same amount of 'Thank You' responses.
Measuring the negative feedback variable was tricky, as the xda-developers forum does not include a functionality to 'dislike' a user post. To get the negative feedback attribute we ran a sentiment analysis of all posts the individual user was mentioned or quoted.

Run Java Program: B3_getAbsNegFeedback

The program went over all posts and analyzed the polarity of each comment and user quote. In order to get the polarity score, the number of negative comments and quotes were counted. This absolute amount of negative feedback was then converted into a relative score dependent on the number of total quotes and mentions a user received. The set-up and actual code of the sentiment analyzer is described in the next chapter, where text polarity measurement is portrayed in terms of dissatisfaction identification.
For the 'number of references' attribute, we calculated the amount of messages in which a particular user was quoted or referred to.
In order to receive the average message size of a user's posts, the number of characters of each message was extracted using a SQL query.
In the pre-processing for the actual cluster analysis, each of the defined attributes was discretized and mapped on a scale of 1 to 1000. By following this approach, the ideal opinion leader would be represented by following scores: (1000 1000 1000 1 1000 1000 1000). The fifth dimension shows the degree of negative feedback, which would be the lowest for opinion leadership. The data was saved into a new SQL table and downloaded as a CSV file for further processing.

member_name	involvement	frequency	response	post_mentions	feedback_pos	feedback_neg	avmessagesize
$wissdr	4	14	1	0	65	24	16
** A - R	4	21	3	0	66	52	21
*Galaxy	2	9	3	0	56	115	19
*se-nse	2	9	1	0	0	58	21
---UFO-	8	5	5	2	2	54	131
-CALIB	4	3	4	2	45	78	25
-Chill	2	3	0	0	11	19	5
-Droidls	6	23	6	2	46	84	12

Figure 16: Opinion leadership attributes

Figure 16 shows an excerpt of the new data with the corresponding relative user scores representing our defined opinion leadership attributes. In order to identify opinion leaders, we made use of an unsupervised machine learning technique. K-means clustering was applied to first measure the distance between two users and then regroup them according to their individual attribute scores.

In the course of applying the k-means clustering approach, we processed the data with R running following commands in the R console.

```
# data import
opinionleader <- read.csv("data.csv", header=TRUE, row.names=1)
# run k-means
results <- kmeans(opinionleader, 6)
```

Firstly, the input data had to be loaded from a CSV file. Therefore, we defined the variable 'opinionleader' and loaded input values from 'data.csv'. The following step shows the actual application of the k-means algorithm. As stated in the literature review, k-means clustering demands the number of clusters to be determined by the analyst.

In order to choose the appropriate cluster solution, Peeples (2016) suggests to compare the number of the sum of squared error for various cluster solutions. As mentioned throughout the literature review, the sum of squared error is defined as the sum of squared distances between each item in the cluster and the cluster center. Thus, a suitable amount of selected clusters can be defined as the solution where the sum of squared error slows considerably. This can be shown graphically in R by applying following statements:

```
wss <- (nrow(opinionleader)-1)*sum(apply(opinionleader,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(opinionleader,center=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Cluster", ylab="Within groups sum of squares")
```

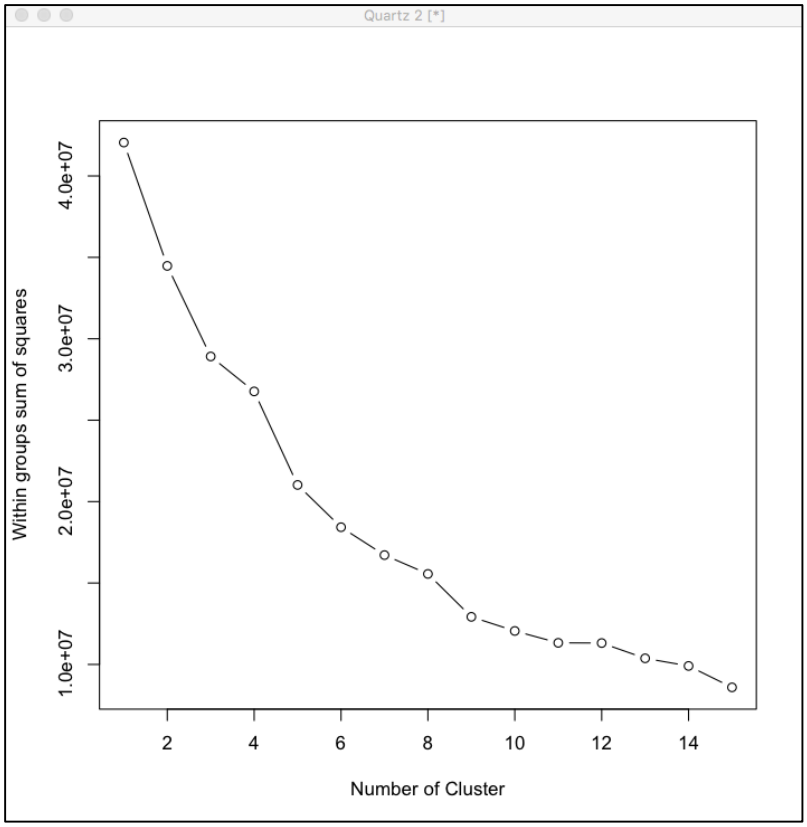


Figure 17: Cluster solutions for k-means clustering

The output is presented in figure 17 as plot of the within groups sum of squares and the corresponding number of clusters. We selected an amount of six groups as the appropriate number of clusters for our opinion leadership analysis as the reduction of sum of squared errors slows down with six clusters.

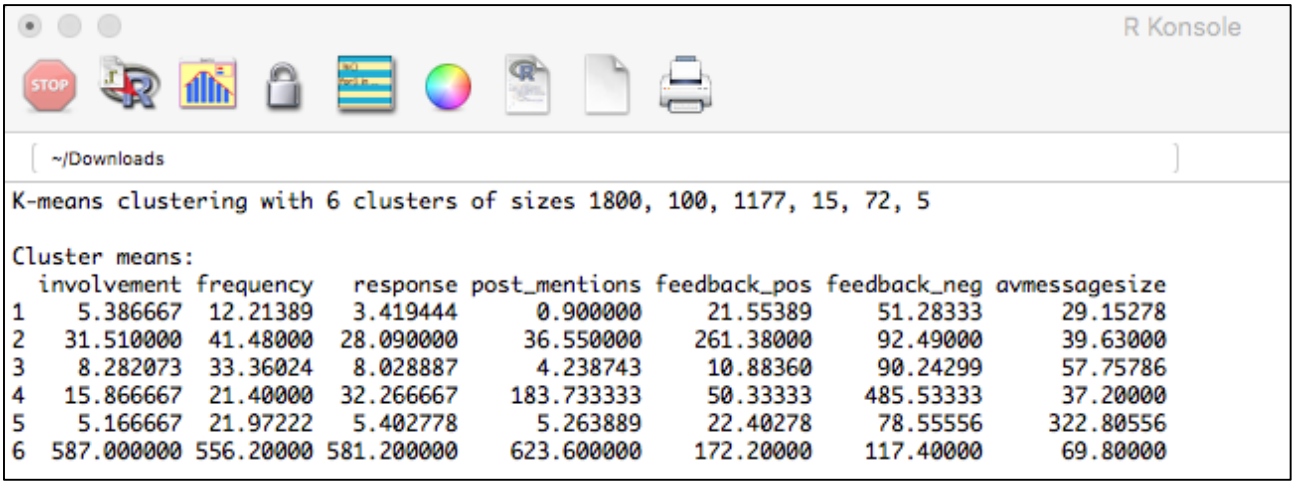


Figure 18: Results of k-means clustering for opinion leadership detection

Figure 18 shows the output of our k-means clustering approach with the pre-defined 6 cluster solution. Furthermore, the cluster sizes (users per cluster) as well as the cluster means of each attribute are portrayed.

We selected cluster number 6 (with 5 users) as our main opinion leadership cluster, showing the best results for almost every variable. As k-means clustering is not capable of identifying an ultimate solution, we selected the result that approached the ideal solution the best. Cluster number 1 and 3, containing 1800 and 1177 users, showed the lowest opinion leadership scores and thus were not of further interest for our analysis. Individuals in cluster 4 were mentioned a lot by other forum members but received a high degree of negative feedback. Thus, these users were not considered as opinion leaders either. Cluster 5 is described by a high average message size, but users within this group scored very low for all other attributes. We characterized this cluster as being spam, as average message size is long but the rate of responses and post mentions scores considerably low.

The 100 users in cluster 2, however, seemed to at least somehow fulfill the opinion leadership attributes. They received a significant amount of positive feedback and very low negative feedback, even though they were not (yet) highly involved in the forum discussions. We described them as our ‘rising star’ users that might show high opinion leadership potential. Thus, we decided not to rule them out and considered them as fulfilling the opinion leadership characteristic for lead userness.

We concluded our analysis with 105 users showing the potential of being opinion leaders in the xda-developers online community.

5.1.4 Dissatisfaction

This chapter deals with the 'dissatisfaction' aspect of lead userness. Lead users are generally described as being unsatisfied by existing products and services (von Hippel 1986).
Pajo et al. (2013) suggests identifying the dissatisfaction aspect with the use of sentiment analysis or the measurement of product related disposition. Related research indicates that lead users tend to express a greater amount of dissatisfaction with the offered products or services than other users (Lüthje and Herstatt 2004). This is why Pajo et al. (2013) measured the sentiment of online posts and used noticeable negative sentiment to separate lead users from other users (Pajo et al. 2013).

We adopted this approach and generated a polarity score for each post. However, we added a further condition. We assumed that dissatisfaction should be only measured for posts that specifically deal with Android topics. Therefore, we only measured the sentiment of the posts that mentioned the terms 'Standard Android', 'Stock Android' and 'Android'.
In short, we identified users with a bad attitude towards general and specific Android by applying sentiment analysis.

We conducted the sentiment analysis with Python. In comparison to Java, Python can handle the relevant libraries with better performance. Furthermore, the selection of available sentiment analysis tools is more mature and tested (Bird, Klein, and Loper 2009).
We evaluated several available sentiment analysis tools. Amongst others, we tested genism (a Python framework for topic modelling), textblob (a python library for processing textual data), the nltk sentiment package and pattern (web mining module for Python).

We decided to use the tool 'Pattern' for our sentiment analysis. Pattern was developed by the Computational Linguistics & Psycholinguistics Research Center of the Universiteit Antwerpen.
We selected Pattern, as it either can import a trained classifier or uses the corpus of SentiWordNet.

SentiWordNet is based on WordNet 3.0, a lexical database containing English words (Miller 1995). SentiWordNet assigned three sentiment scores to each synset in the WordNet 3.0 library: positivity, negativity, objectivity (Baccianella, Esuli, and Sebastiani 2010).

In addition to SentiWordnet, we trained our own classifier following an approach proposed by Pak and Paroubek (2016). They described the use of Twitter content as a training corpus for sentiment analysis.

We accessed the Twitter API and downloaded 500 Tweets that contained the tag '#Android' in combination with positive emoticons ':-)', ':)', '=)', ':D' and other positive tags like '#nice' and '#awesome'. After that we downloaded another 500 Tweets also with the tag '#Android', but now in combination with sad emoticons: ':-(', ':(', '=(', ';('. as well as the tag #fail. We aggregated and cleaned the results and created a training CSV file.

The following two statements exemplify the training set:

haha. Can dwl again mah. Got an awesome android app that allows me to control my quad using tablet eh. But iPad don't allow.	pos
What idiot had the idea of using ComicSans on Android?? (cf. SBB – #fail	neg

Our final classifier used SentiWordNet as well as our training set to conduct the actual sentiment analysis. In the first run, we tested the classifier with a pre-labeled set of data. We could achieve an accuracy score of 67%.

Run python script: sentiment analysis

Python can access MySQL databases as good as Java does. For connecting Python to the database, we used the mysql-connector-python-2.0.3 library.

```
cnx = mysql.connector.connect(user='root', password='CBS',
                              host='localhost',
                              database='masterthese')
msg = "connected!"
print msg

cursor = cnx.cursor()
```

Afterwards, the classifier was loaded.

```
# read training documents into a list of tuples

documents=[]
f = open("Android_trained.txt","r")
for document in f.readlines():
    parts= document.strip().split("\t")
    documents.append((parts[1],bool(int(parts[0]))))
```

The Python script initiated the sentiment analysis for each post and each quote separately. For each post the polarity and subjectivity could possibly range from -1,0 to +1,0. The results were stored into the database.

```
    #Do Sentiment Analysis for Post

    cursor.execute("""
                   SELECT post_post
                   FROM masterdata
                   WHERE IDx like %s
                   """,
                   (i, )
                   )
    row = cursor.fetchone()
    print(row)

    #sentiment analysis

    text = str(row)
    text = text.replace("bytearray(b'", "")
    print text
    senti_tupel = sentiment(text)
    print senti_tupel

    add_pol = senti_tupel[0]
    add_sub = senti_tupel[1]

    print add_pol
    print add_sub
```

We also measured the polarity of the separately stored quotes. The polarity of the quotes was used in the course of the opinion leadership identification.
In the last step, the values were inserted into the database table ‘masterdata’.

```
    cursor.execute ("""
                    UPDATE masterdata
                    SET post_pol=%s, post_sub=%s
                    WHERE IDx=%s
                    """,
                    (add_pol, add_sub, i))
```

For processing 100.000 entries, the sentiment analysis needed roughly 24 hours of calculation time.

The developed sentiment analysis approach identified 268 users showing a general bad attitude towards standard Android.

5.1.5 Ahead of trend

This chapter deals with the identification of the lead user characteristic 'ahead of trend'. The attribute implies that lead users face product needs considerably earlier than the majority of market participants (Schreier and Prügl 2008). We applied Schreier and Prügl's (2008) findings to our approach and assumed that these users would mention and discuss trending topics earlier than the majority of people.

The main process involved the identification of trending topics within a time frame of five years. After that, we identified the users who were the first to discuss these topics.

In order to identify these users, we applied the following six steps:

1) We used part-of-speech tagging in order to identify the main keywords of the posts
2) The gathered keywords were aggregated into 65 packages of monthly intervals (for each month starting from Jan 2011 until May 2016)
3) We counted the most used keywords per month
4) We calculated a monthly growth rate for the most used keywords
5) The keywords with the highest growth rate were obtained
6) We searched for the first five users that used these keywords in the beginning

Firstly, part-of-speech tagging was used to identify the noun phrases / keywords of each sentence. For calculating the term frequencies, we later used several text-mining methods.

As described in the literature review, part-of-speech tagging assigns ‘descriptors’ to each word of the sentence. Only nouns are relevant for the identification of noun phrases / keywords.

Initially, we focused on using the ‘Stanford Log-linear Part-Of-Speech Tagger’ which is based on a maximum entropy algorithm running Java (Toutanova and Manning 2000). However, the results of this tool were not matching our demands and high number of manual steps were involved.
We decided to use ‘textblob’ instead, a very simple and comprehensive tool that matched our requirements. TextBlob is a Python based natural language processing (NLP) toolkit that is based on NLTK and Pattern. As underlying corpora, it accesses Brown and WordNet.

Python Program phraser.py

This program followed the same logic as our other programs. First, it connected to the MySQL database and selected each post. Afterwards it processed the text and inserted the result into the database.
The text processing steps consisted of the following tasks: ‘sentence segmentation’, ‘tokenization’, ‘POS tagging’ and ’selecting the noun phrases from the POS tags’.

```
    ##### TextBlob API #####
    noun_phrases = set(TextBlob(text).noun_phrases)
    # Strip punctuation from ends of noun phrases and exclude long phrases
    result = [strip_punc(np) for np in noun_phrases if len(np.split()) <= 10]
```

The results were inserted into the field ‘post_topics’ of the ‘masterdata’ table. Following example shows the original text and the gathered phrases after processing.

Post content (text)	**Noun phrase (result)**
I don't know why, something about how linux packages zip files opposed to windows, and yes once I figure it out and find all the problems people can have I am going to make up a nice	[u'theme manager apps', u'themes', u'problems people', u'process',

walkthrough on making themes. But I wouldn't be surprised if there were already some theme manager apps in the works to make this whole process easier.	u'linux packages zip files', u'walkthrough']

For extracting the keywords of 100,000 posts, the application took roughly 24 hours.

Secondly, the extracted keywords were further processed in order to identify the trending words. For the second step, we created a new database table 'textmining' and ran the program:

D1_getTextminingTable.java

This program aggregated the keywords into 65 monthly intervals. By applying three 'for-loops', one for the total number of months, one for the number of years, and one for the months of the year, we generated 65 different SQL Statements used to aggregate the keywords per month.

```
for (int i = 1; i <= 59;){
    for (int j = 2011; j <=2016; j++){
        for (int k = 1; k <=12; k++){

            String sql1 = "SELECT GROUP_CONCAT(DISTINCT post_topics SEPARATOR ' ') FROM masterdata WHERE YEAR(post_timestamp) = "+j+"


            ResultSet rs1 = db.runSql(sql1);
            while (rs1.next()) {
                Name = rs1.getString(1);
                System.out.println("ID = " + i);
                System.out.println(Name);
                Name =  " '' "  + Name + " '' ";

                String sql0 = "SET group_concat_max_len = 18446744073709551615";
                String sql2     = "INSERT INTO masterthese2013.textmining (int_id, period, words) VALUES ('"+i+"','"+j+'-'+k+"', ?);"
```

Thirdly, we connected R to the MySQL database and loaded the keywords for each of the 65 months into python:

```
mydb = dbConnect(MySQL(), user='root', password='CBS', dbname='masterthese', host='127.0.0.1')
rs = dbSendQuery(mydb, "SELECT int_id, words FROM textmining;")
data = fetch(rs, n=-1)
corp <- Corpus(DataframeSource(data))
```

R regarded all words of one month as a 'document' / 'Docs' and the set of documents as one 'corpus'. We also removed specific words that were wrongly identified by the POS Tagger.

docs <- tm_map(docs, removeWords, c("example", "word"))
→ Words to be removed in separate file.

Afterwards, we created a document term matrix consisting of the 207 most frequent words. As shown in figure 19, all words are displayed on the x-axis. All months are displayed on the y-axis. The Cells contain the amount of occurrences of the respective word.

dtm <- DocumentTermMatrix(docs)

Docs	tuner	turbocharger	tutorial	tweak
1	0	0	0	0
2	0	0	1	0
3	1	1	2	1
4	2	0	0	1
5	0	0	1	6
6	0	0	6	1
7	0	6	0	7
8	3	7	3	16
9	0	9	1	6
10	1	16	7	6
11	4	18	1	3
12	1	4	2	8
13	2	2	1	4
14	0	5	2	4
15	2	13	1	6
16	0	0	0	0
17	0	0	0	0

```
> inspect(dtms)
<<DocumentTermMatrix (documents: 65, terms: 207)>>
Non-/sparse entries: 12387/1068
Sparsity           : 8%
Maximal term length: 13
Weighting          : term frequency (tf)
```

Figure 19: The 207 most frequent words

The term ‘turbocharger’ was first mentioned by a user in month 3, whereas it was mostly mentioned from month 7 to month 15.

