universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Exploring External Links in Twitter“

verfasst von / submitted by

Simon Brändle, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Diplom-Ingenieur (Dipl.-Ing.)

Wien, 2018 / Vienna 2018

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 935
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Medieninformatik
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Ing. Dr. Wolfgang Klas
Mitbetreut von / Co-Supervisor:	Mag. Dr. Maia Zaharieva

Declaration of Authorship

I hereby declare that i have written this thesis myself, independently and without the aid of unfair or unauthorized resources. Whenever content has been taken directly or indirectly from other sources, this has been indicated and the source referenced. This thesis has not been previously presented as an examination paper in this or any other form in Austria or abroad.

Date, Signature

Abstract

In recent years, since the abundance of published information in social media platforms, research regarding the examination of credibility is becoming increasingly important. Users gather information on such platforms and take part in distribution of content. The body of an information has many facets such as facts, statements, opinions, and sentiment in combination with various types of media. Generally, platforms lack verification of credentials and allow anonymous content to be posted. While anonymity is important in the matter of freedom of speech these information artifacts can be misused to deliberately spread false content. Consequently, for the the average user it is becoming more difficult to distinguish between false and accurate information. Therefore, the aim of this thesis is to develop a prototype, which helps to increase the awareness of inaccurate information. In a first step, we conducted a user study to reveal credibility perception of Twitter posts. Based on the study we identified features to develop automatic credibility classification approaches. To demonstrate the approaches we propose a prototype that allows searching for Twitter posts and filter regarding credibility, sentiment and media type. The proposed prototype enhances the information search process and adds visual cues for calculated sentiment and credibility scores. The evaluation of the prototype in terms of usability suggests a positive user acceptance within the participating user group.

Contents

Chapter 1

Introduction

1.1 Motivation

Social media platforms enable users to create and share content of various media types. Popular social media platforms are social network sites (SNS), where participating users can set up a profile, befriend other users and/or follow other users based on personal preferences. Users can post content such as messages, images and videos, and share opinions and other content depending on the focus of the social network [1]. Moreover, users can browse SNS and gather information on topics and events among others. However, not all content published on SNS is true and trust-able as most SNSs allow anyone to register without verification of credentials.

The motivation for this thesis is to enhance the process of finding messages and attached media-content on SNS by providing approaches to filter for special criterias such as credibility and sentiment. For example, messages on SNS can consist of opinions, facts, and statements or provide comments to images, videos, external links, etc. The sentiment of the messages can be

positive, negative, or neutral or any combinations of them. For the average user it can be difficult to find what he or she is looking for. One might want to find only positive messages during breaking news events and others may want to find negative messages, others may just want to find neutral information and facts. However, content can be published by anyone with various intentions behind it. On the one hand, anonymity is great for people in countries that lack the freedom of speech [2]. People can create critical content, leave feedback, or participate in online protests without the fear of negative effects and punishments [2]. On the other hand, anonymous accounts can be used to spread false rumors [3] or false accusations [4] that can jeopardise the well-being of citizens which follow the stream of news in social media [3]. Spammers or untrustworthy users can generate messages to lure users on malicious websites or to drive targeted spam campaigns [5]. Abusers can create multiple accounts to fake reviews for products [2] or create automatic programs that generate messages to produce a positive or negative sentiment towards political candidates [6].

1.2 Aim and Objectives

The aim of this thesis is to develop a prototype that enables exploring messages and attached media objects on Twitter. The prototype should enable users to find content under the consideration of two crucial aspects: credibility and sentiment. This work will address the issues of credibility in social networks and describe approaches to automatically classify content as credible or not. For this purpose, we will conduct a user study to acquire knowledge about decisions made by users whether or not a message is credible. We then will categorize and analyze the gained knowledge in order

to find possible features to create an approach that automatically classifies credibility. Additionally, this thesis will investigate the possibilities for an automated sentiment analysis and classification. The gained insights from the performed analyses will build the basis for the development of the final prototype. Finally, a user study evaluates the usability of the prototype.

1.3 Structure of the Work

Chapter 2 defines relevant background information on core topics discussed in this thesis. We provide an overview of natural language processing and describe fundamental techniques. This includes the idea of sentiment analysis, the challenges and use cases. Lastly we provide an overview of the term credibility in social media and highlight potential challenges.

Chapter 3 highlights state of the art research related to sentiment analysis, user credibility and social media stream visualization. This chapter outlines approaches for sentiment analysis to classify the sentiment of Twitter messages among others. We highlight projects in the space of credibility that cover different domains to point out the various aspects of credibility in social media. Lastly we highlight social media content visualization possibilities with regard to sentiment analysis and credibility.

Chapter 4 describes the methodology of this thesis and provides an overview of the research design.

Chapter 5 includes the initial user study that aims to identify why users decide whether or not a tweet is credible. We highlight the study design and provide results of an evaluation against a state of the art credibility project. We provide details on user feedback categorization and define feature sets.

Based on experiments with a publicly available data set, we select promising feature sets to create credibility models.

Chapter 6 highlights the design and implementation of a prototype that enables sentiment and credibility based filtering for Twitter messages. We outline the conceptual design and highlight implementation details. Furthermore we conduct a prototype evaluation and present the results.

Chapter 7 finalizes this thesis with a conclusion and includes a summary of the proposed work. Additionally this chapter provides ideas for future work.

Chapter 2

Background

This chapter defines relevant background information on core topics discussed in this thesis. Section 2.1 covers an overview of natural language processing and describes fundamental techniques. Section 2.2 describes the idea of sentiment analysis, its challenges, and use cases. Section 2.3 gives an overview of the term credibility and its challenges in social media.

2.1 Natural Language Processing

Natural language processing (NLP) systems are computer systems that analyze human languages. The goal of this research field is to perform useful tasks on human language [7], for example machine translation, spell checking, and speech recognition. The input is not limited to text but it can be spoken language or keyboard input [8]. Our work integrates and works with textual data from the social media platform Twitter. Therefore, the techniques covered in the following subsections focus on NLP systems that process text input.

2.1.1 Text preprocessing

An NLP system employs different methods to segment the input text. This preprocessing task is an essential part in NLP systems to create units for further processing. Figure 2.1 shows a general NLP system that handles text input and employs different methods that are covered in the next following sections.

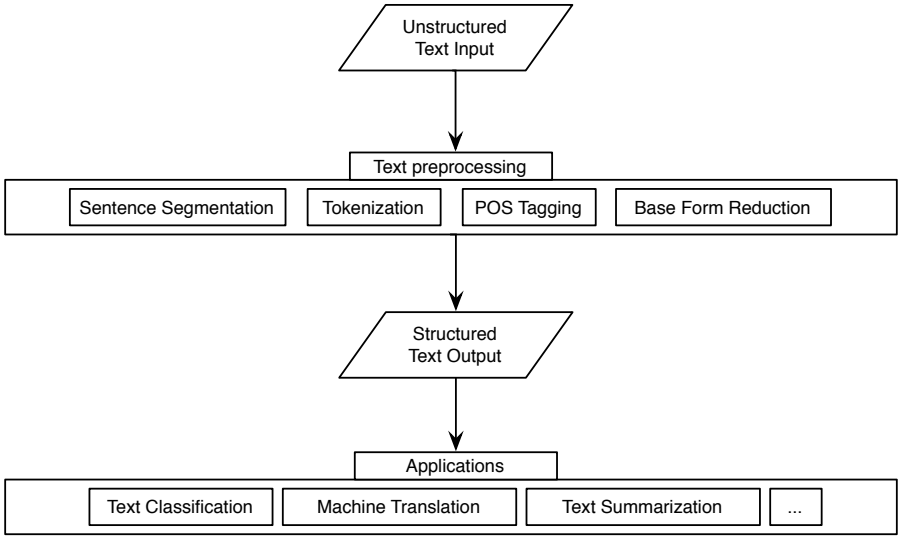

Figure 2.1: Schema of an NLP system that handles text input.

Sentence Segmentation has the goal to find the sentence boundaries of an input text. Finding the end of a sentence is not always trivial. For example, in English a dot can define the end of a sentence or the end of an abbreviation.

Tokenization is another commonly employed technique for text preprocessing. The task is to divide the input text into tokens. A token is a well-defined linguistically meaningful unit such as a word, a number, and a

punctuation [9] [10]. For example, to tokenize the input "Simon's surfboard was stolen in Los Angeles. E-Mail any information to simon@gmail.com." a tokenizer needs to employ methods to handle E-Mail addresses and city names correctly in order to generate a set of tokens. Tokens provide a basis for further processing of the text input. For example, if the tokenizer creates a single token "Los Angeles" instead of "Los" and "Angeles" the following processing methods can identify and tag the word as a noun or city.

Part-of-speech (POS) Tagging aims to label each word of a sentence with its respective parts of speech tag (noun, verb, adjective, preposition, etc.) A POS tag provides significant information about the word and its neighbors. One commonly used tag set for English is the 48-tag Penn Treebank tag set (see Figure 2.2) [11].

The Penn Treebank POS tagset.

1. CC	Coordinating conjunction	25. TO	*to*
2. CD	Cardinal number	26. UH	Interjection
3. DT	Determiner	27. VB	Verb, base form
4. EX	Existential *there*	28. VBD	Verb, past tense
5. FW	Foreign word	29. VBG	Verb, gerund/present participle
6. IN	Preposition/subordinating conjunction	30. VBN	Verb, past participle
7. JJ	Adjective	31. VBP	Verb, non-3rd ps. sing. present
8. JJR	Adjective, comparative	32. VBZ	Verb, 3rd ps. sing. present
9. JJS	Adjective, superlative	33. WDT	*wh*-determiner
10. LS	List item marker	34. WP	*wh*-pronoun
11. MD	Modal	35. WP$	Possessive *wh*-pronoun
12. NN	Noun, singular or mass	36. WRB	*wh*-adverb
13. NNS	Noun, plural	37. #	Pound sign
14. NNP	Proper noun, singular	38. $	Dollar sign
15. NNPS	Proper noun, plural	39. .	Sentence-final punctuation
16. PDT	Predeterminer	40. ,	Comma
17. POS	Possessive ending	41. :	Colon, semi-colon
18. PRP	Personal pronoun	42. (	Left bracket character
19. PP$	Possessive pronoun	43.)	Right bracket character
20. RB	Adverb	44. "	Straight double quote
21. RBR	Adverb, comparative	45. ‘	Left open single quote
22. RBS	Adverb, superlative	46. “	Left open double quote
23. RP	Particle	47. ’	Right close single quote
24. SYM	Symbol (mathematical or scientific)	48. ”	Right close double quote

Figure 2.2: 48-tag Penn Treebank tag set [11].

In a flat output text file, tags are often placed after a word following a slash but these representations can vary. Next is an example of a POS tagger

output that employs the Penn Treebank tag set [11]:

Alan Turing: "I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted."

POS tagged: "I/PRP believe/VBP that/IN at/IN the/DT end/NN of/IN the/DT century/NN the/DT use/NN of/IN words/NNS and/CC general/JJ educated/JJ opinion/NN will/MD have/VB altered/VBN so/RB much/JJ that/IN one/CD will/MD be/VB able/JJ to/TO speak/VB of/IN machines/NNS thinking/VBG without/IN expecting/VBG to/TO be/VB contradicted/VBN ./."

POS tagging faces two major challenges, namely ambiguous and unknown words. There exist some words for which more than one POS tag are possible or which are unknown [10]. A tagger has to implement techniques to handle such types of words and to guess the best fitting tag for them (e.g. by taking into account the context by including tags surrounding the unknown/ambiguous words). POS tagging achieves high accuracies (around 96%–97%) for most Indo-European languages (English, French, etc.) [10]. Tagged sentences can be used in further processing steps or in different applications. For example, an information retrieval system can use tagged sentences to filter out nouns or other use case specific words from a document [7]. Tagged sentences can provide the contextual information for a lemmatizer to choose the appropriate lemma for a word or can be used to employ unsupervised sentiment analysis [12].

Base Form Reduction includes methods to transform a word to its common stem or base form. For example, documents can contain different forms

of a word, such as *organize*, *organizes*, and *organizing*. A transformation to its common stem or base form reduces the token size of a document. In the previous example, depending on the employed method, the all three different tokens are reduced to one token, namely *organize*. Two types of word transformations are stemming and lemmatization.

Stemming is the decomposition of a word to a stem based on different simple but effective heuristics [13]. A widely used stemming algorithm for English is the Porter Stemmer, which does suffix stripping of words [14]. The algorithm implements predefined rule groups to reduce a word to a stem. It consists of different steps of sequentially applied word reductions [13]. For example, one step includes the removal of plurals and "-ed" or "-ing" suffixes, another step turns terminal "-y" to i, and another step maps double suffixes such as "-ization", "-ational", etc. to single ones. The output of the algorithm is the produced stem of the word after all steps have finished.

Another commonly employed technique to reduce a word to its stem is lemmatization. Lemmatization usually refers to a more sophisticated word transformation. It tries to identify the base or the dictionary form of a word, which is known as the lemma. A lemmatizer employs vocabularies (e.g. WordNet [15]) and implements morphological analysis of words to return the lemma in consideration of the context [10]. For example, for the word *saw*, the returned lemma is either *see* or *saw* depending on whether the usage of the word was as a verb or a noun [13]. The following shows an example of an input text with a stemmed and lematized output:

Alan Turing: "I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted."

Porter Stemmer: "I believ that at the end of the centuri the use of word and gener educ opinion will have alter so much that one will be abl to speak of machin think without expect to be contradict ."

Wordnet Lemmatizer: "I believe that at the end of the century the use of word and general educate opinion will have alter so much that one will be able to speak of machine think without expect to be contradict ."

2.1.2 Challenges

The complexity of natural languages is one of the core challenges for NLP systems. Ambiguity is one of its central characteristics. For example, words in sentences can have different meanings based on context. The word *duck* is syntactically ambiguous in terms of parts of speech, as it can be a noun (the animal) or a verb (to lower suddenly) [8]. An NLP system needs to determine parts of the structure of a text and enough information in order to answer the question "Who did what to whom?" [9]. For example, a human reader with knowledge in the English language interprets the sentence as "time flies like an arrow" as the time passes like an arrow, in particular quickly. However, an NLP system that parses the sentence can generate an unexpected number of alternatives. One example is "time flies" a legitimate reading of some kinds of insects being fond of an arrow [16]. A practical NLP system must be good at making disambiguation decisions of word sense, word category, syntactic structure, and semantic scope [9]. NLP systems can employ two approaches: (1) a symbolic approach that consists of a set of rules or (2) a statistical approach that employs machine learning algorithms. Symbolic NLP systems are time consuming to build and do not scale well. For example, adding more rule sets to cover obscure sentence constructs can increase the number of undesired parses for common sentences and vice

versa [9]. A statistical NLP system employs automatic learning strategies to learn lexical and structural preferences from corpora of text. The created statistical models are robust and generalize well and reduce the human effort in producing NLP systems [9]. Statistical NLP systems provide successful disambiguation in large scale systems using naturally occurring text [9].

2.1.3 Applications

An NLP system can be found in different applications and tasks such as (i) summarization of text documents, (ii) machine translation of one human language to another, (iii) speech processing to produce text from spoken text, (iv) spelling and grammar checking, and (v) text classification. Text classification tasks in NLP systems can vary from language identification to author classification [9]. A specialized form of classification is sentiment analysis, where a text is classified in two or more classes based on the polarity of the text [10]. A more detailed background on sentiment analysis is provided in the next Section 2.2.

2.2 Sentiment Analysis

Sentiment analysis, also called *opinion mining*, is a popular field of study analyzing the polarity of text. For example, the sentence "I love my new surf board" has a positive polarity while "I hate my new surf board" has a negative polarity. The distinction is not always as clear as in the provided examples and a single document can contain text with varying sentiments. Sentiment analysis can be used to analyzes people's opinion, sentiments, evaluations, attitudes, and emotions towards entities such as topics and products within

documents. The field of potential applications is broad and with the huge volume of recorded opinionated data from social media and the web it became a very active research field since the year 2000 [17].