Fourthly, we calculated the growth rates of each word. Therefore, we calculated the monthly growth rate per word. In the following step, we calculated the average growth over all 65 months.

Fithly, we inspected those words that showed a high average growth rate. We defined these terms as trend words since they had a low rate of mentions during first months and a strong increase of usage over time. Examples for words with a high average growth rate are: ‘tasker’, ‘miui’, ‘nandroid’, ‘storage’ and ‘busybox’.

We also inspected words with low growth rates. We found words that were continuously mentioned a lot. Therefore, we did not consider these words as trend words. Examples for words with a low growth rate are: ‘android’ ‘app’, ‘google’, ‘direction’ and ‘standard’.
In total, we gathered 42 trend words.

Sixthly, we created a SQL query for each trend word, which selected the five members that mentioned the keyword first.
Following SQL statement shows the query for the sample word ‘tasker’:

```
INSERT INTO masterthese2013.textmining_res (member_name, post_timestamp, post_topics, term)
SELECT member_name, post_timestamp, post_topics, 'tasker'  FROM `masterdata` WHERE `post_post`
LIKE '%tasker%' ORDER BY post_timestamp ASC LIMIT 4
```

The following figure displays the first four users that mentioned the words ‘tasker’, ‘miui’ and ‘nandroid’.

term	member_name	post_topics	post_timestamp
tasker	lucfig	[u'gaming settings', u'miui', u'locale execute plu...	2011-04-06 01:39:00
tasker	zeppeli	[u' /sys/module/lowmemorykiller/parameters/minfree...	2011-04-06 01:51:00
tasker	w4tcho	[u'scripts zeppelinrox', u'tasker', u'launcher suf...	2011-04-13 22:15:00
tasker	w4tcho	[u'new superscript thing', u'tasker', u'results co...	2011-04-14 13:30:00
miui	magiste	[u'ive', u'code', u'appropriate changes', u'build ...	2010-12-13 19:05:00
miui	evolutic	[u"u'hey guys im", u'miui rom']	2011-01-07 20:28:00
miui	Fallouti	[u'chalkwork iconset', u'set', u'cartoon icons pac...	2011-02-27 10:54:00
miui	Fallouti	[u'cute', u'android resources', u'peques', u'neon'...	2011-02-27 10:57:00
nandroid	TheDuc	[u'full nandroid backups', u'apps work']	2009-01-26 03:42:00
nandroid	hova ki	[u'pictures ringtones etc ... ', u'nandroid i', u'...	2009-01-30 04:06:00
nandroid	samysa	[u'nandroid ']	2009-01-30 04:12:00

Figure 20: Output ahead of trend analysis

To summarize, we were able to identify 74 users that posses the ‘ahead of trend’ characteristic.

5.1.6 Product-related knowledge and expertise

Lead users are characterized as possessing more knowledge and use experience than the majority of the market. Furthermore, it is found that use experience and product expertise is positively linked to innovativeness (Schreier and Prügl 2008).

We did not apply specific analytical methods for identifying users that possess the 'product-related knowledge and expertise' characteristic, as our findings indicated that knowledgeable users had already been detected and separated throughout the previous steps of our analysis. Following explanations describe the specific reasons for our decision.

When we started our lead user research among the members of the xda-community, we initially assumed that all users of the forum are characterized as possessing strong use experience and product expertise. Considering that the analyzed forum primarily deals with advanced topics in the field of Android, Android features and Android customization, we did not expect to identify users that possess only weak knowledge in the area of interest.

However, after having downloaded our sample data and preprocessed the gathered information, we realized that a considerable amount of users was only passively involved in the community, i.e. have not submitted any posts. The question raised, whether these individuals could really be characterized as fulfilling the knowledge and product expertise criteria. Due to this fact, we assumed that knowledgeable community members must at least be actively involved in the regular community discussions. Consequently, we narrowed down the possible group of people to those fulfilling the 'involvement' characteristic of lead users.
This decision is in line with literature stating that experts in online forums are highly involved in the ongoing discussions and frequently post new content (Munger and Zhao 2014).

Besides being highly involved, Lyons and Henderson (2005) state that expertise and product knowledge is strongly linked to opinion leadership. Thus, we argue that our identified group of opinion leaders also possess an important attribute that determines their product expertise and knowledge.

In linking both perspectives, we consequently characterized knowledgeable users and product experts as those who fulfilled the ‘involvement’ as well as the ‘opinion leadership’ characteristic. By following this approach, we could identify 105 community members showing the ‘product-related knowledge and expertise’ attribute.

5.1.7 Results and identified lead users

Following Stieglitz et al’s (2014) framework, we applied several analysis methods to identify lead users among the xda-community members.

Firstly, we applied different filtering methods with the aim at uncovering the ‘involvement’ characteristic of lead userness. Secondly, we aggregated seven different attributes and conducted k-means clustering in order to identify opinion leadership among the forum members. Furthermore, we discovered unsatisfied users by applying sentiment analysis. Afterwards, we conducted part-of-speech tagging and further text mining to detect users that were ahead of trend. Finally, we examined product related knowledge and use experience among the community members and concluded that there was a high correlation with opinion leadership and user involvement.

We analyzed every lead user characteristic independently, except the involvement attribute. In other words, the applied analysis approaches were rather complementary than subtractive. Following figure illustrates the application of our methods and the resulting lead user population. The identified lead users are displayed as the intersection of each analysis result.

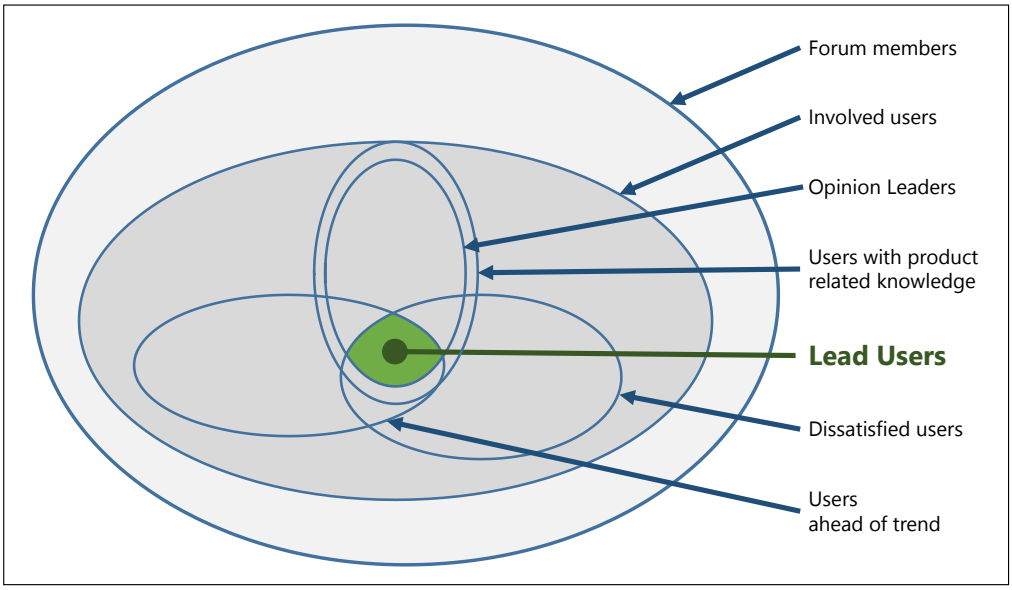
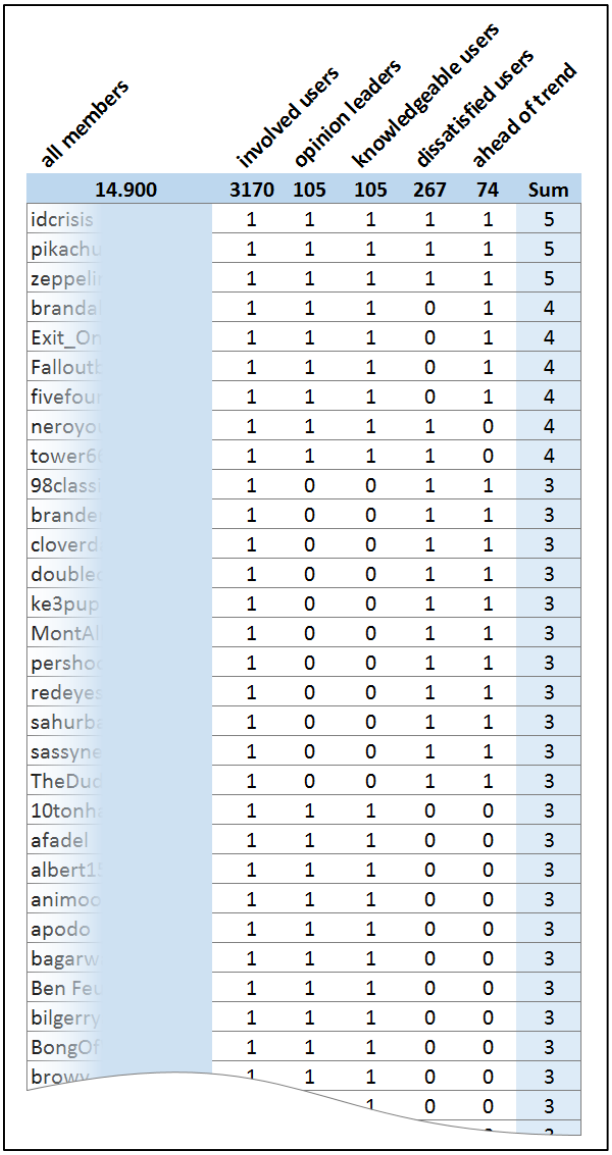


Figure 21: Identified lead user population

The initial dataset contained 14,900 users of the xda-developers community. These individuals are represented by the outmost ellipse. As we defined involvement as a prerequisite for all further analysis steps, we filtered out those users that did not match the 'involvement' characteristic. Thus, 3,170 users were left for further analysis. This new set of users is represented by the second biggest ellipse.

The smaller ellipses show the results of the 'opinion leadership', 'dissatisfaction', 'ahead of trend' and 'product-related knowledge and expertise' analysis.
We identified 105 community members that can be described as opinion leaders. 268 users fulfill the dissatisfaction criteria. Another 74 members of the xda-community forum are ahead of trend and 105 users that possess product-related knowledge and expertise.

By combining the results, we identified three members of the xda-developers community as lead users. These individuals can be described as possessing all five distinctive lead user characteristics.



all members	involved users	opinion leaders	knowledgeable users	dissatisfied users	ahead of trend	
14.900	3170	105	105	267	74	Sum
idcrisis	1	1	1	1	1	5
pikachu	1	1	1	1	1	5
zeppeli	1	1	1	1	1	5
branda	1	1	1	0	1	4
Exit_Or	1	1	1	0	1	4
Fallout	1	1	1	0	1	4
fivefou	1	1	1	0	1	4
neroyo	1	1	1	1	0	4
tower6	1	1	1	1	0	4
98class	1	0	0	1	1	3
brande	1	0	0	1	1	3
cloverd	1	0	0	1	1	3
double	1	0	0	1	1	3
ke3pup	1	0	0	1	1	3
MontAl	1	0	0	1	1	3
pershc	1	0	0	1	1	3
redeye	1	0	0	1	1	3
sahurb	1	0	0	1	1	3
sassyne	1	0	0	1	1	3
TheDuc	1	0	0	1	1	3
10tonh	1	1	1	0	0	3
afadel	1	1	1	0	0	3
albert1	1	1	1	0	0	3
animoo	1	1	1	0	0	3
apodo	1	1	1	0	0	3
bagarw	1	1	1	0	0	3
Ben Fe	1	1	1	0	0	3
bilgerry	1	1	1	0	0	3
BongOf	1	1	1	0	0	3
browy	1	1	1	0	0	3
			1	0	0	3

Fulfilling all 5 lead user characteristics	3
Fulfilling 4 lead user characteristics	6
Fulfilling 3 lead user characteristics	107
Fulfilling 2 lead user characteristics	307
Fulfilling only 1 lead user characteristic	2747

Figure 22: Identified lead users

5.2 Verification of results

Our empirical findings suggest that it is possible to utilize machine learning algorithms and other forms of computer-aided methods to identify each specific lead user characteristic amongst the online community members. We identified and defined those users as fulfilling the lead userness criteria that showed high scores in the analysis we conducted. Only when a user clearly shows high involvement, acts as an opinion leader, is dissatisfied with the current product offerings, shows that he is ahead of trend and possesses product-related knowledge and experience, we

concluded that this individual can be characterized as a lead user in the Android development community.

The goal of the following chapters is to find evidence that our approach is suitable for detecting lead user characteristics in online forums and that the so identified individuals can indeed be described as lead users. In the course of our analysis we made use of different supervised and unsupervised machine learning techniques as well as other computer-aided methods. As these procedures have already been very well documented and verified in its own particular field of interest, e.g. sentiment analysis for exploring dissatisfaction (Pang, Lee, and Vaithyanathan 2002; Pak and Paroubek 2016; Go, Bhayani, and Huang 2009) or k-means clustering for finding opinion leadership (Hudli, Hudli, and Hudli 2012) we do not see the need for further verification of the used mechanisms. Nevertheless, we argue that all these computer-aided techniques put together in a newly developed approach will help researches to identify lead user characteristics and ultimately lead users in online communities. This novel combination of methods definitely needs further verification, as it has never been discussed in the literature before. Due to the uniqueness of our approach and the required proof of concept, it is necessary to select a suitable method to verify the results and findings.

So far, most of the research conducted in the field of lead user identification focuses mainly on the widely used and discussed traditional methods of mass screening, pyramiding and broadcast search. In the recent years, new streams of literature have been evolving seeking to establish new mechanisms to identify lead user behavior. As presented in the literature review, Belz and Baumbach (2010) introduced netnography to lead user research and Pajo et al. (2013, 2015) were the first to apply machine learning techniques in the research area. Even though our own analysis is partly motivated by Pajo et al.'s FLUID approach and also utilized parts of their concept, we cannot rely completely on their findings and method verification. This is due to the fact that our research domain as well as the applied machine learning techniques differ significantly.

Our method of verification is motivated by Belz and Baumbach (2010). They introduced netnography to the field of lead user research and needed to proof the validity of their approach, as netnography has never been used before to identify lead user behavior in online communities. The researchers found verification by applying the traditional technique of mass screening to

their field of study. In order to validate their empirical results, they sent out emails containing a screening questionnaire to the most active members of the studied online community. They concluded that the well-researched and documented method of mass screening indeed helped them to confirm their explorative netnography lead user study.

In the following, we describe a mass screening approach used to verify the results of our own analysis. We show the particular characteristics and method-specific advantages, how we applied the technique as well as discuss the results of the selected approach. Finally, the comparison between our computer-aided technique and the results of the mass screening questionnaire show how the newly developed techniques are suitable for the detection of lead user characteristics and behavior.

5.2.1 Mass screening as method for result verification

As primary method for verifying our analysis and to show that machine learning techniques are suitable for lead user detection, we applied a mass screening search. As described in the literature review, mass screening is one of the most classic methods of lead user detection and it has been successfully applied in a great number of empirical studies (von Hippel, Franke, and Prügl 2009). This rather quantitative approach is based on parallel scanning for lead user characteristics amongst a large entity of product or service users. In our case, this user population is represented by members of the xda-developers community.

We followed a classic mass screening research style by conducting surveys in form of written questionnaires that were sent out to selected community members. Participants were invited to answer questions about their own innovation activities and general behavior in the community. This is in line with studies done by Urban and Von Hippel (1988), Lüthje and Herstatt (2004) or Stockstrom et al. (2016), who clearly show the successful application of mass screening for lead user detection. Furthermore, we selected mass screening to gain from its method-specific characteristics and advantages. As we aimed at conducting the study in a pre-defined user population with clear boundaries, we did not have to make use of the cross-domain search benefits provided by pyramiding or broadcast search. Lüthje and Herstatt (2004) described mass screening as a very effective lead user research method, if the search field is well-defined by clear boundaries and contains a manageable number of potential users. Our mass screening

method was conducted to measure the five lead user characteristics: ahead of trend, dissatisfaction, product-related knowledge and experience, involvement and opinion leadership. Therefore, we defined three questions to measure each lead user characteristic through an online questionnaire. The detailed setup of the screening survey will be described in the next chapter.

In order to compare the results of the mass screening search with the empirical findings of our own analysis, we made use of 'triangulation by method', also known as methodological or multimethod triangulation (Meijer, Verloop, and Beijaard 2002). Triangulation by method can be defined as "gathering information pertaining to the same phenomenon through more than one method, primarily in order to determine if there is a convergence and hence, increased validity in research findings" (Kopinak 1999, 171). In other words, triangulation helps researchers to find validity in their results by applying a second method of study. In accordance to this definition, we applied the mass screening search technique to enhance the validity of our own empirical findings.

The following chapter shows how we set up the research survey, which we later sent out to the xda-developers community.

5.2.2 Defining the screening questionnaire

In order to verify our empirical findings by applying the mass screening technique, we defined a questionnaire designed to cover all five lead user characteristics. Following the research by Belz and Baumbach (2010), we selected three statements that clearly describe each of the lead user characteristics. The participating users were invited to rate these items on a 5-point liker-scale, depending on whether these statements explain their own opinions, feelings and actions or not.

As described in the literature review, lead users are characterized as being ahead of what later becomes general market trends (von Hippel 1986). Thus, they tend to either innovate themselves or show a high rate of adopting new products (Schreier and Prügl 2008). In order to identify the 'ahead of trend' characteristic we asked the community users about their product adoption and innovation behavior. We selected following questions: 'I am one of the first within my circle of friends or the online community who adopts new Android developments', 'I love adopting new

Android software and features before the majority of people to' and 'I am interested in developing new features and apps on my own'.

Secondly, we let user rate statements asking about their satisfaction with the current offerings in the market. A high degree of dissatisfaction fosters innovation activities done by lead users (von Hippel 1986). We let users rate following items: 'I am dissatisfied with the Android features or apps that are currently in the market', 'At the moment my expectations regarding Android software or features is not fulfilled' and 'I have requirements concerning features which are not satisfied by now'.