2.2.1 Classification Types

Sentiment analysis can be considered as a traditional document text classification problem. Depending on the availability of pre-labeled ground truth data, three approaches can be applied to automatically classify a document: supervised, unsupervised, and semi-supervised.

Supervised classification relies on an existing set of labeled data that is used to create a model. This model gets employed to classify and label an unseen document using any supervised learning method, e.g., Naive Bayes [18], support vector machines (SVM) [19], and maximum entropy (MaxEnt) [20].

In an *unsupervised classification* scenario no pre-labeled data is available. In general, unsupervised learning aims at finding "interesting structures" in data samples and makes assumptions of that structures based on different estimation and classification methods [21]. A number of unsupervised approaches in terms of sentiment analysis create or use existing dictionaries/lexicons of words and their respective polarity values. To determine the sentiment of a text unit, such approaches employ predefined functions based on the positive or negative indicators and values defined in the lexicon. Such dictionaries/lexicons include but are not limited to the emotional dictionary from the Linguistic Inquiry and Word Count (LIWC) [22], SentiWordNet [23], Affective norms for English words (ANEW) [24], and AFINN [25]. Other unsupervised approaches such as the one by Turney [12] use an algorithm to find the semantic orientation of a phrase by comparing its similarity to a

negative reference word (e.g. "poor") and a positive reference word (e.g. "excellent") with the help of the AltaVista search engine.

Semi-supervised classification uses unlabeled and labeled data for classification. In many applications, unlabeled data can be obtained more easily than labeled data [26]. Semi-supervised classification algorithms use existing labeled data to to label unknown data and hence make them informative for further usage in the classifier.

2.2.2 Challenges

According to Liu [27] and Feldman [28] the main research tasks of sentiment analysis are investigated at three levels:

Document Level: The task is to classify a complete document whether it expresses a positive or a negative sentiment [27]. This is the simplest form of sentiment analysis as it assumes that the document contains a single opinion on one main object [28].

Sentence Level: The task at this level is to break down the document's sentences and to determine the sentiment of each sentence individually [27].

Entity and Aspect level: This task performs the most fine-grained analysis. It determines the sentiment for different entities contained in a text and/or their different aspects [27]. For example, the entity "surf board" in the sentence "The surf board's price seems fair, given the poor construction quality." has an aspect "price" and "construction quality". Both aspects for the entity have different sentiments: (i) a negative one regarding the "construction quality" aspect and (ii) a

neutral one regarding the "price" aspect.

Feldman [28] mentions two more areas of challenges: comparative sentiment analysis and sentiment lexicon acquisition. *Comparative sentiment analysis* focuses on the sentence level. It tries to identify sentences for a comparable opinion and extracts the preferred entity in each opinion. For example, the sentence "the fin box quality of Surfboard X is of cheaper construction than that of Surfboard Y", compares two entities "Surfboard X" and "Surfboard Y" with regard to their fin box. Clearly, "Surfboard Y" is the preferred entity with respect to the construction quality of the fin box [29]. The challenge is to find the preferred item as comparisons do not underlie a fixed structure such as "Y is better than X". *Sentiment lexicon acquisition* defines approaches to acquire new lexicons and dictionaries. Feldman [28] defines three approaches: hand coding without seed words, expanding sets of seed words from dictionaries, and corpus based approaches where seed words get expanded using a large corpus of documents from a single domain. The challenges of using lexicons in social media are for example, the usage of abbreviations and slang words.

2.2.3 Applications

Opinions play an important role for people. Whenever someone needs to make a decision, an opinion is a key influencer for a behavior [27]. Consumers want to know the opinions of other consumers, e.g. before buying a product. People searching for a new doctor want to know the opinion of other patients. In political debates, people listen to opinions of others and shape their own opinions about, for example, candidates. Nowadays, one is no longer limited to ask friends or family for opinions [27]. Social media platforms and the web

allow users to publicly state opinions about almost anything. Organizations and companies do not need to conduct surveys anymore to gather public opinions about their products and services. However, an average human reader has difficulties identifying relevant information in the vast amount of opinionated data available in social media and the web. To analyze this amount of data automated systems are advantageous.

2.3 Credibility in Social Media

Based on the work of Castillo et. al [30], we use credibility in the sense of believability [30]. An adequate description is "offering reasonable grounds for being believed"[1]. Credible people are believable people and seen as trustworthy when being honest, careful in choice of words, and disinclined to deceive [31]. However, credibility does not solely refer to the person who publishes information. In communication research, information credibility has three parts: (1) message credibility, (2) source credibility, and (3) media credibility [2]. In traditional media such as newspapers, the source of an article is known and media credibility is defined by the medium's owner who takes responsibility for the content published [2]. However, the case for social media is a different one. SNS provide the fundamental basics for the creation and distribution of content, however, they do not check for factual correctness. Anyone can easily register on a SNS, create a profile, and start to publish content without a verification. In many cases, a username is the only information about the author that exists [2].

[1] `http://www.merriam-webster.com/dictionary/credible`

2.3.1 User Archetypes

Users on SNS are able to post content such as messages, images and videos, and share opinions and other content. Users have different intentions in the usage of SNS and even faithful users can distribute wrong information unintentionally. The following descriptions categorize the archetypes of users that share content on SNS.

Typical User: The fundamental participant of a SNS is a user that shares content with friends and followers. The intention is not a harmful one and clearly defined by the SNS' outline. However, a normal user can unintentionally share wrong information based on false perception of credibility of the information source.

Spammers: Spammers drive malicious campaigns inside SNS. They spam users with messages containing advertisements and try to lure users to malicious websites [5].

Software Robots: Software programs written to automatically post on SNS. Software robots can be used in different ways, for example, posting excerpts of blog posts to social media channels or to support candidates in political debates [32].

2.3.2 Challenges

With the increasing usage of social media platforms as information resource in time sensitive events, there is a need to implement credibility checks for content distributed on such platforms [33]. A core challenge in this context is the general user perception of credibility in social media. Morris et al. [33] found, that users have difficulty determining the truthfulness of social media

posts when the author is not known to them. From December 2009 to July 2011, Google provided a real time search page for Twitter posts and Facebook posts among others. Users where able to search for posts and follow certain topics aggregated from different social media platforms. The authors found, that the judgment for credibility of certain posts are often based on heuristics (e.g how often a tweet was re-tweeted by another user) and biased systematically (e.g. topic-related usernames are perceived as more credible) [33]. The authors proposed pieces of information which are important to highlight on search result pages for social media posts to allow for easier credibility assessments (e.g. highlighting re-tweet stats, number of mentions, and author's details) [33]. The highlighting of such meta data is a helpful information, however, it does not express any credibility information on the author itself or the shared content.

Castillo et al. [30] were the pioneers to research information credibility on Twitter. The authors' hypothesis was that the level of credibility of information disseminated through social media can be estimated automatically based on several factors that can be observed in the social media platform itself [30]. The proposed factors are: (i) reactions to certain topics (e.g. if users use opinion expressions that represent positive or negative sentiments), (ii) the level of certainty of users propagating an information (e.g. whether or not they question the information that is given to them), (iii) external sources cited (e.g. whether or not is a URL cited as source from a popular domain), (iv) characteristics of the user that propagates the information (e.g. number of followers of the user on the platform) [30]. The researchers built feature sets based on these factors and employed machine learning algorithms to build a classifier for posts on Twitter. However, challenges arise with the participation of more powerful parties in discussions on social media. For example, a view on discussions in the political domain outlines the amount

of content that needs automatic approaches to verify credibility. Recently, software robots (bots) started to appear. These bots are programmed to interact with humans on social media and try to emulate or alter the behavior of users [34]. Ratkiewicz et al. [32] found, that bots were used during the 2010 U.S. midterm elections to either support specific candidates or to harm other candidates with postings on Twitter that point to fake news websites. Bessi and Ferrara [35] found, that about 400.000 bots were engaged in discussions around the 2016 U.S. presidential election on Twitter[2]. Between 16th of September and 21st of October 2016 roughly 3.8 million posts from a total of 20 million posts were generated by bots [35]. While ongoing research leads to better detection systems, researchers expect an arms race as bots evolve constantly and bring new challenges for credibility assessment methods [35].