Schreier and Prügl (2008) demonstrate that lead users tend to possess more product-related knowledge and use experience than the majority of individuals. Thus, we let community members rate their own perceived knowledge and use experience on following statements: 'Within my circle of friends and the online community I am considered as an Android expert', I know a lot about Android feature and app development' and 'I regularly work on own apps and features for Android'.

As discussed in the literature review, we based our definition of 'involvement' on research conducted in the field of social media analysis. In this context, involvement can be defined by time spent and intensity shown in online communities (Stone 1984). Thus, we asked users about their time spent online and content they created in the community: 'I spend a lot of time in online communities dealing with Android', 'I regularly post new comments in the online community' and 'It is a lot of fun informing myself about new features and apps'.

Lastly, as lead user are characterized as opinion leaders (Morrison, Roberts, and Midgley 2004; Urban and Von Hippel 1988), we let user rate statements regarding their personal influence on other community members. These items were: 'In discussion about Android I tell others more than they tell me', 'I regularly get positive feedback on the threads I open or the comments I make' and 'Other users often refer to or quote my posts'.

We designed the questionnaire in a way that the participating community members can rate each item on a likert-type scale ranging from 1 (=strongly disagree) to 5 (=strongly agree), with each lead user characteristic being represented by three statements.

Scale Item	Item Wording
Ahead of Trend	I am one of the first within my circle of friends or the online community who adopts new Android developments I love adopting new Android software and features before the majority of people do I am interested in developing new features and apps on my own
Dissatisfaction	I am dissatisfied with the Android features or apps that are currently in the market At the moment my expectations regarding Android software or features is not fulfilled. I have requirements concerning features which are not satisfied by now
Product-related knowledge and experience	Within my circle of friends and the online community I am considered as an Android expert I know a lot about Android feature and app development I regularly work on own apps and features for Android
Involvement	I spend a lot of time in online communities dealing with Android I regularly post new comments in the online community It is a lot of fun informing myself about new features and apps
Opinion Leadership	In discussions about Android I tell others more than they tell me I regularly get positive feedback on the threads I open or the comments I make Other users often refer to or quote my posts

The table shows the questionnaire we created for our mass screening approach. The following explanations will show how we set up and ran the survey as well as how the results were measured.

5.2.3 Conducting the survey

We ran the study with the use of an online survey containing the specified 15 lead user statements. Every item of the questionnaire could be rated on a likert-style scale (ranging from 1 to 5) by selecting the appropriate score that represented the participant's own attitude towards

the statement. In order to execute the survey we used surveyXact[2], an online tool that allows generating questionnaires for a wide range of platforms, e.g. emails, desktop computers, smartphones and tablets. Furthermore, surveyXact includes a functionality to download the results as an CSV file, which allowed us to use the data for further analysis. Figure 23 portrays a screenshot from surveXact with the first statement to rate.

Figure 23: First item on the likert-style scale

In order to stay inside the pre-defined search domain and not crossing domain specific boundaries, the questionnaire was only sent to those users that we extracted throughout the main analysis. Furthermore, we only selected users that showed frequent activity in the online community, due to limits regarding the direct messaging functionality of the xda-developers forum. This procedure is in line with Belz and Baumbach (2010), who sent their questionnaire only to the most active members of the studied community. In total, we sent the survey to 1000 users of the xda-developers community.

Every user was contacted with a personalized message and a unique link to the survey. The link also contained the username of the respondent, which was automatically stored in a background table of the surveyXact tool. This ensured that we could match every completed questionnaire with the corresponding user profile and our depending analysis. By clicking on the link, the participant was directed to questionnaire and could start rating the lead user statements.

[2] http://www.survey-xact.dk

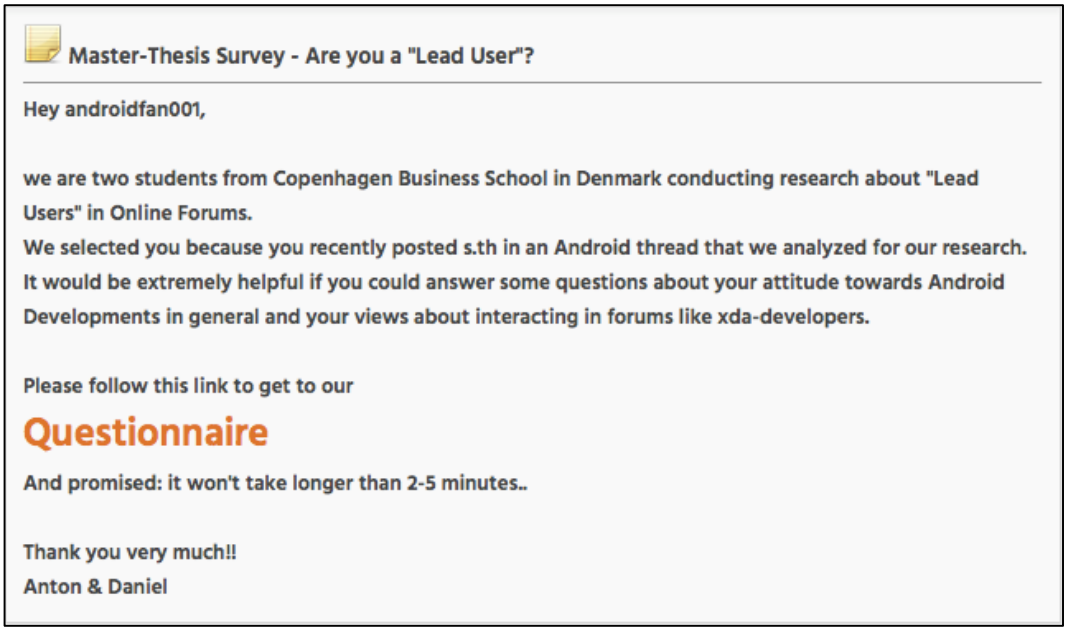


Figure 24: Invitation to the lead user study

Figure 24 portrays the individual message we sent out to every selected member of the xda-developers community.

On the first day, we sent out 250 requests and faced an extremely high response rate. Within the next five hours, we already had 20 completed questionnaires. From some users we also received direct Feedback:

*Please, **** off. Would have said that to a lot of users on the xda developers forum cause of the humongous stupidity going on there.... So, there is your answer*	*I have done it for you. I myself am a more android fan over developer, but I like to keep my stuff up to date so I follow these, and thus I can help others as well trying to figure things out.*	*It was a pleasure guys, I like I have helped*

In total, we ran the survey for 10 days and sent out 1000 questionnaires. Overall, 303 users filled out the survey, showing a response rate of 30 per cent. We downloaded the results and measured the responses, which will be described in the following chapter.

5.2.4 Measurement of results

As mentioned before, we made use of multiple statements in our questionnaire to measure each of the six lead user characteristics. In total, we had 15 questions the participants rated on a scale ranging from 1 (=strongly disagree) to 5 (=strongly agree). We totaled the items without weighting them, resulting in the lead user score. Following this approach, theoretically each participant's lead user score is ranging from 15 to 75, 15 with the lowest and 75 with the highest result.

Before analyzing the survey results, we measured the internal validity of our survey by calculating Cronbach's alpha using R. Therefore, we imported the survey answers into R, loaded the library package 'psych' and applied it to calculate Cronbach's alpha.

```
library(psych)    #make the psych package active
my.data <- read.csv("survey.csv", header=TRUE, row.names=1)  #read in the data
alpha(my.data)   #find alpha
```

We got 0.88 as Cronbach's alpha for the new variable, which shows a high degree of internal validity of our liker-scale survey (Gliem & Gliem 2003).

	Ahead of Trend			Dissatisfaction			Product related knowledge		
xda-name	I am one of the first within my circle of friends or the online community who adopts new Android developments	I love adopting new Android software and features before the majority of people do	I am interested in developing new features and apps on my own	I am dissatisfied with the Android features or apps that are currently in the market	At the moment my expectations regarding Android software or features is not fulfilled.	I have requirements concerning features which are not satisfied by now	Within my circle of friends and the online community I am considered as an Android expert	I know a lot about Android feature and app development	I regularly work on own apps and features for Android
Dark_Eye	5	5	4	2	2	4	4	4	4
78Staff	5	5	4	5	4	2	4	4	4
da__bea	4	5	4	5	5	4	3	4	4
XDAMaxe	5	5	5	5	1	4	4	4	4
max-555	2	3	4	4	4	3	4	4	4
malbert1	3	4	4	5	4	3	4	4	5
fedevd	3	4	4	3	3	4	5	5	5
aceqott	2	3	5	3	3	4	5	5	5
Germain2	3	4	4	4	4	2	4	4	4
Crytech	3	4	4	4	4	2	4	4	5
denzel09	5	5	4	4	3	4	4	5	4
galaxys3	4	5	4	5	1	4	5	4	4
.:Crack:.	5	4	3	4	4	4	3	4	4
Maybelle	5	5	5	4	2	4	4	4	4
King Roll	5	5	2	5	2	5	5	4	1
suprano	4	5	4	5	2	4	4	4	2

Figure 25: Mass screening questionnaire results (extract)

The lead user scores of our questionnaire ranged from 22 to 70 with a median of 46. In accordance to the mass screening research done by Urban and Von Hippel (1988), we conducted a cluster analysis in order to identify a subgroup that scores highest on all lead user attributes. Therefore, we calculated the average score for each characteristic and applied the k-means algorithm to our data sample using R.

The comparison of the number of the sum of squared error showed eight clusters as the appropriate solution.

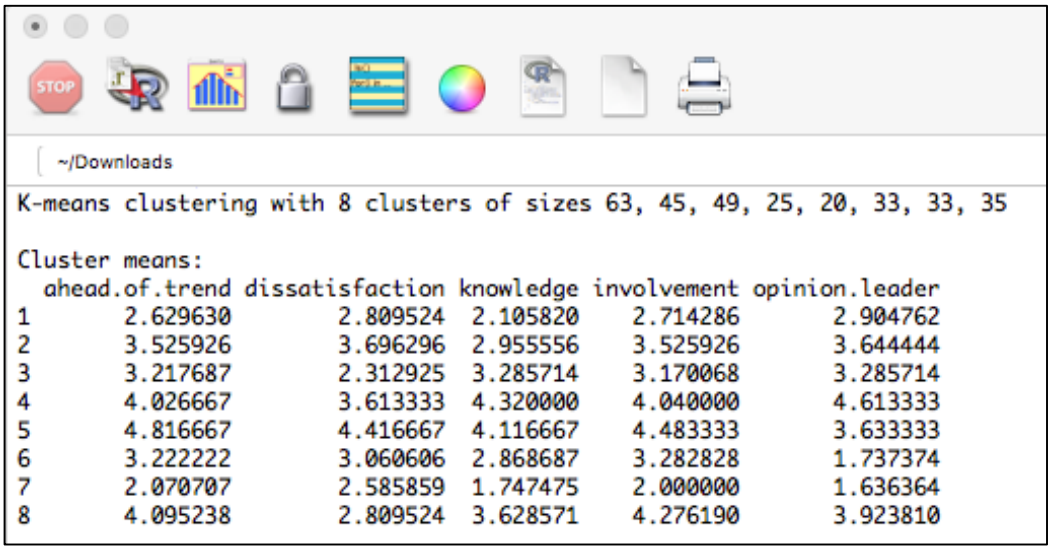


Figure 26: Survey clustering results

Figure 26 shows the results of the applied k-means clustering. We selected cluster 5 as our resulting lead user subgroup, as the sum of the means of each variable scored the highest. Even though cluster 4 was also characterized with high scores for all lead user attributes, we decided to rule it out. This is due to the fact, that cluster 4 scored considerably lower for the ahead of trend and dissatisfaction characteristic compared to cluster 5.

While literature describes all five characteristics as being important for identifying lead userness, the ahead of trend and dissatisfaction attributes are regarded as the strongest in the search for lead user behavior (Schreier and Prügl 2008).

In total, we identified 20 individuals through our mass screening study and regarded them as possessing strong lead user characteristics.

After conducting and analyzing the mass screening questionnaire we made use of 'triangulation by method' in order to compare the findings to the results of our computer-aided lead user identification approach.

5.2.5 Comparison of results

By comparing the results of the conducted mass screening with the findings of our computer-aided lead user research, we aimed at attaining confirmation for our approach. We made use of 'triangulation by method' to enhance the validity of our research and findings. As described before, the idea behind triangulation by method is to apply more than one technique to the field of study in order to determine if there is a convergence in results.

Through our computer-aided lead user search technique, we identified three individuals that possess all analyzed lead user characteristics. These individuals were defined as entirely fulfilling the criteria for lead userness.

Afterwards we applied the traditional lead user search technique of mass screening to the same user population by sending out questionnaires. In total, 303 community members filed out the online survey. The results of the mass screening research approach concluded that there exist 20 community members that possess strong lead user characteristics.

The results of both methods were then triangulated to enhance validity of our computer-aided technique.

Screening / Own approach	Lead users	Non-lead users
Lead users	2	1
Non-lead users	18	282

The table highlights the application of ‘triangulation by method’. It shows that two community members were identified as lead users by both, the mass screening and our computer-aided technique. According to Belz and Baumbach (2010) it is safe to assume that these individuals clearly show strong lead user characteristics, as the result is based on self-assessment by the survey participant (screening) and the external assessment by the researchers (our computer-aided method).
For one individual the identified lead userness could not be verified by the mass screening approach.
Furthermore, 282 users were clearly described as not fulfilling the lead userness as both research methods characterized them as non-lead-users.
The screening questionnaire considered 18 more users as possessing lead user characteristics. These findings were not identical with our research and we were not able to identify these individuals as showing strong lead user behavior.

In total, for 284 users the validity of our approach could be confirmed by conducting the mass screening study. In the case of 19 community members, our results did not match with the verification study. Nevertheless, as user status could be correctly confirmed for a majority of community members, we can argue for a high validity of our computer-aided research technique.

6 Conclusion

Our analysis demonstrated how to identify lead user characteristics in online forums using different computer-aided methods. We could also enhance validity for our technique by conducting a mass screening lead user search in our selected user population. However, even though the findings indicate the successful application of the developed tools, we saw some discrepancies in results when comparing it to the mass screening search. This makes it essential to further discuss our approach and findings as well as to bring other perspectives to our research.

Furthermore, this thesis aims at providing new ideas to the field of study and wants to inspire further academic research in the area of lead user identification.

Therefore, the final chapters revisit and discuss developed tools and findings of this thesis. Furthermore, limitations of research and suggestions for potential future studies is provided. Lastly, we show practical implications for management.

6.1 Discussion

This chapter critically reviews some of the major steps and argumentations of our analysis. We discuss selected analytical methods in order to bring new perspective to our research and to critically deal with our findings.

6.1.1 Computer-aided methods for lead user identification

In order to identify lead user characteristics and lead userness in online forums, we applied several machine learning methods as well as other computer-aided techniques. Following explanations critically review two of the conducted procedures and bring other perspectives to our research.

6.1.1.1 Opinion Leadership

For analyzing the opinion leadership characteristic, we followed a k-means clustering method that relied on individual user profiles of the xda-developers community members. We gathered data on the user's own engagement and how the community reacted to his or her posts. Based on this information we could build subgroups with similar patterns and identify those individuals that scored highest on the selected attributes. However, even though our approach could find

verification in the research and findings of Hudli, Hudli, and Hudli (2012), the question remains, how other academics deal with the topic of opinion leadership in online forums.

Our chosen method was primarily based on the analysis of user content, the members' behavior and the reactions of others. We analyzed how intensive the members interacted with the community, how frequently the users posted new messages and whether the reaction of other individuals was positive or negative.
Other academics suggest that network based approaches to opinion leadership identification might outperform content analysis methods in terms of accuracy. These techniques examine the relationship patterns of users within a social network and identify an individual that is located in the center of the user network (Ning et al. 2012).
However, these methods are regularly criticized as not taking attitudes and feelings expressed in comments into account. Sentiment in text is often seen as equal important for the detection of opinion leaders (Wu et al. 2015).

For our own analysis, we put major focus on the content and the sentiment aspect of opinion leadership detection. We defined opinion leaders as users receiving substantial positive comments and a low degree of negative feedback. Even though our approach was not mainly based on the analysis of social networks within the groups of forum members, we did not completely rule out this perspective. By measuring the degree to which a user was referred to in messages of others and the number of direct responses, allowed us to draw conclusion on a user's centrality amongst the online population.

Consequently, one could argue that our study was only focusing on content specifics. On the other hand, one can see our approach as a compromise in trying to merge both research themes.

6.1.1.2 Ahead of trend

We determined the 'ahead of trend' characteristic by identifying community members that were among the first to mention trending keywords in the online discussions.
The idea was to first calculate the average growth rate of major keywords, followed by the identification of those keywords that showed the highest average growth throughout the last years. Finally, we identified community members that already used these terms in an early stage.

To our knowledge, we were the first to conduct this combination of analysis for identifying the 'ahead of trend' characteristic. This means that we do not have scientific validation for our approach. Yet, when reviewing literature and techniques in the field of trend detection, we recognized that keyword growth has been used before as an indicator for evolving trends (Sorrells 2016; McNeill 2015).

One major drawback of this approach was that we were dealing with historical data. Since we analyzed historical data over a period of five years, we identified some users that were 'ahead of trend' several years ago. For this reason, it is questionable whether these users are still 'ahead of trend' today.

We realized that trend discovery would perform better using social media with a higher rate of new posts (e.g. Twitter or Facebook). Thereby, significant differences of word occurrences can be measured in a relatively short timeframe. In xda-developers, we could only measure trending keywords that evolved over months or even years.

We also evaluated and tested the unsupervised machine learning method 'Term Frequency Inverse Document Frequency' (TF-IDF) as an alternative for our trend analysis. TF-IDF scores 'unique' words of a set of documents higher than words that are commonly used in a corpus. However, we concluded that knowing such rare words does not imply that these terms are trend keywords. This is because TF-IDF is not able to measure the increase in usage. This means that TF-IDF cannot measure differences of word occurrences over time. For that reason, we did not use this analysis method for our trend analysis.

6.1.2 Mass screening for result verification

In order to enhance the validity of our findings we sent out a mass screening questionnaire to selected members of the xda-community and triangulated to results with our computer-aided analysis. We were able to show that for 284 users our results were confirmed by the screening study. Although the outcomes show high validity for our computer-aided approach, the question could be raised whether other traditional techniques of lead user identification, e.g. pyramiding, would have been more appropriate for our field of research.

In the case of this thesis, mass screening was applied to scan for lead user characteristics amongst the xda-developers members. The search field contained a manageable number of individuals with the potential to reach the entire user population. Due to this fact, we saw it as the most suitable technique to confirm our own findings.