[2] 2.8 million unique users in total

Chapter 3

State of the Art

This chapter gives an overview of the state of the art research aresas related to this thesis: sentiment analysis, user credibility, and social media stream visualization.

3.1 Sentiment Analysis

In the following sections, we take a closer look at some state of the art research projects in the context of sentiment analysis in social media. The following presented projects include different supervised and unsupervised approaches which inspired this thesis.

The first supervised approach by Go et al. [36], was one of the first to automatically classify the sentiment of Twitter messages. The second approach by Wang et al. [37] detects public sentiment within Twitter messages in the political domain. Paltoglou and Thelwall [38] present a unsupervised approach that employs a dictionary to detect the polarity of text in social

media. The dictionary is based on *SentiStrenght*[1], an often used program for sentiment analysis. The approach by Turney [12] is the first unsupervised approach to automatically classifies reviews of different domains. Finally, the last approach by Nielsen [25] presents an evaluation of different dictionaries that can be employed for sentiment analysis in social media.

3.1.1 Supervised Approaches

Go et al. [36] proposed an approach to classify the sentiment of Twitter messages as positive or negative. The authors used *Distant Supervision* to generate a set of labeled data. They employed the Twitter API to gather training data using emoticons, e.g. ":(" and ":)" indicating positive and negative sentiment in a tweet respectively. The result was a training data set with 1,600,000 tweets (50% positive and 50% negative). The researchers used unigram, bigram, unigram+bigram, and unigram+POS features to learn three different classifiers: Naive Bayes [18], maximum entropy (MaxEnt) [20], and support vector machines (SVM) [19]. Furthermore, a keyword-based classifier was used as a baseline to compare the results. To reduce the feature space the authors replaced all URLs and usernames in a tweet with predefined tokens and removed repeated letters in tweets (e.g. "huuuungry" to "hungry"). To generate the test data, the authors employed the Twitter API using queries from different domains (e.g. consumer products, companies, and people). They manually checked the results and marked a tweet as positive or negative if the tweet contained a sentiment. The resulting test set contained 359 tweets (49.30% negative and 50.70% positive tweets). The classification on unigram features resulted in an accuracy of 81.3%, 80.5%, and 82.2% using the Naive Bayes [18], MaxEnt [20] and SVM respectively [19]. The authors

[1] `http://sentistrength.wlv.ac.uk/`

did not provide results for bigram features because the feature space was very sparse and the overall accuracy dropped [36]. For unigram+bigram features the accuracy improved for Naive Bayes [18] (from 81.3% to 82.7%) and MaxEnt [20] (from 80.5% to 82.7%) but declined for SVM [19] (from 82.2% to 81.6%). For unigram+POS features the accuracy dropped to 79.9%, 79.9%, 81.9% for Naive Bayes [18], MaxEnt [20] and SVM [19] in comparison to the unigram features.

Wang et al. [37] proposed a supervised approach to create a sentiment analysis model using data in the political domain. The dataset was created by collecting tweets from the candidates based on manual rule sets. The authors employed Amazon Mechanical Turk[2] to annotate the collected data and to create the baseline sentiment model. The about 800 annotators were asked to label a tweets' sentiment in one out of four categories: positive, negative, neutral, or unsure. Additionally, they were asked to annotate each tweet as sarcastic or humorous, to define its sentiment on a scale from positive to negative, and to estimate the tweet author's political orientation. The collected and annotated data consisted of nearly 17,000 tweets (16% positive, 56% negative, 18% neutral, and 10% unsure). The sentiment model was build using the general sentiment and the additional sarcasm and humor labels [37]. The authors employed the Naive Bayes classifier [18] on unigram features extracted from the tweets to learn the sentiment model. The authors used Christopher Potts' basic Twitter tokenizer[3] to create the features since it correctly handles numbers, HTML Tags, Twitter-specific phenomenas (e.g. mentions, extracting intact URLs), common emoticons, repetition of symbols, and unicode characters [37]. The classifier achieved 59% accuracy in the classification of unknown tweets in one of the four general sentiment

[2] `https://www.mturk.com`

[3] `http://sentiment.christopherpotts.net/tokenizing.html`

categories: negative, positive, neutral, and unsure.

Despite the fact that the two presented approaches employed Naive Bayes on unigram features, a direct comparison between the results is not possible as both employed data sets vary strongly in terms of domain and quantity. Go et al. [36] used an almost 1000x bigger training data set with two labels, positive and negative, while Wang et al. [37] used a training set with 17,000 tweets and four labels (positive, negative, neutral, and unsure).

3.1.2 Unsupervised Approaches

Paltoglou and Thelwall [38] proposed an algorithm for opinion[4] and polarity detection of text in social media. The algorithm uses the emotional dictionary LIWC as basis. LIWC includes 905 words in two categories especially related to sentiment analysis (positive and negative emotions). The authors use adapted weights for the dictionary which are better suited for the type of informal communication usually found in social media [39]. The weights are the result of the development of a sentiment strength detection algorithm (SentiStrength) in short informal text [39]. Paltoglou's and Thelwall's [38] approach is a typical example for an unsupervised method because no training and no reference corpus is required. For data preprocessing, the authors use the Porter Stemmer [14] to stemm all dictionary lemmas and the processed text. The algorithm checks for every word in a document if there is an entry in the LIWC dictionary. If an entry exists, it extracts the polarity and intensity. The sentiment of a word can range from $C_{pos} = \{+1,+2,...,+5\}$ for positive emotion words (+5 means very positive) and $C_{neg} = \{-1,-2,...,-5\}$ for negative emotion words (-5 means very negative). To produce the final document scores, the authors implement additional prose-driven functionali-

[4]Detecting if a text contains an expression of opinion or if it is objective

ties to detect terms which work as modifiers for the initially found sentiment of a word. For example, negators (e.g. "not") or intensifiers (e.g. "very") decrement or increment the sentiment of a word based on predefined criteria. The score of a document $\{C_{pos}, C_{neg}\}$ is the maximum positive and the maximum negative value produced on the C_{pos} and C_{neg} scales. To classify the document as positive or negative, the class with the highest absolute value is used. For example a {3,-2} document score is classified as positive while a score of {3,-4} is classified as negative. A document is classified as objective, when the score is {1,-1}. Conflicts of equality, such as {2,-2}, are solved by giving preference to the class with the largest number of positive or negative tokens. The proposed approach is compared against three machine learning approaches, namely Naive Bayes [18], SVM [19], and MaxEnt [20] using unigram features with different weighting schemes[5]. Paltoglou and Thelwall [38] used free available datasets consisting of Twitter tweets [40], Digg messages [41], and MySpace comments [42] to train and test all approaches. The proposed unsupervised approach outperformed the considered classifiers in terms of the achieved F1-score [13]. The results indicate that the proposed approach is a reliable solution for working with informal communication on the web [38].

Turney [12] proposed a different approach for unsupervised sentiment analysis. The approach classifies a review as recommended or not recommended based on the average semantic orientation of its phrases. The author analyzed reviews from different domains such as automobiles, banks, movies, and travel and proposed a three-step approach to detect sentiment in reviews by extracting two-word phrases based on pattern of tags. In step one, a part-of-speech tagger identifies phrases in the input text that contains

[5]For example, the authors employed the frequency of a term or its presence/absence as a weight in the corresponding feature function

adjectives or adverbs. In the second step, an algorithm called *Pointwise Mutual Information and Information Retrieval* (PMI-IR) determines the semantic orientation of the extracted phrases. To check for positive or negative associations the PMI-IR compares an extracted phrase with its similarity to a positive ("poor") and a negative ("excellent") reference word. The third step involves the classification of the review as recommended or not recommended by comparing the average semantic orientation of the phrases and a baseline value. Turney's [12] results differ among the four domains. While the classification for all 410 reviews results in an average of 74% accuracy, the movie domain achieved the lowest accuracy (65.83%). In contrast, the automobile domain achieved the highest accuracy (85%). Turney's [12] interpretation is that within the movie domain, the whole is not the sum of the parts, e.g. a movie review can be positive in terms of filming but negative in terms of acting and again positive in terms of the plot.