However, in the course of its application we also struggled with the distinctive disadvantages of the approach. Even though it was theoretically possible to fully reach the selected user population, yet only one-third of the addressed members completed the survey. This absence of user data possibly had a considerable impact on the outcomes of our result verification, as we were not able to receive the screening responses of all users. Furthermore, the success of mass screening is highly dependent on the valid self-assessment of the participants. When manually reviewing the survey, we noticed that even users that did not post regularly in the community rated themselves as being highly involved in the ongoing discussions. This potential overrating of user attributes might interfere with the correct identification of lead user characteristics.

The execution of a pyramiding search approach, however, might have overcome these method-specific disadvantages. By applying pyramiding search, we would have started to ask community members to name other persons that, in their opinion, know more about Android development or possess better information about other experts in the product area. In general, this process continues until the very top of the search pyramid is reached and the potential lead users are identified. In fact, this search approach might have reduced time and increased efficiency of our verification study, as we would not have been dependent on many members answering the questionnaire. Furthermore, the risk of invalid self-assessments could have been reduced. On the other hand, pyramiding search potentially would have leaded us to users that were located outside of our research domain and thus, were not covered by our computer-aided technique. In this case, we possibly would not have been able to enhance validity of our study at all.

Taking both perspectives into account, mass screening was still considered as the most appropriate verification technique in the specific research setting. It allowed us to stay in the pre-defined domain and we were not confronted with potential lead users outside the xda-developers community.

6.2 Limitations and further research

Our research is limited by several factors. This chapter reviews important shortcomings, conditions and assumptions that potentially impacted the research design, the application of our research techniques as well as the interpretation and verification of our results. Furthermore, we give recommendations for further research.

Firstly, our analysis is limited by the chosen research context. We purely focused on the area of Android development as we assumed that individuals in this domain possess high affinity towards social media and own development of new tools and features. Furthermore, we narrowed our research down to one particular online discussion forum. Due to these circumstances, it cannot be argued for general validity of our research approach, i.e. that the developed tools can be applied to other domains such as fashion or sports.

Future research should aim at answering the question whether and how our approach of identifying lead users can be applied to other online forums dealing with various different topics, like sports, fashion or food. Furthermore, as we particularly narrowed down our research to the field of online forums, the question arises how our developed tools and methods would perform in the context of other social media platforms, like Facebook, Twitter, or LinkedIn.

Secondly, we applied our research techniques only to a sample dataset of 100.000 posts crawled from the xda-developers community. The selection of the appropriate topics and threads was conducted manually and thus, influenced by our assumptions regarding the most relevant subcategories in the field of general Android software development, Android software and hacking and Android themes. Some innovative users might be more active in other areas of the xda-developers community and thus, were not regarded as lead users by the results of our analysis.

The opportunity exists to tap into the field of big data analysis by not limiting the dataset, but analyzing millions of samples instead. Following this approach, researchers would not need to rely on self-assessment of the appropriate topics and threads in the online forum. Big data analysis would allow examining the entire community with all its different topics and subcategories. Furthermore, even though our developed tools and approaches are theoretically

compatible with big data analysis, it would be interesting to see the actual application of the methods in new circumstances.

Furthermore, the developed and utilized computer-aided techniques are highly dependent on the specific characteristics of the underlying data analysis methods. We mainly used machine learning algorithms for identifying lead user attributes among the selected user population. Supervised and unsupervised learning itself is dependent on several important factors.
First, sentiment analysis and part of speech tagging are highly influenced by quality and quantity of the chosen training corpora. Thus, we are not able to argue that the outcome of both procedures is not limited by the selected training set.
Second, as stated in the literature review, there is no guarantee that unsupervised machine learning algorithms do obtain the globally optimal solution. In other words, the results of our k-mean clustering approach used for detecting opinion leaders might be different when re-running the analysis. Thus, we cannot argue that the most appropriate solution for opinion leadership is identified using clustering algorithms.

In reviewing the theoretical basis for our analysis, we showed that different techniques might be suitable for applying to our data analysis. Future research might further investigate different tools and approaches for computer-aided detection of lead user characteristics. One field of interest is the comparison between the different machine learning algorithms in terms of internal quality and results, e.g. different training corpora for sentiment analysis might be compared.

For enhancing the validity of our analysis, we used a mass screening questionnaire and triangulated the results. The ability to verify our empirical results was limited by the amount of answers we received from the community members. Unfortunately, we were not able to screen the entire user population due to the relatively low response rate of the users.
Furthermore, we only sent the survey to a sample of 1000 users due to limits regarding the direct messaging functionality of the xda-developers forum.
Lastly, the applied mass screening search was cross-sectional in nature, as it represented a ‘snap-shot’ of the status quo at that time. This means, that we might have disregarded community members who had not been active for a certain period of time. Our empirical analysis was based on historical data from the year 2011 on. Thus, we might also have identified lead users that did

not actively engage in the community during the time when we conducted the mass screening search.

Future research might aim at comparing different methods for result verification in the field of lead user study. Furthermore, it would be interesting to see whether our results would look different if applying the method for a longer period.

6.3 Implications for management

This thesis introduced a novel approach for led user detection in the highly dynamic online environment. The described techniques should not only motivate academics to tap into novel fields of lead user research, but also inspire corporate organizations to adopt the newly generated insights and use them for product development processes.

Our approach of lead user detection has the great potential to considerably speed up corporate innovation activities. Businesses, so far, mainly rely on the traditional methods of mass screening, pyramiding and broadcast search when watching out for innovative user behavior. These techniques are often criticized as being cost intensive, slow and require substantial manual work. Machine learning and other computer-aided approaches will help organizations to speed up the detection of lead users in the marketplace. Consequently, in faster identifying lead user innovations and thus, evolving market trends, producers get the chance to significantly speed up time-to-market of new products and services.

Furthermore, we described that data gathered from social media platforms is highly relevant and valuable for research and development departments. Even though, companies have already recognized the value of user content in social media, they mainly use this information for marketing and communication purposes. Our work should inspire management to look at this data from a completely different perspective. Innovative users are actively engaging in online forums and on other social media platforms. Firms need to invest in sophisticated IT infrastructure in order to successfully access, store, analyze and report this data and to get valuable insights.

References

Amadeo, Ron. 2013. “Google’s Iron Grip on Android: Controlling Open Source by Any Means Necessary | Ars Technica.” October 21. http://arstechnica.com/gadgets/2013/10/googles-iron-grip-on-android-controlling-open-source-by-any-means-necessary/1/.

Amaro, Suzanne, and Paulo Duarte. 2015. “Travel Social Media Involvement: A Proposed Measure.” In *Information and Communication Technologies in Tourism 2015: Proceedings of the International Conference in Lugano, Switzerland, February 3 - 6, 2015*, edited by Iis Tussyadiah and Alessandro Inversini, 213–25. Cham: Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-14343-9_16.

Android.com. 2016. “Android.” https://www.android.com/.

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. 2010. “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.” In *LREC*, 10:2200–2204.

Belz, Frank-Martin, and Wenke Baumbach. 2010. “Netnography as a Method of Lead User Identification.” *Creativity and Innovation Management* 19 (3): 304–313. doi:10.1111/j.1467-8691.2010.00571.x.

Bhattacherjee, Anol. 2012. *Social Science Research: Principles, Methods, and Practices*. 2 edition. CreateSpace Independent Publishing Platform.

Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit*. O’Reilly Media.

———. 2015. “Natural Language Processing.” http://www.nltk.org/book/.

Bodendorf, F., and C. Kaiser. 2010. “Detecting Opinion Leaders and Trends in Online Communities.” In *Fourth International Conference on Digital Society, 2010. ICDS ’10*, 124–29. doi:10.1109/ICDS.2010.29.

Boyed, Danah, Kate Crawford, Melvin Kranzberg, and Geoffrey Bowker. 2012. “Critical Questions for Big Data: Provocations for a Cultural, Technological, and Scholarly Phenomenon.” *Information, Communication & Society* 15 (5): 662–679.

Brill, Eric. 1995. “Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging.” In *Proceedings of the Third Workshop on Very Large Corpora*, 30:1–13. Somerset, New Jersey: Association for Computational Linguistics.

Brockhoff, Klaus. 2003. “Customers’ Perspectives of Involvement in New Product Development.” *International Journal of Technology Management* 26 (5–6): 464–81. doi:10.1504/IJTM.2003.003418.

Burke, John G., and John Lawrence Enos. 1963. “Petroleum Progress and Profits: A History of Process Innovation.” *Technology and Culture* 4 (1): 82. doi:10.2307/3101350.

Cambria, Erik, Haixun Wang, and Bebo White. 2014. “Guest Editorial: Big Social Data Analysis.” *Knowledge-Based Systems* 69: 1–2. doi:http://dx.doi.org/10.1016/j.knosys.2014.07.002.

Christensson, P. 2011. “Web Forum Definition.” February 17. http://techterms.com/definition/web_forum.

Church, Kenneth Ward. 1988. “A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text.” In *Proceedings of the Second Conference on Applied Natural Language Processing*, 136–143. Association for Computational Linguistics.

Cutting, Doug, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. “A Practical Part-of-Speech Tagger.” In *Proceedings of the Third Conference on Applied Natural Language Processing*, 133–140. Association for Computational Linguistics.

Dave, Kushal, Steve Lawrence, and David M. Pennock. 2003. “Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews.” In *Proceedings of the 12th International Conference on World Wide Web*, 519–528. WWW ’03. New York, NY, USA: ACM. doi:10.1145/775152.775226.

DMR Stats. 2016. “YouTube Statistics.” *DMR Stats*. April 29. http://expandedramblings.com/index.php/youtube-statistics/.

Dobie, Alex. 2015. “Android History Interview: Cyanogen’s Steve Kondik.” *Android Central*. November 18. http://www.androidcentral.com/cyanogens-steve-kondik-android-history-interview.

Finley, Klint. 2013. “Out in the Open: The Place Where Android Thrives Outside of Google’s Control.” *WIRED*. December 2. http://www.wired.com/2013/12/xda/.

Flynn, Leisa Reinecke, Ronald E. Goldsmith, and Jacqueline K. Eastman. 1996. “Opinion Leaders and Opinion Seekers: Two New Measurement Scales.” *Journal of the Academy of Marketing Science* 24 (2): 137.

Fraley, C., and A. E. Raftery. 1998. “How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis.” *The Computer Journal* 41 (8): 578–88. doi:10.1093/comjnl/41.8.578.

Franke, Nikolaus, and Eric von Hippel. 2003. “Satisfying Heterogeneous User Needs via Innovation Toolkits: The Case of Apache Security Software.” *Research Policy* 32 (7): 1199–1215. doi:10.1016/S0048-7333(03)00049-0.

Franke, Nikolaus, Eric von Hippel, and Martin Schreier. 2006. “Finding Commercially Attractive User Innovations: A Test of Lead-User Theory.” *Journal of Product Innovation Management* 23 (4): 301–15. doi:10.1111/j.1540-5885.2006.00203.x.

Freeman. 1968. “Chemical Process Plant : Innovation and the World Market.” *National Institute Economic Review* 45 (1): 29–51. doi:10.1177/002795016804500104.

Gandhewar, Nisarg, and Rahila Sheikh. 2010. “Google Android: An Emerging Software Platform for Mobile Devices.” *International Journal on Computer Science and Engineering* 1 (1): 12–17.

Gartner. 2012. “Social Analytics.” *Gartner IT Glossary*. June 29. http://www.gartner.com/it-glossary/social-analytics.

Gatignon, Hubert, and Thomas S. Robertson. 1985. “A Propositional Inventory for New Diffusion Research.” *Journal of Consumer Research* 11 (4): 849–67.

Gentleman, R., and V. J. Carey. 2008. “Unsupervised Machine Learning.” In *Bioconductor Case Studies*, 137–57. Use R! Springer New York. http://link.springer.com/chapter/10.1007/978-0-387-77240-0_10.

Gliem, Rosemary R., and Joseph A. Gliem. 2003. “Calculating, Interpreting, And Reporting Cronbach’s Alpha Reliability Coefficient For Likert-Type Scales.” https://scholarworks.iupui.edu/handle/1805/344.

Go, Alec, Richa Bhayani, and Lei Huang. 2009. “Twitter Sentiment Classification Using Distant Supervision.” *Processing* 150 (12).

Hedley, Jonathan. 2016. “Jsoup Java HTML Parser, with Best of DOM, CSS, and Jquery.” https://jsoup.org/.

Herstatt, Cornelius, and Eric von Hippel. 1992. “From Experience: Developing New Product Concepts via the Lead User Method: A Case Study in a ‘low-Tech’ Field.” *Journal of Product Innovation Management* 9 (3): 213–21. doi:10.1016/0737-6782(92)90031-7.

Hienerth, Christoph, Marion Poetz, and Erich von Hippel. 2007. “Exploring Key Characteristics of Lead User Workshop Participants: Who Contributes Best to the Generation of Truly Novel Solutions?” In . http://www2.druid.dk/conferences/viewabstract.php?id=1609&cf=9.

Hudli, S.A., A.A. Hudli, and A.V. Hudli. 2012. “Identifying Online Opinion Leaders Using K-Means Clustering.” In , 416–19. doi:10.1109/ISDA.2012.6416574.

IDC Research. 2015. “IDC: Smartphone OS Market Share.” *Www.idc.com*. July. http://www.idc.com/prodserv/smartphone-os-market-share.jsp.

Internet live stats. 2016. “Twitter Usage Statistics - Internet Live Stats.” http://www.internetlivestats.com/twitter-statistics/.

Jain, Anil K. 2010. “Data Clustering: 50 Years beyond K-Means.” *Pattern Recognition Letters*, Award winning papers from the 19th International Conference on Pattern Recognition (ICPR)19th International Conference in Pattern Recognition (ICPR), 31 (8): 651–66. doi:10.1016/j.patrec.2009.09.011.

Jain, Anil K., and Richard C. Dubes. 1988. *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Jeppesen, Lars Bo, and Lars Frederiksen. 2006. “Why Do Users Contribute to Firm-Hosted User Communities? The Case of Computer-Controlled Music Instruments.” *Organization Science* 17 (1): 45–63. doi:10.1287/orsc.1050.0156.

Jeppesen, Lars Bo, and Keld Laursen. 2009. “The Role of Lead Users in Knowledge Sharing.” *Research Policy* 38 (10): 1582–89. doi:10.1016/j.respol.2009.09.002.

Jurafsky, Dan, and James H Martin. 2014. *Speech and Language Processing*. Pearson.

Karlgaard, Rich. 2005. “Ten Laws of the Modern World.” *Forbes* 175 (10): 33.

Kopinak, Janice Katherine. 1999. “The Use of Triangulation in a Study of Refugee Well-Being.” *Quality and Quantity* 33 (2): 169–83. doi:10.1023/A:1026447822732.

Kotsiantis, S. B. 2007. “Supervised Machine Learning: A Review of Classification Techniques.” In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, 3–24. Amsterdam, The Netherlands, The Netherlands: IOS Press. http://dl.acm.org/citation.cfm?id=1566770.1566773.

Kozinets, Robert V. 2002. “The Field Behind the Screen: Using Netnography for Marketing Research in Online Communities.” *Journal of Marketing Research* 39 (1): 61–72. doi:10.1509/jmkr.39.1.61.18935.

Lakhani, K. R. 2006. “Broadcast Search in Problem Solving: Attracting Solutions from the Periphery1.” In *Technology Management for the Global Future, 2006. PICMET 2006*, 6:2450–68. doi:10.1109/PICMET.2006.296842.

Lilien, Gary L., Pamela D. Morrison, Kathleen Searls, Mary Sonnack, and Eric von Hippel. 2002. “Performance Assessment of the Lead User Idea-Generation Process for New Product Development.” *Management Science* 48 (8): 1042–59.

Lincoln, Yvonna S., and Egon G. Guba. 1985. *Naturalistic Inquiry*. Expanded. Beverly Hills, Calif: Sage Pubn.

Liu, Hongzhou, Venugopalan Ramasubramanian, and Emin Gün Sirer. 2005. “Client Behavior and Feed Characteristics of RSS, a Publish-Subscribe System for Web Micronews.” In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, 3–3. USENIX Association.

Lüthje. 2004. “Characteristics of Innovating Users in a Consumer Goods Field: An Empirical Study of Sport-Related Product Consumers.” *Technovation* 24 (9): 683–95. doi:10.1016/S0166-4972(02)00150-5.

Lüthje, Christian, and Cornelius Herstatt. 2004. “The Lead User Method: An Outline of Empirical Findings and Issues for Future Research.” *R&D Management* 34 (5): 553–68. doi:10.1111/j.1467-9310.2004.00362.x.

Lyons, Barbara, and Kenneth Henderson. 2005. “Opinion Leadership in a Computer-Mediated Environment.” *Journal of Consumer Behaviour* 4 (5): 319–29. doi:10.1002/cb.22.

Malik, Sanjay Kumar, and SAM Rizvi. 2011. “Information Extraction Using Web Usage Mining, Web Scrapping and Semantic Annotation.” In *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*, 465–469. IEEE.

Mansfield, E. 1968. *Industrial Research and Technological Innovation: An Econometric Analysis*. New York, NY, USA.

Martínez-Álvarez, F., A. Troncoso, J. C. Riquelme, and J. M. Riquelme. 2007. “Partitioning-Clustering Techniques Applied to the Electricity Price Time Series.” In *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, edited by Hujun Yin, Peter Tino, Emilio Corchado, Will Byrne, and Xin Yao, 990–99. Lecture Notes in Computer Science

4881. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/978-3-540-77226-2_99.

McNeill, Mhairi. 2015. “Text Mining in R Tutorial: Term Frequency & Word Clouds.” *Deltadna.com*. April 10. https://deltadna.com/blog/text-mining-in-r-for-term-frequency/.

Meijer, Paulien C., Nico Verloop, and Douwe Beijaard. 2002. “Multi-Method Triangulation in a Qualitative Study on Teachers’ Practical Knowledge: An Attempt to Increase Internal Validity.” *Quality and Quantity* 36 (2): 145–67. doi:10.1023/A:1014984232147.

Miller, George A. 1995. “WordNet: A Lexical Database for English.” *Communications of the ACM* 38 (11): 39–41.

Montalvo, Roberto E. 2011. “Social Media Management.” *International Journal of Management & Information Systems (IJMIS)* 15 (3): 91. doi:10.19030/ijmis.v15i3.4645.

Morrison, Pamela D, John H Roberts, and David F Midgley. 2004. “The Nature of Lead Users and Measurement of Leading Edge Status.” *Research Policy* 33 (2): 351–62. doi:10.1016/j.respol.2003.09.007.

Munger, Tyler, and Jiabin Zhao. 2014. “Automatically Identifying Experts in on-Line Support Forums Using Social Interactions and Post Content.” In , 930–35. IEEE. doi:10.1109/ASONAM.2014.6921697.