There exist different dictionaries which are used to perform unsupervised sentiment analysis on text. However, some are better suited for text found in social media than others. Nielsen [25] presents an evaluation of a dictionary for sentiment analysis of social media text. In 2009, Nielsen [25] set up a word list (AFINN-96) for online sentiment analysis in the context of the United Nation Climate Conference (COP15). It included 1,468 different words and a few phrases. Nielsen [25] started the list with a set of obscene words and a few positive words. AFINN-96 grew with the examination of the COP15 tweets. The current version, AFINN-111, includes words from the public domain such as the *Original Balanced Affective Word List*[6] and the Internet slang words included in the Urban Dictionary[7]. As in SentiStrength [39], the scores for a word range from -5 (very negative) to +5 (very positive). In

[6] `http://www.sci.sdsu.edu/CAL/wordlist/origwordlist.html`
[7] `http://www.urbandictionary.com`

AFINN-111, most of the positive words are labeled with +2 and most of the negative words with -2. AFIN-111 has 2,477 unique words and 15 phrases and is biased towards negative words (1,598) in comparison to positive words (878). One single phrase is labeled with valence 0. The evaluation of AFINN-111 includes a comparison of the dictionary against the *Affective Norms for English Words* (ANEW) [24], General Inquirer[8], OpinionFinder [43], and SentiStrength [39]. Nielsen employed a data set of 1,000 tweets labeled by Amazon Mechanical Turk[9] (AMT) annotators from a previous study which is used as ground truth [44]. To compute a sentiment score, the algorithm looks up the words in the dictionaries and the corresponding valence score. Nielsen implements different weighting schemes to calculate the sentiment strength of a tweet. For example, the author sums sup all valences and divides the sum by the number of words or uses the sum of valence without normalization of the words. To evaluate the performances in comparison to the AMT ground truth data set, Nielsen uses two different correlation coefficients as metrics: Pearson correlation and Spearman's rank correlation. The results show that AFINN-111 performs slightly better than ANEW. The more elaborated SentiStrenght shows overall best performances. GeneralInquirer (3,392 words) and OpinionFinder (6,442 words), despite having the largest word lists, perform not as good as SentiStrenght, ANEW, and AFINN-111. Nielsen concludes that the performance of AFINN-111 over ANEW may be due to the inclusion of Internet slang and obscene words [25].

[8] `http://www.wjh.harvard.edu/~inquirer/`
[9] `https://www.mturk.com`

3.2 User Credibility

User credibility and detecting spammy users in social media is a growing concern platforms currently face. Spam is not only presented in email messages but social media platforms are more and more targets of cybercriminals and dubious businesses. Information credibility in social media as a broader term encapsulates user credibility by defining sets of different features not only belonging to the user itself but to the content a user produces [45]. In the following, we take a closer look at some state of the art research projects in the context of user credibility. The demonstrated projects cover different domains to point out the different aspects of user credibility. The first approach by Stringhini et al. [5] detects spammers in social networks. The second approach by Gupta et al. [46] detects trustworthy users on Twitter [46]. Finally, the last approach by Diakopoulos et al. [47] identifies credible sources for journalists in Twitter [47].

Stringhini et al. [5] presented an approach to detect spammers in social networks. The authors investigated three platforms and obtained data within a 12 month period for Facebook and an 11 month period for MySpace and Twitter by creating a set of honeynet accounts and logging all activity these accounts generated (e.g. friend requests, messages, invitations). On Facebook, these accounts were created within different geographic networks. In the past, most Facebook users were grouped in such networks and the default privacy setting was to allow all people in the same geographical network to view each other's profiles[10]. With the collected data the researchers were able to identify general characteristics of spammers in social networks. The authors identified four different types of spammers according to their spam strategy: (1) spam content on the own profile, (2) post on feeds of users,

[10]Facebook deprecated geographic networks in October 2009

(3) send direct messages, (4) and post on their own feeds. The researchers developed six features to detect spam profiles such as Following/Followers (FF) ratio[11], URL ratio, message similarity, friend choice, messages sent, and friend number. The authors built two systems to detect spammers on Facebook and Twitter and both systems use the Random Forest classifier [48]. For Facebook, the researchers trained the classifier using a training data set of 1,000 Facebook profiles. 173 were spam profiles that contacted the honeynet accounts and 827 were manually checked profiles as samples for legitimate users. The authors applied the classifier to 790,951 profiles belonging to the Los Angeles and New York geographic networks and identified 130 spammers including 7 false positives [5]. The authors hypothesis', why the number of detected spammers is so low, is that spammers usually do not join geographic networks. This is backed by the fact that no spam user which contacted the honeynet accounts was a member in such a geographic network [5]. The approach was adapted to Twitter and since most profiles on Twitter are public, the researchers could develop a system to detect spammers in live data. The classifier was trained with a training data set of 1,000 Twitter profiles. 500 were spam profiles coming either from the ones that contacted the honeynet accounts or manually selected from the public timeline to increase diversity among spam profiles [5]. The other 500 profiles were manually selected from the public timeline as samples for legitimate profiles [5]. The researchers trained the classifier and modified or removed features which where identified as not useful. For example, FF ratio was intended to reflect the Following/Followers ratio on Twitter. The authors expected the ratio to be large for spammers and low for regular users. However, when the researchers examined legitimate Twitter profiles they found that a profile can have a fairly high number of followers, but following thou-

[11]Following/Followers ratio only on Twitter possible

sands of other profiles. The result is a high ratio value which can lead to false positives. The feature was weighted by dividing the FF ratio by the number of followers. The friend choice feature attempts to detect whether or not a profile likely used a list of names to pick its friends [5]. However, it was omitted, since spammers and legitimate profiles in the training set had very similar values for this parameter [5]. A 10-fold cross validation yielded an estimated false positive ratio of 2.5% and a false negative ratio of 3% on the training set [5]. Twitter's API call limitation (20,000 calls per hour) led to an implementation as a service where users can flag profiles as spam accounts. After around 3 months and 135,834 crawled profiles, the researchers detected 15,932 spam profiles and reported them to Twitter with the response that only 75 were false positives [5].

Gupta et al. [46] developed a browser-plugin, a REST API, and a web application to display real-time credibility rating for a tweet named *TweetCred*. To create the credibility model the researchers adopted a semi-supervised learning-to-rank approach and used a collection of features. The authors restrict the features to those which can be derived from a single tweet in real-time. They used 45 features and grouped them into six feature-sets: (1) tweet meta-data (e.g. source of tweet, tweet contains geo-coordinates), (2) tweet content simple (e.g. number of characters, tweet contains "via"), (3) tweet content linguistic (e.g. presence of swear words, presence of pronouns), (4) tweet author (e.g. number of followers, time since user is on twitter), (5) tweet network (e.g. number of retweets, number of mentions), (6) tweet links (e.g. ratio of likes / dislikes for a YouTube video) [46]. The training data consisted of tweets from six prominent events in 2013. To label the data and to create a ground-truth the authors used a crowd-sourcing provider where annotators labeled tweets in four categories: (1) definitely credible, (2) seems credible, (3) definitely incredible, and (4) none (of the

mentioned). Based on this labeled data the authors tested multiple learning-to-rank algorithms, namely Coordinate Ascent [49], AdaRank [50], RankBoost [51], and SVM-rank [52], in order to rank tweets by credibility. Two metrics were used to evaluate the results, Normalized Discounted Cumulative Gain (NDCG) and run time. The different ranking schemes were evaluated using 4-fold cross validation on the training data. The results show that AdaRank [50] and Coordinate Ascent [49] perform best in terms of NDCG@n (n={25, 50, 75, 100}) and SVM-rank [52] is second. The researchers also present train time and test time for all methods where SVM-rank requires lowest training times. Given the future work use case in developing a system which re-trains the credibility model using feedback from users [46] and hence need lower train times, the authors implemented the SVM-rank [52] in the system. The top 10 features for the model built using SVM-rank [52] were: (1) tweet contains "via", (2) number of characters, (3) number of unique characters, (4) number of words, (5) user has location in profile, (6) number of retweets, (7) age of tweet, (8) tweet contains a URL, (9) ratio number of statuses/followers of the author, and (10) ratio FF of the author [46].