Naqa, Issam El, Ruijiang Li, and Martin J. Murphy. 2015. *Machine Learning in Radiation Oncology: Theory and Applications*. Springer.

Nedelcu, Alex. 2012. “How To Build a Naive Bayes Classifier.” https://alexn.org/blog/2012/02/09/howto-build-naive-bayes-classifier.html.

Nigam, Kamal. 1999. “Using Maximum Entropy for Text Classification.” In *In IJCAI-99 Workshop on Machine Learning for Information Filtering*, 61–67.

Ning, Ma, Liu Yijun, Tian Ruya, and Li Qianqian. 2012. “Recognition of Online Opinion Leaders Based on Social Network Analysis.” In *Active Media Technology*, edited by Runhe Huang, Ali A. Ghorbani, Gabriella Pasi, Takahira Yamaguchi, Neil Y. Yen, and Beijing Jin, 7669:483–92. Berlin, Heidelberg: Springer Berlin Heidelberg. http://link.springer.com/10.1007/978-3-642-35236-2_48.

Obar, Jonathan A., and Steven S. Wildman. 2015. “Social Media Definition and the Governance Challenge: An Introduction to the Special Issue.” *SSRN Electronic Journal*. doi:10.2139/ssrn.2647377.

Open Handset Alliance. 2007. “Industry Leaders Announce Open Platform for Mobile Devices | Open Handset Alliance.” *Industry Leaders Announce Open Platform for Mobile Devices*. November 5. http://www.openhandsetalliance.com/press_110507.html.

Oracle. 2016. “MySQL :: MySQL Connector/J 5.1 Developer Guide.” *connector/J*. http://dev.mysql.com/doc/connector-j/5.1/en/.

Owen, Thomas. 2010. “Google Exec: Android Was ‘best Deal Ever.’” *VentureBeat*. October 27. http://venturebeat.com/2010/10/27/google-exec-android-was-best-deal-ever/.

Pajo, Sanjin, Paul-Armand Verhaegen, Dennis Vandevenne, and Joost R. Duflou. 2013. “Analysis of Automatic Online Lead User Identification.” In *Smart Product Engineering*, edited by Michael Abramovici and Rainer Stark, 505–14. Lecture Notes in Production Engineering. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/978-3-642-30817-8_49.

———. 2015. “Towards Automatic and Accurate Lead User Identification.” *Procedia Engineering*, TRIZ and Knowledge-Based Innovation in Science and Industry, 131: 509–13. doi:10.1016/j.proeng.2015.12.445.

Pak, Alexander, and Patrick Paroubek. 2016. “Twitter as a Corpus for Sentiment Analysis and Opinion Mining.” In , 1320–26. Accessed May 20. https://www.semanticscholar.org/paper/Twitter-as-a-Corpus-for-Sentiment-Analysis-and-Pak-Paroubek/ad8a7f620a57478ff70045f97abc7aec9687ccbd/pdf.

Pang, Bo, and Lillian Lee. 2008. “Opinion Mining and Sentiment Analysis.” *Found. Trends Inf. Retr.* 2 (1–2): 1–135. doi:10.1561/1500000011.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. “Thumbs Up?: Sentiment Classification Using Machine Learning Techniques.” In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, 79–86. EMNLP ’02. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1118693.1118704.

Peeples, Matthew. 2016. “K-Means Analysis R-Script.” Accessed May 16. http://www.mattpeeples.net/kmeans.html.

Piller, Frank T., and Dominik Walcher. 2006. “Toolkits for Idea Competitions: A Novel Method to Integrate Users in New Product Development.” *R&D Management* 36 (3): 307–18. doi:10.1111/j.1467-9310.2006.00432.x.

Poetz, Marion K., and Reinhard Prügl. 2010. “Crossing Domain-Specific Boundaries in Search of Innovation: Exploring the Potential of Pyramiding*.” *Journal of Product Innovation Management* 27 (6): 897–914. doi:10.1111/j.1540-5885.2010.00759.x.

Porter, Constance Elise. 2004. “A Typology of Virtual Communities: A Multi-Disciplinary Foundation for Future Research.” *Journal of Computer-Mediated Communication* 10 (1). http://esc-web.lib.cbs.dk/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2004-21827-003&site=ehost-live.

Rana Sobh, and Chad Perry. 2006. “Research Design and Data Analysis in Realism Research.” *European Journal of Marketing* 40 (11/12): 1194–1209. doi:10.1108/03090560610702777.

Robson, C., and K. McCartan. 2016. *Real World Research*. Wiley. http://eu.wiley.com/WileyCDA/WileyTitle/productCd-111874523X.html.

Rogers, Everett M. 2010. *Diffusion of Innovations, 4th Edition*. Simon and Schuster.

Rogers, Everett M., and David G. Cartano. 1962. “Methods of Measuring Opinion Leadership.” *The Public Opinion Quarterly* 26 (3): 435–41.

Rogers, Everett M., and F. Floyd Shoemaker. 1971. *Communication of Innovations: A Cross-Cultural Approach*. Free Press.

Russakovskii, Artem. 2010. “Custom ROMs For Android Explained - Here Is Why You Want Them.” Blog. *Android Police*. May 1. http://www.androidpolice.com/2010/05/01/custom-roms-for-android-explained-and-why-you-want-them/.

Samuel, A. L. 1959. “Some Studies in Machine Learning Using the Game of Checkers.” *IBM J. Res. Dev.* 3 (3): 210–229. doi:10.1147/rd.33.0210.

Saunders, Mark, Philip Lewis, and Adrian Thornhill. 2015. *Research Methods for Business Students*. 7. Auflage. New York: Financial Times Prent.

Sawhney, Mohanbir, Gianmario Verona, and Emanuela Prandelli. 2005. “Collaborating to Create: The Internet as a Platform for Customer Engagement in Product Innovation.” *Journal of Interactive Marketing* 19 (4): 4–17. doi:10.1002/dir.20046.

Schreier, Martin, Stefan Oberhauser, and Reinhard Prügl. 2007. “Lead Users and the Adoption and Diffusion of New Products: Insights from Two Extreme Sports Communities.” *Marketing Letters* 18 (1/2): 15–30. doi:10.1007/s11002-006-9009-3.

Schreier, Martin, and Reinhard Prügl. 2008. “Extending Lead-User Theory: Antecedents and Consequences of Consumers’ Lead Userness*.” *Journal of Product Innovation Management* 25 (4): 331–346. doi:10.1111/j.1540-5885.2008.00305.x.

Sorrells, Mitra. 2016. “Infomous - Super Bowl XLVII on Twitter.” *Infomous*. Accessed May 31. http://www.infomous.com/site/events/NFL/.

Statista. 2016. “Leading Global Social Networks 2016 | Statistic.” *Statista*. April. http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.

Stieglitz, Stefan, Linh Dang-Xuan, Axel Bruns, and Christoph Neuberger. 2014. “Social Media Analytics: An Interdisciplinary Approach and Its Implications for Information Systems.” *Business & Information Systems Engineering* 6 (2): 89–96. doi:10.1007/s12599-014-0315-7.

Stockstrom, Christoph S., René Chester Goduscheit, Christian Lüthje, and Jacob Høj Jørgensen. 2016. “Identifying Valuable Users as Informants for Innovation Processes: Comparing the Search Efficiency of Pyramiding and Screening.” *Research Policy* 45 (2): 507–16. doi:10.1016/j.respol.2015.11.002.

Stone, Robert N. 1984. “The Marketing Characteristics of Involvement.” *Advances in Consumer Research* 11 (1): 210–15.

Sugiyama, Masashi, and Motoaki Kawanabe. 2012. *Machine Learning in Non-Stationary Environments - Introduction to Covariate Shift Adaptation*. Adaptive Computation and Machine Learning. MIT Press. http://site.ebrary.com.esc-web.lib.cbs.dk/lib/kbhnhh/reader.action?docID=10547396&ppg=16.

Toutanova, Kristina, and Christopher D Manning. 2000. “Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger.” In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, 63–70. Association for Computational Linguistics.

Twitter. 2016. “API Rate Limits.” *Twitter Developers*. https://dev.twitter.com/rest/public/rate-limiting.

Urban, Glen I., and Eric Von Hippel. 1988. “Lead User Analyses for the Development of New Industrial Products.” *Management Science* 34 (5): 569–82.

Vatrapu, Ravi. 2013. “Understanding Social Business.” In *Emerging Dimensions of Technology Management*, edited by K.B. Akhilesh, 147–58. India: Springer India. http://link.springer.com/10.1007/978-81-322-0792-4_11.

vBulletin. 2016. “vBulletin - about Us.” http://www.vbulletin.com/en/about-us/.

von Hippel, Eric. 1986. “Lead Users: A Source of Novel Product Concepts.” *Management Science* 32 (7): 791–805.

———. 2005. *Democratizing Innovation*. Cambridge, Mass.: MIT Press.

von Hippel, Eric, Nikolaus Franke, and Reinhard Prügl. 2009. “Pyramiding: Efficient Search for Rare Subjects.” *Research Policy* 38 (9): 1397–1406. doi:10.1016/j.respol.2009.07.005.

von Hippel, Eric von, Stefan Thomke, and Mary Sonnack. 1999. “Creating Breakthroughs at 3M.” *Harvard Business Review*. September 1. https://hbr.org/1999/09/creating-breakthroughs-at-3m.

Voutilainen, Atro. 2005. “Part-of-Speech Tagging.” *The Oxford Handbook of Computational Linguistics*, 219–232.

Wu, Chao, Chunlin Li, Wei Yan, Youlong Luo, Xijun Mao, Shumeng Du, and Mingming Li. 2015. “Identifying Opinion Leader in the Internet Forum.” http://www.sersc.org/journals/IJHIT/vol8_no11_2015/38.pdf.

XDA Changelog. 2014. “XDA Changelog.” *Xda-Developers*. http://www.xda-developers.com/changelog/.

Xu, Rui, and D. Wunsch. 2005. “Survey of Clustering Algorithms.” *IEEE Transactions on Neural Networks* 16 (3): 645–78. doi:10.1109/TNN.2005.845141.

Zeng, D., H. Chen, R. Lusch, and S. H. Li. 2010. “Social Media Analytics and Intelligence.” *IEEE Intelligent Systems* 25 (6): 13–16. doi:10.1109/MIS.2010.151.

Zhai, Z., H. Xu, and P. Jia. 2008. “Identifying Opinion Leaders in BBS.” In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT ’08*, 3:398–401. doi:10.1109/WIIAT.2008.37.

Appendix

1 Database Tables

TABLE: **URLTABLE**

#	Name	Typ	Kollation
1	**ID**	varchar(8)	latin1_swedish_ci
2	**PID**	int(11)	
3	**URL**	varchar(200)	latin1_swedish_ci

TABLE: **MASTERDATA**

#	Name	Typ	Kollation
1	**IDx**	int(6)	
2	**post_primary**	int(6)	
3	**post_id**	int(6)	
4	**thread_id**	int(10)	
5	**post_timestamp**	datetime	
6	**post_post**	text	latin1_swedish_ci
7	**post_topics**	text	latin1_swedish_ci
8	**post_pol**	float	
9	**post_sub**	float	
10	**post_length**	int(6)	
11	**post_thanks**	int(5)	
12	**member_name**	varchar(80)	latin1_swedish_ci
13	**member_type**	varchar(80)	latin1_swedish_ci
14	**member_thanks**	int(5)	
15	**member_posts**	int(5)	
16	**member_joindate**	datetime	
17	**post_quotes**	int(3)	
18	**post_quote0**	text	latin1_swedish_ci
19	**post_quote_name0**	varchar(80)	latin1_swedish_ci
20	**post_quote_pol0**	float	
21	**post_quote1**	text	latin1_swedish_ci
22	**post_quote_name1**	varchar(80)	latin1_swedish_ci
23	**post_quote_pol1**	float	
24	**post_quote2**	text	latin1_swedish_ci
25	**post_quote_name2**	varchar(80)	latin1_swedish_ci
26	**post_quote_pol2**	float	
27	**post_quote3**	text	latin1_swedish_ci
28	**post_quote_name3**	varchar(80)	latin1_swedish_ci
29	**post_quote_pol3**	float	
30	**post_quote4**	text	latin1_swedish_ci
31	**post_quote_name4**	varchar(80)	latin1_swedish_ci
32	**post_quote_pol4**	float	

TABLE: **ANALYSIS**

#	Name	Typ	Kollation
1	**ID**	int(5)	
2	**member_name**	varchar(80)	latin1_swedish_ci
3	**involvement**	int(5)	
4	**frequency**	int(5)	
5	**response**	int(5)	
6	**post_mentions**	int(5)	
7	**feedback_pos**	int(5)	
8	**feedback_pos_rel**	float	No
9	**feedback_neg**	int(5)	
10	**feedback_neg_rel**	float	No
11	**avmessagesize**	int(5)	
12	**member_joindate**	datetime	
13	**member_lastactivity**	datetime	
14	**time_diff_months**	int(11)	

TABLE: **OPNIONLEADER_REL**

#	Name	Typ	Kollation
1	**ID**	int(5)	
2	**member_name**	varchar(80)	latin1_swedish_ci
3	**frequency**	int(5)	
4	**response**	int(5)	
5	**post_mentions**	int(5)	
6	**feedback_pos**	int(5)	
7	**feedback_neg**	int(5)	
8	**avmessagesize**	int(5)	
9	**involvement**	int(5)	

TABLE: **SENTIMENTS**

#	Name	Typ	Kollation
1	**member_name**	varchar(80)	latin1_swedish_ci
2	**involvement**	int(11)	
3	**av_post_pol**	float	
4	**av_post_sub**	float	

TABLE: **TEXTMINING**

#	Name	Type	Collation
1	**int_id**	int(11)	
2	**period**	varchar(20)	latin1_swedish_ci
3	**words**	mediumtext	latin1_swedish_ci
4	**ranking**	text	latin1_swedish_ci

TABLE: **TEXTMINING_RES**			
#	**Name**	**Type**	**Collation**
1	**term**	varchar(100)	latin1_swedish_ci
2	**member_name**	varchar(80)	latin1_swedish_ci
3	**post_topics**	text	latin1_swedish_ci
4	**post_timestamp**	datetime	

2 Program process

delete content of tables if necessary.
TRUNCATE URLtable;
TRUNCATE masterdata;
TRUNCATE analysis;
TRUNCATE opinionleader_rel;

ACQUISITION THE DATA

Run Java Program: A1_parseURL.java
The URLTABLE is the basis for the scrapping of the main data since it contains all the URLs

Run Java Program: A2_ParseContent.java
This program is the core element of the data acquisition part.
Number of posts downloaded in a one-week period from 7.05.2016 - 14.05.2016 -> 101.655
Average Download Speed -> ca. 1000 posts per hour

Number of unique posters -> 14901
Date of first post: 2009-01-06 20:36:00
Date of last post: 2016-05-11 23:29:00

DATA CLEANSING

Remove Apostrophe from user names
UPDATE masterdata SET member_name = REPLACE(member_name, "'", "");

Remove Quotes from Posts
Each post can contain 0...n quotes from other users. The program **A2_ParseContent.java** already identified the quotes in a post and copied up to 5 quotes in extra data elements. So they can be used for further processing. In order to do sentiment analysis and further processing with the main post, the quotes need to be subtracted from the main post now.
If the writer of the post used more than 5 quotes, they need to be deleted manually. In 100.000 posts this happened 24 times.
remove manually where post_quotes > 5
SELECT * FROM `masterdata` WHERE post_quotes >= 5 ORDER BY `post_id` ASC
And set post_quotes to the max. possible value:
UPDATE masterdata SET post_quotes = "5" WHERE post_quotes > 5;

UPDATE masterdata SET post_post = REPLACE(post_post, post_quote0, "") WHERE post_quote0 IS NOT NULL;
UPDATE masterdata SET post_post = REPLACE(post_post, post_quote1, "") WHERE post_quote1 IS NOT NULL;

```
UPDATE masterdata SET post_post = REPLACE( post_post, post_quote2, "" ) WHERE post_quote2 IS
NOT NULL;
UPDATE masterdata SET post_post = REPLACE( post_post, post_quote3, "" ) WHERE post_quote3 IS
NOT NULL;
UPDATE masterdata SET post_post = REPLACE( post_post, post_quote4, "" ) WHERE post_quote4 IS
NOT NULL;

UPDATE masterdata SET post_post = REPLACE( post_post, "Quote: ", "" );

UPDATE masterdata SET post_quote0 = REPLACE( post_quote0, "Quote: Originally Posted by ", "" );
UPDATE masterdata SET post_quote1 = REPLACE( post_quote1, "Quote: Originally Posted by ", "" );
UPDATE masterdata SET post_quote2 = REPLACE( post_quote2, "Quote: Originally Posted by ", "" );
UPDATE masterdata SET post_quote3 = REPLACE( post_quote3, "Quote: Originally Posted by ", "" );
UPDATE masterdata SET post_quote4 = REPLACE( post_quote4, "Quote: Originally Posted by ", "" );
```

Count a column up! → Create an ID to enable Java tool
As a last step for modifications of the table MASTERDATA a new internal ID for each row of the table is given. This is necessary for the programs that will follow in this process.

```
SELECT @i:=0;
UPDATE masterdata SET IDx = @i:=@i+1;
```

SENTIMENT ANALYSIS

Run python script: sentiment analysis
For the 100.000 entries the sentiment analysis needed roughly 24hrs. calculation time.

ANALYSIS FUNCTIONS FOR INVOLVEMENT

```
INSERT INTO analysis (member_name, involvement)
SELECT member_name, COUNT( * )
FROM masterdata
GROUP BY member_name
ORDER BY COUNT( * ) DESC;
```

Get member_joindate
In order to get the length of membership the joindate of the member and the time of the last post is identified and stored for each user individually.

```
UPDATE analysis a INNER JOIN (
SELECT member_name, member_joindate FROM masterdata GROUP BY member_name)
b ON a.member_name = b.member_name
SET a.member_joindate = b.member_joindate;
```

Get member_lastactivity and write to analysis table

```
UPDATE analysis a INNER JOIN (
SELECT member_name, MAX(post_timestamp) max_timestamp
FROM masterdata GROUP BY member_name)
b ON a.member_name = b.member_name
SET a.member_lastactivity = b.max_timestamp;
```

Calculate Time Difference in Months

```
UPDATE analysis a INNER JOIN (
SELECT member_name, TIMESTAMPDIFF (MONTH, member_joindate, member_lastactivity) time_diff
FROM analysis)
b ON a.member_name = b.member_name
SET a.time_diff = b.time_diff;
```

Get frequency
The frequency score sets the involvement score in relation to a certain period of time.

```
UPDATE analysis a INNER JOIN (
SELECT member_name, involvement/time_diff_months*10 freq FROM analysis)
b ON a.member_name = b.member_name
SET a.frequency = b.freq;
```

We defined certain criteria in order to determine if a user is actively involved on a regular basis in the forum. Users that do not fit into those criteria will be deleted in the following steps:
Delete members where member_joindate > 01.02.2016
DELETE from analysis where member_joindate > '2016-02-01 00:00:00'
→ 30 Users were filtered out.