Diakopoulos et al. [47] also addressed the issue of finding relevant and trustworthy sources in social media. The authors developed a prototype which allows users to find and assess information sources in social media based on Twitter data. The authors worked together with a group of journalists to gain deeper knowledge in the structures and processes a journalist employs in order to find credible persons and trustworthy reports in social media. Based on the gained insights Diakopoulos et al. [47] created a prototype of a web application which shows relevant information for journalists in case of "breaking news events". Using machine learning algorithms and dictionary-based approaches the researchers developed a number of classifiers to cover different use cases, e.g. an eyewitness detector to check whether or

not a tweet author is an eyewitness of an event. To create the eyewitness classifier the authors employed the emotional dictionary LIWC [22] which not only contains words for sentiment analysis but it also provides a rich set of words categorized across 70 different language dimensions [47]. The classifier checks if the words of a tweet belong to LIWC categories which reflect a user's presence during an event in space/time (e.g. percept, see, hear, feel). The result of the classifier is a binary label whether or not a tweet's author is likely to be an eyewitness. The authors evaluated the prototype interface with the group of journalists and became some helpful suggestions on filters for future work and development, e.g. a sentiment analysis based filter to find more fact-based information sources instead of opinionated.

3.3 Social Media Stream Visualization

Visualizing content from social media networks heavily depends on the use case and on the employed data. Different approaches focus on different cues and favour a more visual or a more textual output. In this section, we take a closer look at some state of the art research projects that propose visualization possibilities of data derived from social media. The first presented approach by Hao et al. [53] displays sentiment analysis results with different color cues [53]. The second approach by Diakopoulos et al. [47] proposes a user-profile centric interface [47]. Finally the last approach by Marcus et al. [54] develops a dashboard interface which updates on live events and provides overall information for tweets and tweet authors [54].

Hao et al. [53] propose an approach to visualize sentiment analysis on Twitter data streams (see Figure 3.1). The dataset consisted of 59,614 tweets of comments for a predefined movie (KungFu Panda). The visual output is

a *Pixel Sentiment Geo Map* and a *Pixel Sentiment Calendar*. An image of a map of the world is used for the *Pixel Sentiment Geo Map* and it displays the analyzed data. For every sentiment a different color is defined (green for positive, red for negative, gray for neutral) and displayed on the world map where the color indicated a tweet's sentiment and the position on the map the location the tweet comes from. *The Pixel Sentiment Calendar* is a table representation, where each row stands for a topic and each column for a time interval. Within the columns, every opinion is represented by a cell whose color depicts the sentiment of the underlying opinion.

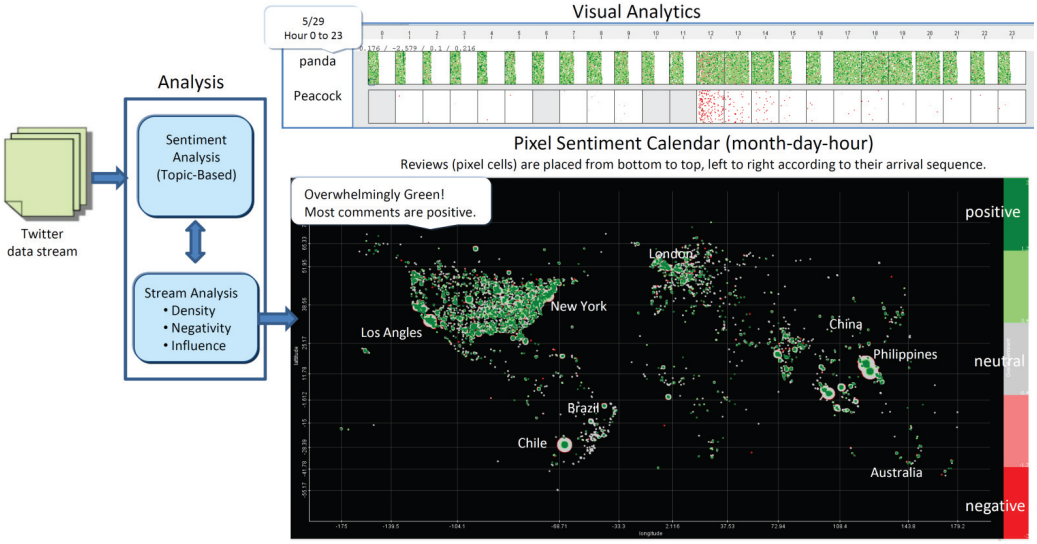

Figure 3.1: Pixel Sentiment Calendar and Pixel Sentiment Geo Map [53].

Diakopoulos et al. [47] developed an interface for journalists to find potential sources in social media named *Seriously Rapid Source Review* (SRSR) (see Figure 3.2). The interface is user-profile centric and incorporates many of the standard elements from Twitter such as full user name, screen name, followers, and followees [47]. In addition, the SRSR interface provides aggregated and derived profile information as add-on cues such as an eyewitness icon to help to identify eyewitnesses and a pie chart to indicate the top three locations where the user's Twitter contacts are located. Instead of using only colors for different values, the authors used small icons and details which compose a good overview over the calculated data. Different filter options allow the user to browse through different SRSR defined user types (e.g, natural person, blogger, corporate) and to sort based on different values (e.g. # times retweeted).

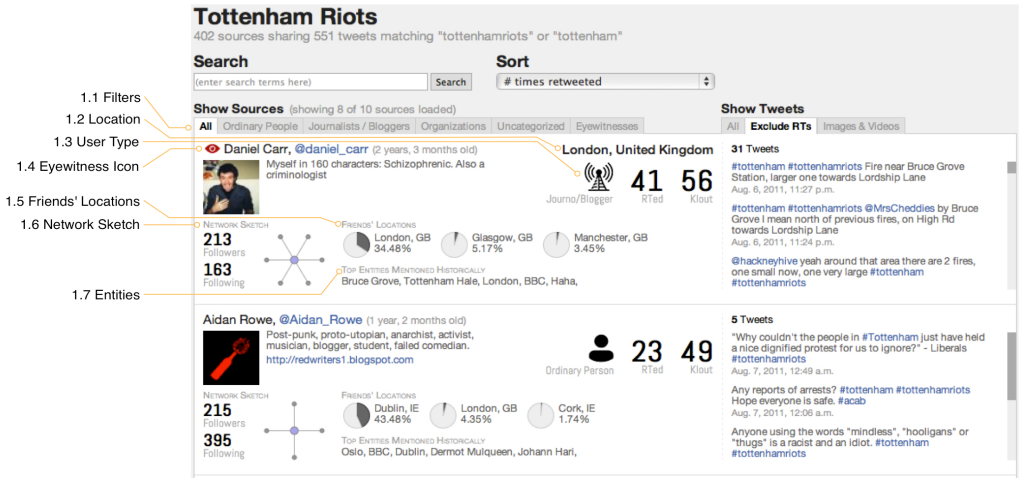

Figure 3.2: SRSR prototype interface [47].

Marcus et al. [54] created a web application named *TwitInfo* to summarize and visualize events on Twitter (see Figure 3.3). The authors created a platform where users can define a name for an event by using different keywords and hashtags. The application streams from the Twitter API, extracts

tweets matching the search criteria, and creates a dashboard for the event. Every new tweet for the event is displayed on the time-axis. An event peak detection system is implemented to highlight peaks of high tweet activity. The user can drill down these peaks where certain sub events can be further explored. The dashboard shows the frequency of the messages, relevant tweets, a world map with the location of the tweets and popular links as well as the overall sentiment of an event.

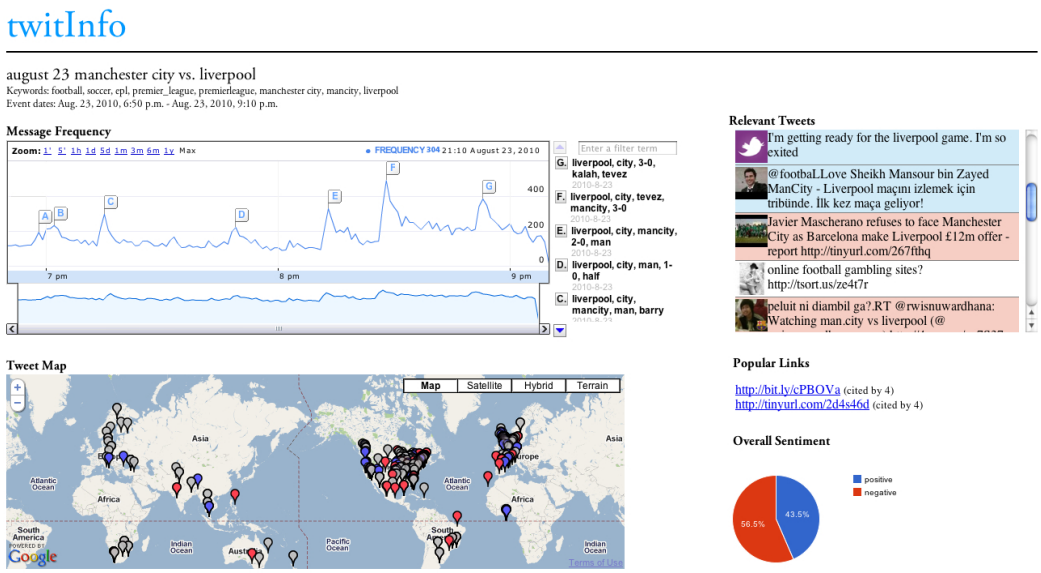

Figure 3.3: TwitInfo Dashboard [54].

Chapter 4

Methodology

This chapter outlines the methodology of this thesis. We identify the critical blocks in order to develop a prototype that enables searching for messages and attached media objects on Twitter under the consideration of two crucial aspects: credibility and sentiment.

4.1 Overview

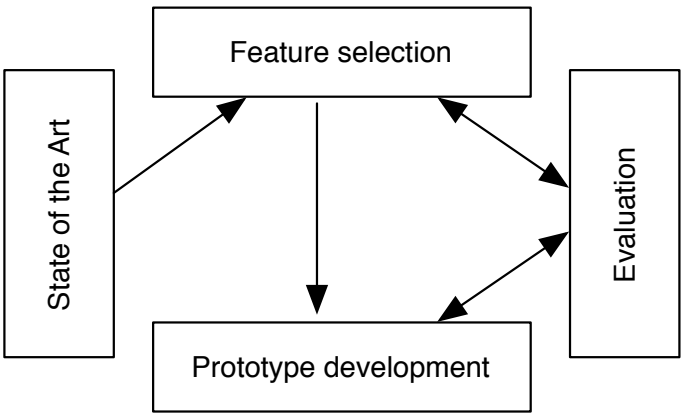

Figure 4.1: Methodology overview.

State of the Art: A state of the art literature research provides an overview of current projects in the space of this thesis as well as a fundamental basis of knowledge.

Feature selection: The feature selection derives from the acquired knowledge of the state of the art literature research. A critical challenge to overcome is the fast pace social media moves forward. A user study informs about potential new challenges in todays social media usage and allows to identify concepts and features regarding user credibility. User feedback supports the selection process and allows to validate new concepts against already valid approaches. To overcome the challenge of potential users, the user study is divided in two study groups with one containing experts and one containing sporadic social media platform users.

Evaluation: Evaluation is a critical concept to measure the quality of the proposed work. Actual challenges are quality and availability of data sets and

a selection of metrics that allow meaningful comparison. The focus for the evaluation is on data sets that are freely available and used in other projects. Evaluation metrics of similar approaches are used to allow a comparison of projects using the same data sets.

Prototype development: The prototype is a tool to demonstrate the implementation of the proposed concepts. An evaluation of the prototype is critical and allows to collect user feedback which can be applied in further iterations and future work.

Chapter 5

User Study

This chapter contains implementation details and results of the initial conducted user study. Section 5.1 describes the design of the study followed by the setup process. Section 5.2 describes the evaluation of the user study and the subsequent feature definition. Section 5.3 describes the feature selection process and the evaluation of the selected feature sets.

5.1 Design

The aim of the study was to identify, why users decide whether or not a tweet is credible. The study was two-fold: First, we wanted to evaluate on which basis our users agree with TweetCred [46], an approach that rates a tweet with values ranging from 1 (”low credibility”) to 7 (”high credibility”) [46]. Second, we asked for written feedback to understand the important factors for a user when judging the credibility of a tweet in order to explore features that describe credible and not credible tweets.

5.1.1 Setup

We used the freely available data set from Sanders[1] to construct the data sets for the study. The initial data set consisted of 5,513 tweets with manually labeled sentiment annotations. We chose a sentiment based data set to explore correlations between credibility among others. However, at the time of writing 1,102 tweets (19.9%) weren't available anymore due to the age of the data (e.g tweets deleted by the user or user account removed). We imported the data set into a database resulting in a total of 4,411 tweets created by 3,566 users.

To add a baseline for credibility we employed TweetCred on the data set and annotated the rating for every tweet. Table 5.1 displays the total count of tweets for a TweetCred rating.

TweetCred Rating	**# Tweets**
1	1,635
2	786
3	785
4	685
5	409
6	96
7	11
Total	4,411

Table 5.1: Number of tweets for TweetCred rating

We defined three human-readable categories: "Not Credible", "Don't Know", and "Credible" to create the data sets. We mapped the TweetCred ratings according to the categories in Table 5.2 and omitted 1,194 tweets (27%) for the fuzzy values 3 and 5.

[1] `http://www.sananalytics.com/lab/twitter-sentiment/`

TweetCred Rating	Category	# Tweets
{1,2}	Not Credible	2,421
{4}	Don't Know	685
{6,7}	Credible	107
	Total	3,213

Table 5.2: Mapping of TweetCred rating to category.

We randomly chose 100 English tweets from the category "Not Credible", 100 English tweets from the category "Don't Know", and 58 English tweets from the category "Credible" (the category "Credible" contained 49 non-English tweets and were therefore omitted).

Category	# Tweets
Not Credible	100
Don't Know	100
Credible	58
Total	258

Table 5.3: Number of tweets per category.

The 258 tweets became annotated with additional data points to explore different correlations. We manually labeled every author of a tweet as a person or company account. We checked if an author defined a location in his profile and verified the existence of the location by employing the Gate Cloud Service[2] using the service *English Named Entity Recognizer for tweets*. We used the same service to verify if a user stated a real name in its profile and we used the service *Part-of-Speech Tagger for tweets* to extract the POS-Tags of a tweet. Ultimately we manually labeled whether an author used a graphic, a photo as a profile picture or none.

[2] `https://cloud.gate.ac.uk`

5.1.2 Implementation

The user study was separated in two participation groups. The first group (Regular Users) consisted of ten persons that were asked to rate the prepared 258 tweets. The only requirement to participate was to have a simple knowledge of the Twitter platform. The data was distributed over all participants resulting in nine data sets with 26 tweets and one with 24 tweets. Every participant received a unique link to access the evaluation platform. Figure 5.1 shows the interface of the platform.