Delete members where response <1
DELETE from analysis where response = '0'
→ 5413 Users were filtered out.

Delete members where involvement < 5
DELETE from analysis where involvement < '5'
→ 6287 Users were filtered out.

→ From the 14901 users we retrieved in the first place there are 3171 users left for our further analysis.

Count a column up! → Create an ID to enable Java tool
After deleting not relevant users, a new internal ID for each row of the table is given. This is necessary for the programs that will follow in this process.

```
SELECT @i:=0;
UPDATE analysis SET ID = @i:=@i+1;
```

ANALYSIS FUNCTIONS FOR OPINION LEADER

Run Java Program: getAbsoluteResponse.java
Import Parameters: None
This program goes over every user and determines how many times each of his posts was quoted by other users.

Run Java Program: getAbsolutePostMentions.java
Import Parameters: None
This program identifies the number of mentions of each username in all other posts.

Feedback_pos
The following SQL statement aggregates all “thanks” from other users for all posts of one user. The gathered number is stored into the analysis table with the corresponding username.

```
UPDATE analysis a INNER JOIN (
SELECT member_name, SUM( `post_thanks` ) `post_thanks`
FROM masterdata
GROUP BY member_name
)b ON a.member_name = b.member_name
SET a.feedback_pos = post_thanks;
```

feedback_pos_rel
In a second statement a relative score is generated from the total number of thanks and the total number of posts (involvement) of that user. This score is also stored into the analysis table.

```
UPDATE analysis a INNER JOIN (
SELECT member_name, feedback_pos/involvement*100 rel FROM analysis)
```

b ON a.member_name = b.member_name
SET a.feedback_pos_rel = b.rel;

feedback_neg

Run Java Program: B3_getAbsNegFeedback
Import Parameters: None

feedback_neg_rel
This absolute number is then also converted into a relative score dependent on the number of mentions, the quotes which are related to the posts with negative sentiment and the total number of posts of that user.

UPDATE analysis a INNER JOIN (
SELECT member_name, feedback_neg/(post_mentions+response)/involvement*100 rel FROM analysis)
b ON a.member_name = b.member_name
SET a.feedback_neg_rel = b.rel;

Averagesizemessage
This SQL Statement generates the average text length (number of characters of posts) of all posts of one user. This number is also stored into the analysis table in the row with the corresponding user.
UPDATE analysis a
INNER JOIN (
SELECT member_name, AVG(post_length)`av`
FROM masterdata
GROUP BY member_name)
b ON a.member_name = b.member_name
SET avmessagesize = av;

Aggregate Results for analysis
In the first analysis round we created a further table in order to get a place to store the results of the analysis analysis results.
CREATE TABLE opinionleader_rel LIKE analysis;
In the second time the table was already created and needed to be cleared.
TRUNCATE opinionleader_rel

Insert ID and member_names
In this new table the users with their IDs were inserted with this statement:
INSERT INTO opinionleader_rel(member_name, ID)
SELECT member_name, ID
FROM analysis;

Run Java Program: C1_getRelativeOpinionLeader.java
Import Parameters: Scale
This program does two major steps. First it determines the highest entry for all the cetegories that are defined in the analysis table. (**involvement, frequency, response, post_mentions, feedback_pos_rel, feedback_neg_rel, avmessagesize**)
The Import Parameter of this program is a value that determines the resolution of the scale. The value should be a power of ten. If the values are very distributed it makes sense to set this value to 1000.

Creation of View
As a next step, we created a database view in order to prepare the data for exporting as CSV file.
SELECT Member_name, involvement, frequency, response, post_mentions, feedback_pos, feedback_neg, avmessagesize FROM opinionleader_rel;
The result of this View is the following table:

The CSV file could be imported to the statistic program R for the k-means cluster Analysis.

```
x <- read.csv("opinion.csv", header=TRUE, row.names=1)

wss <- (nrow(x)-1)*sum(apply(x,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(x,center=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Cluster", ylab="Within groups sum of squares")

fit <- kmeans(x, 4)

y <- data.frame(x, fit$cluster)
```

ANALYSIS OF PRODUCT DISSATISFACTION

Keyword “Android” / “Stock Android” “Standard Android”
We select for each user all posts that are related to the search term “Android” and aggregate the polarity and subjectivity only for these posts.
DB-Table: Sentiments

```
INSERT INTO sentiments( member_name, involvement )
SELECT member_name, involvement
FROM analysis;

UPDATE sentiments a
INNER JOIN (
SELECT member_name, AVG( post_pol )`av_pol`, AVG( post_sub  )`av_sub`
FROM masterdata WHERE post_post LIKE ('%Android%')
GROUP BY member_name)
b ON a.member_name = b.member_name
SET av_post_pol = av_pol, av_post_sub = av_sub ;
```

→ **1234 of the 3169 Users talk about “Android” topics in general**

DATA ANALYSIS FOR “AHEAD OF TREND”

Run Python Program phraser.py
This program is a Sentence phrase tokenizer based on NLTK and the Brown library.
It extracts only the nouns from the sentences and fills it in post_topics of the masterdata table

→ Analysing the 100.000 posts took ca. 24std.

Run the Java Program D1_getTextminingTable
This program aggregates the tokens into 65 monthly intervals. Starting from Jan 2011 until May 2016.
This program uses basically this SQL statement for each interval:

```
SELECT GROUP_CONCAT(DISTINCT post_topics SEPARATOR ' ') FROM masterdata WHERE
YEAR(post_timestamp) = 2011 AND MONTH(post_timestamp) = 1
```

We will now further clean this data set with R and get the term frequencies.
With R and mySQL
Sources:

Import the following libraries

```
library(tm)
library(qdap)
library(qdapDictionaries)
library(dplyr)
library(RColorBrewer)
library(ggplot2)
library(scales)
```

```
library(Rgraphviz)
library(wordcloud)
library(RMySQL)
```

Connect to MySQL
→ R Statements:

```
mydb = dbConnect(MySQL(), user='root', password='CBS2015', dbname='masterthese2013',
host='127.0.0.1')
rs = dbSendQuery(mydb, "SELECT int_id, words FROM textmining;")
data = fetch(rs, n=-1)
corp <- Corpus(DataframeSource(data))
inspect(corp)
```

Display the first document from Corpus:

```
strwrap(corp[[1]])
```

Further cleaning of data

```
Docs = corp

toString <- content_transformer(function(x, from, to) gsub(from, to, x))
docs <- tm_map(docs, toString, "forum.xda-developers.com", "")

toSpace <- content_transformer(function(x, pattern) gsub(pattern, " ", x))

docs <- tm_map(docs, removeNumbers)
docs <- tm_map(docs, toSpace, "u'")
docs <- tm_map(docs, toSpace, "uu")
docs <- tm_map(docs, removePunctuation)
docs <- tm_map(docs, toSpace, "http")
docs <- tm_map(docs, toSpace, "www")

docs <- tm_map(docs, removeWords, stopwords("english"))
docs <- tm_map(docs, stripWhitespace)
```

Remove specific words:

```
docs <- tm_map(docs, removeWords, c("department", "email"))
```

→ Words to be removed in separate file.

Check single Documents

```
strwrap(docs[[1]])
```

Creating a document term matrix

```
dtm <- DocumentTermMatrix(docs)
```

Exploring the Document Term Matrix

```
freq <- colSums(as.matrix(dtm))
length(freq)
ord <- order(freq)
freq[tail(ord)]

class(dtm)
dim(dtm)
```

Transpose:

```
tdm <- TermDocumentMatrix(docs)
```

Removing Sparse Terms

```
dim(dtm)
dtms <- removeSparseTerms(dtm, 0.2)
dim(dtms)
inspect(dtms)
```

→ 207 terms remain

Plot words

```
freq <- colSums(as.matrix(dtms))
length(freq)
[1] 207
ord <- order(freq)
freq
```

→ words in order of their frequencies

Export Words as CSVs

Document Term Matrix

```
m <- as.matrix(dtm)
dim(m)
[1]    65 61546
write.csv(m, file="dtms_final.csv")
```

Word Frequencies

```
n <- as.matrix(freq)
dim(n)
[1] 207   1
write.csv(n, file="freq_final.csv")
```

Plot correlations

```
plot(dtms,
terms=findFreqTerms(dtms, lowfreq=100)[1:40],
corThreshold=0.5)
```

Plot Word Frequencies

```
subset(wf, freq>500) %>%
ggplot(aes(word, freq)) +
geom_bar(stat="identity") +
theme(axis.text.x=element_text(angle=45, hjust=1))
```

Plotting Word Clouds

```
set.seed(123)
wordcloud(names(freq), freq, min.freq=40)
set.seed(142)
wordcloud(names(freq), freq, max.words=100)
```

Colored version:

```
set.seed(142)
wordcloud(names(freq), freq, min.freq=100, colors=brewer.pal(6, "Dark2"))
set.seed(142)
dark2 <- brewer.pal(6, "Dark2")
wordcloud(names(freq), freq, min.freq=100, rot.per=0.2, colors=dark2)
```

Calculate Growth of Words

Get Average Growth of Words over all time intervals of 65 months.
Show words with the high growth rates → Words that come up as trend words, Low rate of mentions in the beginning and strong increase
Show words with low growth rates. → Words that are continuously used a lot.

Get correlated Users
SQL-Lookups for users that used these terms of high growth first.

Examples for word “tasker” and “miui”

```
INSERT INTO masterthese2013.textmining_res (member_name, post_timestamp, post_topics, term)
SELECT member_name, post_timestamp, post_topics, 'tasker'  FROM `masterdata` WHERE `post_post`
LIKE '%tasker%' ORDER BY post_timestamp ASC LIMIT 5

INSERT INTO masterthese2013.textmining_res (member_name, post_timestamp, post_topics, term)
SELECT member_name, post_timestamp, post_topics, 'miui'  FROM `masterdata`
WHERE `post_post` LIKE '%miui%' ORDER BY post_timestamp ASC LIMIT 5;
```

Create Views for Export to Excel

List with all members
export_members_all

Create view with involvement
export_members_involved

Create view with all opinionleaders
export_analysis
export_analysis_sort

Create view with all dissatisfied members
export_members_diss

Create View members results from textmining
Export_tm_users
Export_members_tm
Example SQL-Statement for creating this view:

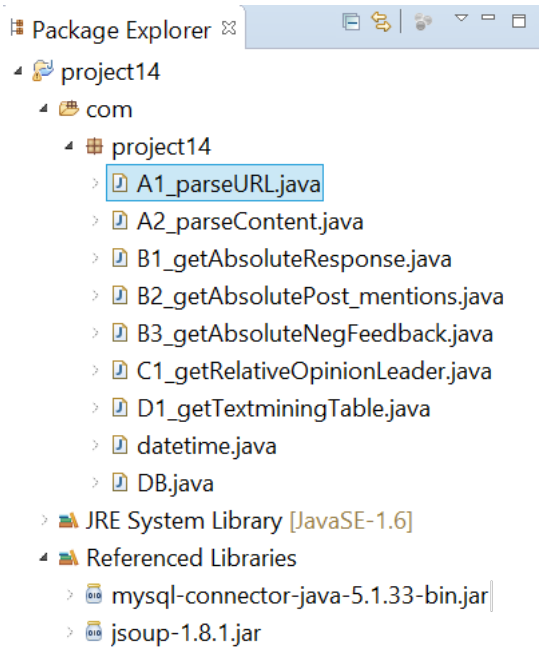
```
CREATE VIEW export_members_tm
AS SELECT a.member_name, involvement, term, post_topics, post_timestamp
FROM textmining_res a, analysis b
WHERE a.member_name=b.member_name;

SELECT member_name,
GROUP_CONCAT(DISTINCT post_quote_name0 SEPARATOR '; ') quoter0,
GROUP_CONCAT(DISTINCT post_quote_name1 SEPARATOR '; ') quoter1,
GROUP_CONCAT(DISTINCT post_quote_name2 SEPARATOR '; ') quoter2,
GROUP_CONCAT(DISTINCT post_quote_name3 SEPARATOR '; ') quoter3,
GROUP_CONCAT(DISTINCT post_quote_name4 SEPARATOR '; ') quoter4
FROM masterdata GROUP BY member_name
ORDER BY `masterdata`.`member_name`  DESC

CREATE VIEW export_members_network
```

3 Java Code

Overview



A1_parseURL.java

```
package project14;

import java.io.IOException;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;


public class A1_parseURL {
    public static void connection(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("driver ok");
        } catch (ClassNotFoundException e){
            e.printStackTrace();
        }
    }
    public static void ConnectionToMySql (String ID, int postcountint, String printpostcount){
        connection();

        //String host = "jdbc:mysql://127.0.0.1:3306/masterthese";
        String host = "jdbc:mysql://localhost:3306/masterthese";
        //String host = "jdbc:mysql://144.76.19.105:3306/masterthese";
        String username = "root";
        String password = "CBS";
        try {
            Connection connect = DriverManager.getConnection(host, username, password);
            System.out.println("connected to mySQL db");
            String sql = "INSERT INTO URLtable(ID, PID, URL)VALUES (?,?,?)";

            PreparedStatement statement = (PreparedStatement) connect.prepareStatement(sql);

            statement.setString (1,ID);
            statement.setInt    (2,postcountint);
```

```
            statement.setString (3,printpostcount);
            statement.executeUpdate();

            statement.close();
            connect.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public static void main (final String[] args) throws IOException {

        //INSERT VARIABLES HERE:
        //HTML address of Thread
        String html = "http://forum.xda-developers.com/showthread.php?t=991276";
        //int threadid = 1943625;
        //Page number of first Thread-page
        int pnum = 1293;


        //Get highest Page-number

        //Jsoup connection
        Document doc1 = Jsoup.connect(html +"&page="+ pnum)
                .userAgent("Mozilla").timeout(6000).get();

        //get number of pages
        int j = 0;
        Elements pages = doc1.select("span[class=pageofpages vbmenu_control]");
        Element printpages = pages.get (j);
        System.out.println("Seitenanzahl: " + printpages.text());
        //modify result
        String modpages = printpages.text();
        String before1 = "of ";
        //String after1 = " ÊÊ 1 2 3";
        //Seitenanzahl: Page 1 of 415    //public String substring(int beginIndex, int endIndex)
        String totalpages = modpages.substring(modpages.indexOf(before1) + before1.length());
        System.out.println("Seitenanzahl: " + totalpages);
        //transform String into Integer
        int y = Integer.parseInt(totalpages);


        // While-Loop for Pages
        while  (pnum <= y)
        {
            //Jsoup connection
            Document doc = Jsoup.connect(html +"&page="+ pnum)
                    .userAgent("Mozilla").timeout(20000).get();

            //Define Locations in HTML


            // For-Loop for Post-entries
            for (int i = 0; i <= 9; i++){
                // 10 Entries per page. After 10th (i=9) post Page-ID (pnum) raises one
                if (i == 9) {pnum=pnum+1;}

                //Postinfo
                Elements postcount = doc.select("a[class=postCount]");
                Elements postbit = doc.select("a[class=postbit-anchor]");
                //Get Data from HTML
                Element printpostcount = postcount.get(i);
                Element printpostbit = postbit.get(i);

                // Print and modify Data

                System.out.println("i: " + i);
                System.out.println("Pagenumber: " + pnum);
                System.out.println("Postcount: " + printpostcount.text());
                System.out.println("Postbit: " + printpostbit.attr("id"));
                System.out.println("Posturl: " + printpostcount.attr("href"));


                //modify ID
```

```
                String modID = printpostcount.attr("href");
                String beforeID = "&p=";
                String afterID = "&postcount";
                //showpost.php?s=4b85e5faa8e599d774f6936c9af9f8e3&p=38734709&postcount=108
//public String substring(int beginIndex, int endIndex)
                String ID = modID.substring(modID.indexOf(beforeID) + beforeID.length(),
modID.indexOf(afterID));
                System.out.println("ID: " + ID);

                String modprintpostcount = printpostcount.text();
                String postcountnew = modprintpostcount.substring(1);
                int postcountint = Integer.parseInt(postcountnew);
                System.out.println("Postcount: " + postcountint);

                //Leading Post has missing Postcount
                String postbit1 = printpostbit.attr("id").substring(4);
                System.out.println("ID1: " + postbit1);

                if (pnum == 1) {
                    //http://forum.xda-developers.com/showpost.php?p=39178981&postcount=1

                    postcountint=postcountint-1;
                    System.out.println("Posturl_alternative: " + "http://forum.xda-
developers.com/showpost.php?p="+postbit1+"&postcount="+postcountint);
                    ConnectionToMySql (postbit1, postcountint, "http://forum.xda-
developers.com/showpost.php?p="+postbit1+"&postcount="+postcountint);
                }

                else {
                    //Write to mysql db

                    ConnectionToMySql (ID, postcountint, "http://forum.xda-
developers.com/"+printpostcount.attr("href"));
                }
                System.out.println("-----------------------------------------------");
            }
        }
    }

}
```

A2_parseContent.java

```
package project14;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;


public class A2_parseContent {



    public static void connection(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("driver ok");
        } catch (ClassNotFoundException e){
```

```
            e.printStackTrace();
        }
    }
    public static void ConnectionToMySql (int IDx, int IDint, int postcountint, int threadid,
String printdates, String printposts, String post_topics, int post_pol, int post_sub,int length,
int thanksglobalint, String printpostersnames, String membertype, int memberthanksint, int
memberpostsint, String memberjoin, int quoteint, String postquote0, String postquote_name0, int
postquote_pol0, String postquote1, String postquote_name1, int postquote_pol1, String postquote2,
String postquote_name2, int postquote_pol2, String postquote3, String postquote_name3, int
postquote_pol3, String postquote4, String postquote_name4, int postquote_pol4){
        connection();
        String host = "jdbc:mysql://127.0.0.1:3306/masterthese";
        // 144.76.19.105
        String username = "root";
        String password = "CBS";
        try {
            Connection connect = DriverManager.getConnection(host, username, password);
            System.out.println("connected to mySQL db");
            String sql = "INSERT INTO masterdata(IDx, post_primary, post_id, thread_id,
post timestamp, post post, post topics, post pol, post sub, post length, post thanks,
member name, member type, member thanks, member posts, member joindate, post quotes, post quote0,
post quote name0, post quote pol0, post quote1, post quote name1, post quote pol1, post quote2,
post_quote_name2, post_quote_pol2, post_quote3, post_quote_name3, post_quote_pol3, post_quote4,
post_quote_name4, post_quote_pol4)VALUES
(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

            PreparedStatement statement = (PreparedStatement) connect.prepareStatement(sql);

            statement.setInt    (1,IDx);
            statement.setInt    (2,IDint);
            statement.setInt    (3,postcountint);
            statement.setInt    (4,threadid);
            statement.setString(5,printdates);
            statement.setString(6,printposts);
            statement.setString(7,post_topics);
            statement.setInt    (8,post_pol);
            statement.setInt    (9,post_sub);
            statement.setInt    (10,length);
            statement.setInt    (11,thanksglobalint);
            statement.setString(12,printpostersnames);
            statement.setString(13,membertype);
            statement.setInt    (14,memberthanksint);
            statement.setInt    (15,memberpostsint);
            statement.setString(16,memberjoin);
            statement.setInt    (17,quoteint);
            statement.setString(18,postquote0);
            statement.setString(19,postquote_name0);
            statement.setInt    (20,postquote_pol0);
            statement.setString(21,postquote1);
            statement.setString(22,postquote_name1);
            statement.setInt    (23,postquote_pol1);
            statement.setString(24,postquote2);
            statement.setString(25,postquote_name2);
            statement.setInt    (26,postquote_pol2);
            statement.setString(27,postquote3);
            statement.setString(28,postquote_name3);
            statement.setInt    (29,postquote_pol3);
            statement.setString(30,postquote4);
            statement.setString(31,postquote_name4);
            statement.setInt    (32,postquote_pol4);
            statement.executeUpdate();

            statement.close();
            connect.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }



    public static DB db = new DB();
    public static String URL;
    public static String ID;
```

```
    public static String sqldate;
    public static String memberdate;
    public static int maxpidint;
    //INSERT THREAD-ID HERE          vvvv
    public static int threadid = 991276;
    public static Map<String, String> map = new HashMap<String, String>();


    public static void main (final String[] args) throws SQLException, IOException {

        String sql = "SELECT MAX(PID) FROM URLtable;";
        ResultSet rs = db.runSql(sql);
        while (rs.next()) {
            String Maxid = rs.getString(1);
            System.out.println(Maxid);
            maxpidint = Integer.parseInt(Maxid);
        }
        //INSERT START-ID HERE   vvvv
        for (int i = 19461; i <= maxpidint; i++){
            String sql1 = "SELECT URL FROM URLtable WHERE PID ="+i+";";
            String sql2 = "SELECT ID FROM URLtable WHERE PID ="+i+";";
            ResultSet rs1 = db.runSql(sql1);
            while (rs1.next()) {
                URL = rs1.getString(1);
                System.out.println(URL);
            }
            ResultSet rs2 = db.runSql(sql2);
            while (rs2.next()) {
                ID = rs2.getString(1);
                System.out.println(ID);
            }

            //Jsoup connection
            Document doc = Jsoup.connect(URL)
                    .userAgent("Mozilla").timeout(20000).get();


            //Define Locations in HTML

            //Analyze Number of quotes in post
            Elements postquote = doc.select("div[class=bbcode-quote-text]");
            //ALT:  table[cellpadding=6][cellspacing=0] tr:contains(Originally Posted by)
            //Elements all quotes = doc.getElementsContainingText("Originally Posted by");
table[cellpadding=6][cellspacing=0]  tr:contains(Originally Posted by)
            //p:contains(Originally Posted by);
            int quoteint = (postquote.size());
            System.out.println(quoteint);

            //.size()!=0

            //Postinfo

            Elements postcount = doc.select("a[class=postCount]");
            Elements dates = doc.select("span[class=time]");
            Elements posts = doc.select("div[id^=post_message_]");
            Elements thanks = doc.select("div[id^=post_thanks_box_]");

            //Userinfo

            Elements posternames = doc.select("a.bigfusername");
            Elements memberinfo = doc.select("div[class=postbit-userinfo-cell]"); //for member-
status
            Elements memberthanks0 = doc.select("span[class=userThanksText]");
            Elements memberposts0 = doc.select("div[class=pbuser user-posts]");
            Elements memberjoin0 = doc.select("div[class=pbuser user-joindate]");
            //Element countryimg = doc.select("div[class=moreUserInfo] img").first();

            int y = 0;

            //Get Data from HTML
            Element printpostcount = postcount.get(y);
            Element printdates = dates.get(y);
            Element printposts = posts.get(y);
```

```
Element printthanks = thanks.get(y);
Element printposternames;

if (posternames.isEmpty()){
    continue;
}
else{
    printposternames = posternames.get(y);
}
Element printmemberinfo = memberinfo.get(y);
Element printmemberthanks0 = memberthanks0.get(y);
Element printmemberposts0 = memberposts0.get(y);
Element printmemberjoin0 = memberjoin0.get(y);

//String country = (countryimg.attr("title"));


// Print and modify Data


System.out.println("Postcount: " + printpostcount.text());
//System.out.println("i: " + i);
System.out.println("Timestamp: " + printdates.text());

// Convert Date to sql readable format

String datestring = printdates.text();
datestring = datestring.replaceAll("([0-9]{1,2})(st|nd|rd|th)(.*)", "$1$3");
System.out.println("Timestamp reduced: "+ datestring);

String inputformat ="dd MMMMM yyyy, hh:mm a";
String format1 ="yyyy-MM-dd HH:mm:ss";
//format needed for sql: YYYY-MM-DD HH:MM:SS
SimpleDateFormat sdf = new SimpleDateFormat(inputformat, Locale.ENGLISH);
SimpleDateFormat sdf1 = new SimpleDateFormat(format1);

try {
    Date date = sdf.parse(datestring);
    System.out.println(date);
    System.out.println(sdf1.format(date));
    sqldate = (sdf1.format(date));
} catch (ParseException ex) {
    ex.printStackTrace();
}

int length = printposts.text().length( );
System.out.println("Post-Length: " + length);
System.out.println("Post: " + printposts.text());


// process quotes and extract usernames of "Originally Posted by "

if (quoteint>0) {

    for (int j = 0; j < quoteint; j++){

        Element printpostquote = postquote.get(j);
        //System.out.println("Postquote: " + printpostquote.text());
        String postquotevar = printpostquote.text();
        String postquotevarname = printpostquote.html();

        //System.out.println("Postquote: "+postquotevar);
        map.put("postquote"+j, postquotevar);

        // get keyset value from map
        // Set keyset=map.keySet();

        // check key set values
        // System.out.println("Key set values are: " + keyset);

        //conditionally get
        System.out.println("postquote : " + map.get("postquote"+j));

        // get name of quoted user
```

```
                    if (postquotevarname.toLowerCase().contains("<strong>".toLowerCase())) {

                        String before = "<strong>";
                        String after = "</strong>";
                        postquotevarname =
postquotevarname.substring(postquotevarname.indexOf(before) + before.length(),
postquotevarname.indexOf(after));

                        System.out.println(j + " Posted by: " + postquotevarname);

                        map.put("postquote_name"+j, postquotevarname);
                    }


                    // deduct the quote length from post length
                    int lengthquote = map.get("postquote"+j).length( );
                    System.out.println("Post-Length: " + lengthquote);
                    length = length - lengthquote - 7;
                    System.out.println("Post-Length: " + length);
                    try {
                        Thread.sleep(1000);
                    } catch(InterruptedException ex) {
                        Thread.currentThread().interrupt();
                    }
                }
            }



            //Modify Thanks-Results
            //Format before: The Following 19 Users Say Thank You...
            //Format before: The Following 4 Users Say Thank You to Chainfire For This Useful
Post: [ View ] AproSamurai(15th November 2011), justlovejoy(16th November 2011), moneyover(13th
July 2012), ugothakd(15th November 2011)

            String modthanks = printthanks.text();
            //System.out.println("Is String printthanks empty? :" + text.isEmpty());
            String before = "Following ";
            String after = "User";
            String size = null;

            String thanksglobal;
            if (modthanks.isEmpty()){
                System.out.println("Thanks: " + "0");
                thanksglobal = ("0");
            } else {
                size = modthanks.substring(modthanks.indexOf(before) + before.length(),
modthanks.indexOf(after));
                if (size.length()>1){
                    System.out.println("Thanks: " + size);
                    thanksglobal = size;
                } else {
                    System.out.println("Thanks: " + "1");
                    thanksglobal =("1");
                }
            }

            System.out.println("User: " + printpostersnames.text());

            //Modify Memberinfo
            //Format before: OP Senior Moderator / Senior Recognized Developer - Where is my
shirt? Thanks Meter: 50,487   9,143 posts Join Date:Joined: Oct 2007 Donate to Me More Less
            String modmemberinfo = printmemberinfo.text();
            String cut1 = "Thanks Meter: ";
            String membertype = modmemberinfo.substring((0), modmemberinfo.indexOf(cut1));
            System.out.println("Membertype: " + membertype);

            // Modify Total Thanks, Total Posts and Join Date
            // Thanks Meter: 50,504
            // 9,146 posts
            // Join Date:Joined: Oct 2007

            String modmemberthanks = printmemberthanks0.text();
```

```
            String memberthanks = modmemberthanks.substring(14);
            System.out.println("Member Total Thanks: " + memberthanks);

            String modmemberposts = printmemberposts0.text();
            String cut2 = " posts";
            String memberposts = modmemberposts.substring(0, modmemberposts.indexOf(cut2));
            System.out.println("Member No. of Posts: " + memberposts);

            String modmemberjoin = printmemberjoin0.text();
            String memberjoin = modmemberjoin.substring(18);
            System.out.println("Member Join Date: " + memberjoin);

            //  (modmemberinfo.contains(cut4)){
            // memberjoin = modmemberinfo.substring(modmemberinfo.indexOf(cut3)+ cut3.length(),
modmemberinfo.indexOf(cut4));

            // modify and parse member join date to sql date

            String datestring1 = memberjoin;
            // Mar 2011
            String inputformat1 ="MMM yyyy";
            String format2 ="yyyy-MM-dd";
            //sql: YYYY-MM-DD HH:MM:SS
            SimpleDateFormat sdf2 = new SimpleDateFormat(inputformat1, Locale.ENGLISH);
            SimpleDateFormat sdf3 = new SimpleDateFormat(format2);


            try {
                Date date1 = sdf2.parse(datestring1);
                System.out.println(date1);
                System.out.println(sdf3.format(date1));
                memberdate = (sdf3.format(date1));
            } catch (ParseException ex) {
                ex.printStackTrace();
            }


            String modprintpostcount = printpostcount.text();
            String postcountnew = modprintpostcount.substring(1);
            int postcountint = Integer.parseInt(postcountnew);

            int IDint = Integer.parseInt(ID);

            thanksglobal = thanksglobal.replaceAll("[^\\d.]", "");
            int thanksglobalint = Integer.parseInt(thanksglobal);

            memberthanks = memberthanks.replaceAll("[^\\d.]", "");
            int memberthanksint = Integer.parseInt(memberthanks);

            memberposts = memberposts.replaceAll("[^\\d.]", "");
            int memberpostsint = Integer.parseInt(memberposts);

            int IDx             = Integer.parseInt("0");
            String post_topics = ("");
            int post_pol        = Integer.parseInt("0");
            int post_sub        = Integer.parseInt("0");
            int postquote_pol0  = Integer.parseInt("0");
            int postquote_pol1  = Integer.parseInt("0");
            int postquote_pol2  = Integer.parseInt("0");
            int postquote_pol3  = Integer.parseInt("0");
            int postquote_pol4  = Integer.parseInt("0");

            // if (post_quote_name0 = is Null(){post_quote_name0 = ""
            //Write to mysql db

            ConnectionToMySql (IDx, IDint, postcountint, threadid, ""+sqldate+"",
""+printposts.text(), post_topics, post_pol, post_sub, length, thanksglobalint,
""+printposternames.text(),""+membertype,memberthanksint,memberpostsint,""+memberdate+"",
quoteint, map.get("postquote0"),map.get("postquote_name0")+"", postquote_pol0,
map.get("postquote1"), map.get("postquote_name1")+"", postquote_pol1, map.get("postquote2"),
map.get("postquote_name2")+"", postquote_pol2, map.get("postquote3"),
map.get("postquote_name3")+"", postquote_pol3, map.get("postquote4"),
map.get("postquote_name4")+"", postquote_pol4);
```

```
            // empty hashmap table from quote information
            map.clear();

            System.out.println("-----------------------------------------------");

        }
    }

}
```

B1_getAbsoluteResponse.java

```
package project14;

import java.io.IOException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class B1_getAbsoluteResponse {
        public static DB db = new DB();
        public static int Count;
        public static int maxidint;
        public static String Name;

        public static void main (final String[] args) throws SQLException, IOException {

                String sql = "SELECT MAX(ID) FROM analysis;";
                ResultSet rs = db.runSql(sql);
                while (rs.next()) {
                        String Maxid = rs.getString(1);
                        System.out.println(Maxid);
                        maxidint = Integer.parseInt(Maxid);
                }
                for (int i = 1; i <= maxidint; i++){

                        String sql1 = "SELECT member_name FROM analysis WHERE ID ="+i+";";

                        ResultSet rs1 = db.runSql(sql1);
                        while (rs1.next()) {
                                Name = rs1.getString(1);
                                System.out.println("ID = " + i);
                                System.out.println(Name);
                        }

                        String sql2 = "SELECT COUNT( * ) FROM masterdata WHERE post_quote0 LIKE
('%"+Name+"%') OR post_quote1 LIKE ('%"+Name+"%')OR post_quote2 LIKE ('%"+Name+"%')OR post_quote3
LIKE ('%"+Name+"%')OR post_quote4 LIKE ('%"+Name+"%')";

                        ResultSet rs2 = db.runSql(sql2);
                        while (rs2.next()) {
                                Count = rs2.getInt(1);
                                System.out.println(Count);
                        }


                        String sql3 = "UPDATE `masterthese`.`analysis` SET response = ? " + "WHERE
`analysis`.`ID` = "+i+";";

                        PreparedStatement pst = db.conn.prepareStatement(sql3);
                        pst.setInt (1, Count);
                        pst.execute();
                        pst.close();


                        System.out.println("Inserted Count in response");


                }

        }

}
```

B2_getAbsolutePost_mentions.java

```
package project14;

import java.io.IOException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class B2_getAbsolutePost_mentions{
    public static DB db = new DB();
    public static int Count;
    public static int maxidint;
    public static String Name;

    public static void main (final String[] args) throws SQLException, IOException {

        String sql = "SELECT MAX(ID) FROM analysis;";
        ResultSet rs = db.runSql(sql);
        while (rs.next()) {
            String Maxid = rs.getString(1);
            System.out.println(Maxid);
            maxidint = Integer.parseInt(Maxid);
        }
        for (int i = 1; i <= maxidint; i++){

            String sql1 = "SELECT member_name FROM analysis WHERE ID ="+i+";";

            ResultSet rs1 = db.runSql(sql1);
            while (rs1.next()) {
                Name = rs1.getString(1);
                System.out.println("ID = " + i);
                System.out.println(Name);
            }

            String sql2 = "SELECT COUNT( * ) FROM masterdata WHERE post_quote0 LIKE
('%"+Name+"%') OR post_quote1 LIKE ('%"+Name+"%')OR post_quote2 LIKE ('%"+Name+"%')OR post_quote3
LIKE ('%"+Name+"%')OR post_quote4 LIKE ('%"+Name+"%')";

            ResultSet rs2 = db.runSql(sql2);
            while (rs2.next()) {
                Count = rs2.getInt(1);
                System.out.println(Count);
            }


            String sql3 = "UPDATE `masterthese`.`analysis` SET response = ? " + "WHERE
`analysis`.`ID` = "+i+";";

            PreparedStatement pst = db.conn.prepareStatement(sql3);
            pst.setInt (1, Count);
            pst.execute();
            pst.close();


            System.out.println("Inserted Count in response");
        }

    }

}
```

B3_getAbsoluteNegFeedback.java

```
package project14;

import java.io.IOException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class B3_getAbsoluteNegFeedback {
    public static DB db = new DB();
    public static int Count;
    public static int maxidint;
```

```
    public static String Name;

    public static void main (final String[] args) throws SQLException, IOException {

        String sql = "SELECT MAX(ID) FROM analysis;";
        ResultSet rs = db.runSql(sql);
        while (rs.next()) {
            String Maxid = rs.getString(1);
            System.out.println(Maxid);
            maxidint = Integer.parseInt(Maxid);
        }
        for (int i = 1; i <= maxidint; i++){

            String sql1 = "SELECT member_name FROM analysis WHERE ID ="+i+";";

            ResultSet rs1 = db.runSql(sql1);
            while (rs1.next()) {
                Name = rs1.getString(1);
                System.out.println("ID = " + i);
                System.out.println(Name);
            }


            String sql2 ="SELECT COUNT( * ) FROM masterdata WHERE post_post LIKE ('%"+Name+"%')
OR post_quote0 LIKE ('%"+Name+"%') OR post_quote1 LIKE ('%"+Name+"%') OR post_quote2 LIKE
('%"+Name+"%') OR post_quote3 LIKE ('%"+Name+"%') OR post_quote4 LIKE ('%"+Name+"%')  AND
post_pol < '0' ";


            ResultSet rs2 = db.runSql(sql2);
            while (rs2.next()) {
                Count = rs2.getInt(1);
                System.out.println(Count);
            }


            String sql3 = "UPDATE `masterthese`.`analysis` SET feedback_neg = ? " + "WHERE
`analysis`.`ID` = "+i+";";

            PreparedStatement pst = db.conn.prepareStatement(sql3);
            pst.setInt (1, Count);
            pst.execute();
            pst.close();


            System.out.println("Inserted Count in feedback_neg");
        }
    }

}
```

C1_getRelativeOpinionLeader

```
package project14;

import java.io.IOException;
import java.math.BigDecimal;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class C1_getRelativeOpinionLeader {
    public static DB db = new DB();

    // insert here the resolution of scale
    public static int scale = 1000;
    public static BigDecimal ScaleDec = new BigDecimal(scale);
    // variable for ID count
    public static int maxidint;

    // variables for Frequency
    public static int MaxFreq;
    public static int AbsFreq;
```

```
public static BigDecimal MultiFreqDec;
public static int RelFreq;

// variables for Response
public static int MaxRes;
public static int AbsRes;
public static BigDecimal MultiResDec;
public static int RelRes;

// variables for post mentions
public static int MaxMent;
public static int AbsMent;
public static BigDecimal MultiMentDec;
public static int RelMent;

// variables for feedback pos
public static int MaxPos;
public static int AbsPos;
public static BigDecimal MultiPosDec;
public static int RelPos;

// variables for feedback neg
public static int MaxNeg;
public static int AbsNeg;
public static BigDecimal MultiNegDec;
public static int RelNeg;

// variables for average message size
public static int MaxAv;
public static int AbsAv;
public static BigDecimal MultiAvDec;
public static int RelAv;

// variables for involvement
public static int MaxInv;
public static int AbsInv;
public static BigDecimal MultiInvDec;
public static int RelInv;

public static void main (final String[] args) throws SQLException, IOException {
    //get highest ID
    String sql = "SELECT MAX(ID) FROM analysis;";
    ResultSet rs = db.runSql(sql);
    while (rs.next()) {
        String Maxid = rs.getString(1);
        System.out.println(Maxid);
        maxidint = Integer.parseInt(Maxid);


        //Frequency
        String sql0 = "SELECT MAX(frequency)FROM analysis;";
        ResultSet rs0 = db.runSql(sql0);
        while (rs0.next()) {
            MaxFreq = rs0.getInt(1);
            System.out.println("Frequency Max: "+MaxFreq);
        }

        BigDecimal MaxFreqDec = new BigDecimal(MaxFreq);
        // 100 : highest frequency
        MultiFreqDec = ScaleDec.divide(MaxFreqDec,2, BigDecimal.ROUND_HALF_UP);
        System.out.println("Frequency Multiplier: "+MultiFreqDec);


        //Response
        String sqlres = "SELECT MAX(response)FROM analysis;";
        ResultSet rsres = db.runSql(sqlres);
        while (rsres.next()) {
            MaxRes = rsres.getInt(1);
            System.out.println("Response Max: "+MaxRes);
        }

        BigDecimal MaxResDec = new BigDecimal(MaxRes);
        // 100 : highest frequency
        MultiResDec = ScaleDec.divide(MaxResDec,4, BigDecimal.ROUND_HALF_UP);
```

```
        System.out.println("Response Multiplier: "+MultiResDec);


        //post mentions
        String sqlment = "SELECT MAX(post_mentions)FROM analysis;";
        ResultSet rsment = db.runSql(sqlment);
        while (rsment.next()) {
            MaxMent = rsment.getInt(1);
            System.out.println("Post mentions Max: "+MaxMent);
        }

        BigDecimal MaxMentDec = new BigDecimal(MaxMent);
        // 100 : highest frequency
        MultiMentDec = ScaleDec.divide(MaxMentDec,2, BigDecimal.ROUND_HALF_UP);
        System.out.println("Post mentions Multiplier: "+MultiMentDec);


        //feedback_pos
        String sqlpos = "SELECT MAX(feedback_pos_rel)FROM analysis;";
        ResultSet rspos = db.runSql(sqlpos);
        while (rspos.next()) {
            MaxPos = rspos.getInt(1);
            System.out.println("Feedback Positive Max: "+MaxPos);
        }

        BigDecimal MaxPosDec = new BigDecimal(MaxPos);
        // 100 : highest frequency
        MultiPosDec = ScaleDec.divide(MaxPosDec,2, BigDecimal.ROUND_HALF_UP);
        System.out.println("Feedback Pos Multiplier: "+MultiPosDec);


        //feedback neg
        String sqlneg = "SELECT MAX(feedback_neg_rel)FROM analysis;";
        ResultSet rsneg = db.runSql(sqlneg);
        while (rsneg.next()) {
            MaxNeg = rsneg.getInt(1);
            System.out.println("Feedback Negative Max: "+MaxNeg);
        }

        BigDecimal MaxNegDec = new BigDecimal(MaxNeg);
        // 100 : highest frequency
        MultiNegDec = ScaleDec.divide(MaxNegDec,2, BigDecimal.ROUND_HALF_UP);
        System.out.println("Feedback Neg Multiplier: "+MultiNegDec);


        //Avmessagesize
        String sqlav = "SELECT MAX(avmessagesize)FROM analysis;";
        ResultSet rsav = db.runSql(sqlav);
        while (rsav.next()) {
            MaxAv = rsav.getInt(1);
            System.out.println("Average Message Size Max: "+MaxAv);
        }

        BigDecimal MaxAvDec = new BigDecimal(MaxAv);
        // 100 : highest frequency
        MultiAvDec = ScaleDec.divide(MaxAvDec,4, BigDecimal.ROUND_HALF_UP);
        System.out.println("Average Message Size Multiplier: "+MultiAvDec);

        //Involvement
        String sqlinv = "SELECT MAX(involvement)FROM analysis;";
        ResultSet rsinv = db.runSql(sqlinv);
        while (rsinv.next()) {
            MaxInv = rsinv.getInt(1);
            System.out.println("Involvement Max: "+MaxInv);
        }

        BigDecimal MaxInvDec = new BigDecimal(MaxInv);
        // 100 : highest frequency
        MultiInvDec = ScaleDec.divide(MaxInvDec,4, BigDecimal.ROUND_HALF_UP);
        System.out.println("Involvement Multiplier: "+MultiInvDec);

    }
    for (int i = 1; i <= maxidint; i++){
```

```
//Frequency
String sql1 = "SELECT frequency FROM analysis WHERE ID ="+i+";";
ResultSet rs1 = db.runSql(sql1);
while (rs1.next()) {
    AbsFreq = rs1.getInt(1);
    System.out.println("Frequecy Absolute = " + AbsFreq);
}
BigDecimal AbsFreqDec = new BigDecimal(AbsFreq);
BigDecimal RelFreqDec = MultiFreqDec.multiply(AbsFreqDec);

System.out.println("Frequency Relative: "+RelFreqDec);
BigDecimal RelFreqDecScaled = RelFreqDec.setScale(0, BigDecimal.ROUND_HALF_UP);
RelFreq = RelFreqDecScaled.intValue();
System.out.println("Frequency Relative Int: "+RelFreq);


//Response
String sqlres1 = "SELECT response FROM analysis WHERE ID ="+i+";";
ResultSet rsres1 = db.runSql(sqlres1);
while (rsres1.next()) {
    AbsRes = rsres1.getInt(1);
    System.out.println("Response Absolute = " + AbsRes);
}
BigDecimal AbsResDec = new BigDecimal(AbsRes);
BigDecimal RelResDec = MultiResDec.multiply(AbsResDec);

System.out.println("Response Relative: "+RelResDec);
BigDecimal RelResDecScaled = RelResDec.setScale(0, BigDecimal.ROUND_HALF_UP);
RelRes = RelResDecScaled.intValue();
System.out.println("Response Relative Int: "+RelRes);



//Post mentions
String sqlment1 = "SELECT post_mentions FROM analysis WHERE ID ="+i+";";
ResultSet rsment1 = db.runSql(sqlment1);
while (rsment1.next()) {
    AbsMent = rsment1.getInt(1);
    System.out.println("Post Mentions Absolute = " + AbsMent);
}
BigDecimal AbsMentDec = new BigDecimal(AbsMent);
BigDecimal RelMentDec = MultiMentDec.multiply(AbsMentDec);

System.out.println("Post Mentions Relative: "+RelMentDec);
BigDecimal RelMentDecScaled = RelMentDec.setScale(0, BigDecimal.ROUND_HALF_UP);
RelMent = RelMentDecScaled.intValue();
System.out.println("Post Mentions Relative Int: "+RelMent);



//feedback pos
String sqlpos1 = "SELECT feedback_pos_rel FROM analysis WHERE ID ="+i+";";
ResultSet rspos1 = db.runSql(sqlpos1);
while (rspos1.next()) {
    AbsPos = rspos1.getInt(1);
    System.out.println("Feedback Pos Absolute = " + AbsPos);
}
BigDecimal AbsPosDec = new BigDecimal(AbsPos);
BigDecimal RelPosDec = MultiPosDec.multiply(AbsPosDec);

System.out.println("Feedback Pos Relative: "+RelPosDec);
BigDecimal RelPosDecScaled = RelPosDec.setScale(0, BigDecimal.ROUND_HALF_UP);
RelPos = RelPosDecScaled.intValue();
System.out.println("Feedback Pos Relative Int: "+RelPos);


//feedback pos
String sqlneg1 = "SELECT feedback_neg_rel FROM analysis WHERE ID ="+i+";";
ResultSet rsneg1 = db.runSql(sqlneg1);
while (rsneg1.next()) {
    AbsNeg = rsneg1.getInt(1);
    System.out.println("Feedback Neg Absolute = " + AbsNeg);
}
BigDecimal AbsNegDec = new BigDecimal(AbsNeg);
```

```
            BigDecimal RelNegDec = MultiNegDec.multiply(AbsNegDec);

            System.out.println("Feedback Neg Relative: "+RelNegDec);
            BigDecimal RelNegDecScaled = RelNegDec.setScale(0, BigDecimal.ROUND_HALF_UP);
            RelNeg = RelNegDecScaled.intValue();
            System.out.println("Feedback Neg Relative Int: "+RelNeg);

            //Avmessagesize
            String sqlav1 = "SELECT avmessagesize FROM analysis WHERE ID ="+i+";";
            ResultSet rsav1 = db.runSql(sqlav1);
            while (rsav1.next()) {
                AbsAv = rsav1.getInt(1);
                System.out.println("Average Message Size Absolute = " + AbsAv);
            }
            BigDecimal AbsAvDec = new BigDecimal(AbsAv);
            BigDecimal RelAvDec = MultiAvDec.multiply(AbsAvDec);

            System.out.println("Average Message Size Relative: "+RelAvDec);
            BigDecimal RelAvDecScaled = RelAvDec.setScale(0, BigDecimal.ROUND_HALF_UP);
            RelAv = RelAvDecScaled.intValue();
            System.out.println("Average Message Size Relative Int: "+RelAv);



            //Involvement
            String sqlinv1 = "SELECT involvement FROM analysis WHERE ID ="+i+";";
            ResultSet rsinv1 = db.runSql(sqlinv1);
            while (rsinv1.next()) {
                AbsInv = rsinv1.getInt(1);
                System.out.println("Involvement Absolute = " + AbsInv);
            }
            BigDecimal AbsInvDec = new BigDecimal(AbsInv);
            BigDecimal RelInvDec = MultiInvDec.multiply(AbsInvDec);

            System.out.println("Involvement Relative: "+RelInvDec);
            BigDecimal RelInvDecScaled = RelInvDec.setScale(0, BigDecimal.ROUND_HALF_UP);
            RelInv = RelInvDecScaled.intValue();
            System.out.println("Involvement Relative Int: "+RelInv);


            String sql3     =   "UPDATE `masterthese`.`opinionleader_rel` SET frequency = ?,
response = ?, post_mentions = ?, feedback_pos = ?, feedback_neg = ?, avmessagesize = ?,
involvement = ? " + "WHERE `ID` = "+i+";";


            PreparedStatement pst = db.conn.prepareStatement(sql3);
            pst.setInt (1, RelFreq);
            pst.setInt (2, RelRes);
            pst.setInt (3, RelMent);
            pst.setInt (4, RelPos);
            pst.setInt (5, RelNeg);
            pst.setInt (6, RelAv);
            pst.setInt (7, RelInv);
            pst.execute();
            pst.close();


            System.out.println("------------------------------");
        }
    }
}
```

D1_getTextminingTable.java

```
package project14;

import java.io.IOException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class D1_getTextminingTable {
```

```
    public static DB db = new DB();

    public static String Name;

    public static void main (final String[] args) throws SQLException, IOException {


        for (int i = 1; i <= 59;){
            for (int j = 2011; j <=2016; j++){
                for (int k = 1; k <=12; k++){

                    String sql1 = "SELECT GROUP_CONCAT(DISTINCT post_topics SEPARATOR ' ') FROM
masterdata WHERE YEAR(post_timestamp) = "+j+" AND MONTH(post_timestamp) = "+k+";";


                    ResultSet rs1 = db.runSql(sql1);
                    while (rs1.next()) {
                        Name = rs1.getString(1);
                        System.out.println("ID = " + i);
                        System.out.println(Name);
                        Name =  " '' "  + Name + " '' ";

                        String sql0 = "SET group_concat_max_len = 18446744073709551615";
                        String sql2     = "INSERT INTO masterthese.textmining (int_id, period,
words) VALUES ('"+i+"','"+j+'-'+k+"', ?);" ;

                        PreparedStatement pst1 = db.conn.prepareStatement(sql0);
                        pst1.execute();
                        pst1.close();
                        PreparedStatement pst = db.conn.prepareStatement(sql2);
                        pst.setString (1, Name);
                        pst.execute();
                        pst.close();


                        System.out.println("Inserted Row");

                        i=i+1;

                    }
                }
            }
        }
    }
}

package project14;

import java.io.IOException;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;


public class datetime
public class datetime {

    public static void main (final String[] args) throws SQLException, IOException {

        String value = "20th November 2011, 08:06 PM";
        value = value.replaceAll("([0-9]{1,2})(st|nd|rd|th)(.*)", "$1$3");
        System.out.println(value);
        String format ="dd MMMM yyyy, hh:mm a";
        String format1 ="yyyy-MM-dd' 'HH:mm:ss";
        //sql: YYYY-MM-DD HH:MM:SS
        SimpleDateFormat sdf = new SimpleDateFormat(format);
        SimpleDateFormat sdf1 = new SimpleDateFormat(format1);
        try {
            Date date = sdf.parse(value);
            System.out.println(date);
```

```
            //System.out.println(sdf1.format(date));
            String sqldate = (sdf1.format(date));
            System.out.println(sqldate);

        } catch (ParseException ex) {
            ex.printStackTrace();
        }

    }

}
```

Connection to database

```
package project14;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class DB {

    public Connection conn = null;

    public DB() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            String url = "jdbc:mysql://127.0.0.1:3306/masterthese";
            // 144.76.19.105
            conn = DriverManager.getConnection(url, "root", "CBS");
            System.out.println("conn built");
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }


    public ResultSet runSql(String sql) throws SQLException {
        Statement sta = conn.createStatement();
        return sta.executeQuery(sql);
    }

    public boolean runSql2(String sql) throws SQLException {
        Statement sta = conn.createStatement();
        return sta.execute(sql);
    }

    @Override
    protected void finalize() throws Throwable {
        if (conn != null || !conn.isClosed()) {
            conn.close();
        }
    }

}
```

4 Python Code

nounphraster.py

```
#!/usr/bin/env python
#from   future   import print function
import mysql.connector

from textblob import TextBlob
```

```
from textblob.utils import strip_punc
import time
from nltk.tokenize import TabTokenizer
from textblob.taggers import NLTKTagger


cnx = mysql.connector.connect(user='root', password='CBS',
                              host='localhost',
                              database='masterthese')
msg = "connected!"
print msg

cursor = cnx.cursor()

for i in range(1,500):
    query = ("SELECT post_post FROM masterdata where IDx like %s ;")
    IDx = i
    cursor.execute(query, (IDx, ))

    row = cursor.fetchone()
    text = str(row)
    text = text.replace("bytearray(b'", "")

##### NLTK Tokinzer #####
    from nltk.tokenize import *
    tok = Token(TEXT=text)
    WhitespaceTokenizer().tokenize(tok)
    print (tok['SUBTOKENS'])
    result1 = (tok['SUBTOKENS'])

    ##### TextBlob API #####
    noun_phrases = set(TextBlob(text).noun_phrases)
    # Strip punctuation from ends of noun phrases and exclude long phrases
    result = [strip_punc(np) for np in noun_phrases if len(np.split()) <= 10]

##### TextBlob API / TABTokenizer #####
    #tokenizer = TabTokenizer()
    #blob = TextBlob(text, tokenizer=tokenizer)
    #result = blob.tokens
    #print result

##### TextBlob API / NLTK Tagger #####
    #nltk tagger = NLTKTagger()
    #blob = TextBlob(text, pos tagger=nltk tagger)
    #result = blob.pos tags

    print "--"
    print i
    print result
    print "----------------------------"

    # take here result for textblob or result1 for NLTK Tokenizer
    out = str(result)
    cursor.execute ("""
            UPDATE masterdata
            SET post topics=%s
            WHERE IDx=%s
            """, (out, i))
    # commit Data to the database
    cnx.commit()
    time.sleep(.100)

cursor.close()
cnx.close()
```

Sentiment_analysis.py

```
import mysql.connector
import time
from pattern.en import sentiment, polarity, subjectivity, positive, wordnet, ADJECTIVE
```

```
cnx = mysql.connector.connect(user='root', password='CBS',
                              host='localhost',
                              database='masterthese')
msg = "connected!"
print msg

cursor = cnx.cursor()


for i in range(6571,101655):

    #Do Sentiment Analysis for Post

    cursor.execute("""
                   SELECT post post
                   FROM masterdata
                   WHERE IDx like %s
                   """,
                   (i, )
                   )
    row = cursor.fetchone()
    print(row)

    #sentiment analysis

    text = str(row)
    text = text.replace("bytearray(b'", "")
    print text
    senti_tupel = sentiment(text)
    print senti_tupel

    add_pol = senti_tupel[0]
    add_sub = senti_tupel[1]

    print add_pol
    print add_sub
    print i

    add_pol = float(add_pol)
    add_sub = float(add_sub)

    cursor.execute ("""
                    UPDATE masterdata
                    SET post pol=%s, post sub=%s
                    WHERE IDx=%s
                    """,
                    (add_pol, add_sub, i))

    #Do Sentiment Analysis for Quote 0 to 4 if filled.

    query = ("SELECT post_quotes FROM masterdata where IDx like %s ;")
    IDx = i
    cursor.execute(query, (IDx, ))
    row = cursor.fetchone()

    j = row[0]

    if j >= 1:
        print j
        for k in range(0,j):

            # define variable collumn
            t   = str(k)
            col =  'post_quote' + t
            print col

            IDx = i


            query1 = "SELECT %s FROM masterdata where IDx like %s ;" %(col,IDx, )

            cursor.execute(query1)
```

```
            row1 = cursor.fetchone()


            #sentiment analysis

            text = str(row1)
            text = text.replace("bytearray(b'", "")
            print text
            senti_tupel = sentiment(text)
            print senti_tupel

            add_pol = senti_tupel[0]
            #add_sub = senti_tupel[1]

            print add_pol
            #print add sub
            print i

            add_pol = float(add_pol)
            #add sub = float(add sub)

            #define varaible collumn
            col1 =  'post_quote_pol' + t
            print col1

            query2 = "UPDATE masterdata SET %s=%s WHERE IDx=%s ;" %(col1, add_pol, i)

            cursor.execute (query2)



    # commit Data to the database
    cnx.commit()
    time.sleep(.100)

cursor.close()
cnx.close()
```