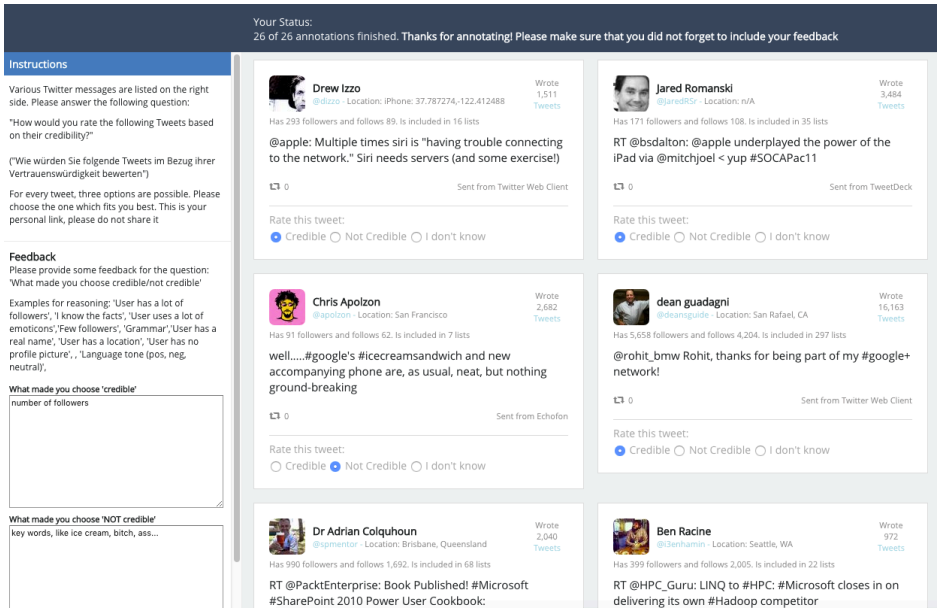

Figure 5.1: Evaluation platform.

An instructional text was located on the left side of the screen. On the right side the tweets were listed in a random order. The tweet listing included a form field for every tweet with three options: "Credible", "Don't Know" or "Not Credible": The users had a time span of seven days to complete the two following tasks:

(1) To choose one of the three options and rate every tweet based on the following question: "How would you rate the following tweets based on their credibility?".

(2) To provide textual feedback for the question: "What made you choose credible / not credible?".

The second group (Expert Users) consisted of six persons with deeper knowledge of Twitter (e.g. familiar with the notation of Twitter messages such as abbreviations, usage of hashtags and mentions). The data set for the Expert Users was compiled from a fraction of the rated tweets of the Regular Users. Precisely, the data set consisted of tweets from the category "Credible" and "Not Credible" where no agreement with TweetCred and the Regular User group could be obtained. We asked the Expert Users to complete the same two tasks as stated above and gathered the data within a week.

5.2 Evaluation

The evaluation of the user study is separated into three parts: (1) We first evaluate the rating results of our user study in comparison with the ratings of TweetCred. (2) We then analyze the data set to obtain quantitative results from the underlying data. (3) Finally, we evaluate the received textual feedback to identify feature categories and features for credibility assessment.

5.2.1 User Ratings compared to TweetCred Ratings

Table 5.4 displays the confusion matrix for the 258 tweets data set with the ratings of the Regular Users in comparison to the TweetCred ratings.

		Regular Users		
		Credible	Not Credible	Don't Know
TweetCred	Credible	35 (13.57%)	11 (4.26%)	12 (4.65%)
	Not Credible	37 (14.34%)	43 (16.67%)	20 (7.75 %)
	Don't Know	53 (20.54%)	31 (12.02%)	16 (6.20%)

Table 5.4: TweetCred vs. Regular User group ratings.

We can observe that 11 tweets (4.26%) which TweetCred rated as "Credible", obtained a "Not Credible" rating by the Regular Users. The same can be observed for the 37 tweets (14.34%) that TweetCred rated as "Not Credible" but obtained a "Credible" rating by the Regular Users. To assert the ratings for this 48 tweets (18.60%), we distributed data sets of 16 tweets per person to the Expert Users. The combination of the rating from the Regular Users and the Expert Users resulted in a total of three ratings for every one of the 48 tweets. Table 5.5 displays the results based on a majority-agreement and highlights a discrepancy of TweetCred ratings and user ratings for tweets in the category "Credible" and "Not Credible".

		Majority		
		Credible	Not Credible	Don't Know
TweetCred	Credible	4 (8.33%)	7 (14.58%)	0
	Not Credible	29 (60.42%)	4 (8.33%)	4 (8.33%)
	Don't Know	0	0	0

Table 5.5: TweetCred vs. majority ratings.

Of the initial 258 user rated tweets, 94 tweets (36.43%) obtained an agreement with the TweetCred ratings. These results are in the range of the

observations from the authors of the TweetCred study. The researchers received feedback for 1,273 tweets observing a 40.14% level of agreement (see Figure 5.2).

	Observed	95% Conf. interval
Agreed with score	40.14	(36.73, 43.77)
Disagreed with score	59.85	(55.68, 64.26)
Disagreed: score should be higher	48.62	(44.86, 52.61)
Disagreed: score should be lower	11.23	(9.82, 13.65)
Disagreed by 1 point	8.71	(7.17, 10.50)
Disagreed by 2 points	14.29	(12.29, 16.53)
Disagreed by 3 points	12.80	(10.91, 14.92)
Disagreed by 4 points	10.91	(9.17, 12.89)
Disagreed by 5 points	6.52	(5.19, 8.08)
Disagreed by 6 points	6.59	(5.26, 8.16)

Figure 5.2: TweetCred agreement based on 1,273 tweets [46].

Additionally, the authors found that users which disagreed with the rating felt that the credibility rating should have been higher [46]. Although we summarized and categorized the ratings of TweetCred in to three manually defined categories, we can see a trend into the same direction. Table 5.6 shows the agreement and disagreement scores from our study.

Agreed with score	36.43%
Disagreed with score	63.57%
Disagreed: score should be higher	42.64%
Disagreed: score should be lower	20.93%

Table 5.6: Agreement/Disagreement scores of the Regular User group.

Out of a total of 164 tweets (63.57%) that obtained a disagreement with TweetCred, 110 tweets (42.64%) obtained a user rating that was higher

whereas 54 tweets (20.93%) obtained a rating that was lower. Hence, our findings overlap with the results from the authors of TweetCred that TweetCred tends to produce credibility scores that are lower than what users expect [46].

5.2.2 User Rating Results

Based on the data of the user study with the Regular Users and a total of 258 tweets, we observed the following results.

We found the majority of tweets that obtained a "Credible" rating belong to the account type "Company". Out of a total of 66 tweets (25.58%) that belong to company accounts, 41 tweets (62.12%) were rated as "Credible" (see Figure 5.3).

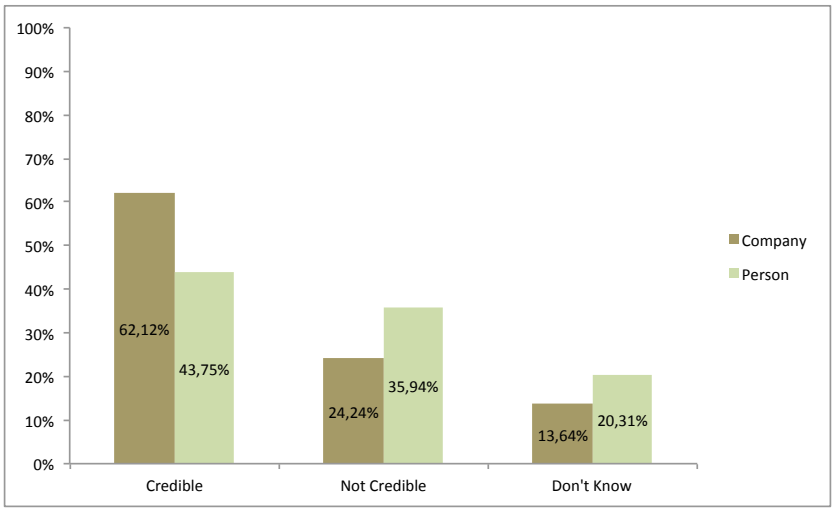

Figure 5.3: Observed credibility for company / person

Next we analyzed account type and profile picture type in correlation with credibility. We separated the profile pictures in to three groups: "Graphic", "Photo" and "None". We found that account types of "Company" are more

likely to use graphical profile pictures whereas "Person" accounts are more likely to use photos as profile pictures. We observed a tendency to obtain a "Not Credible" rating when no profile picture is defined (see Figure 5.4).

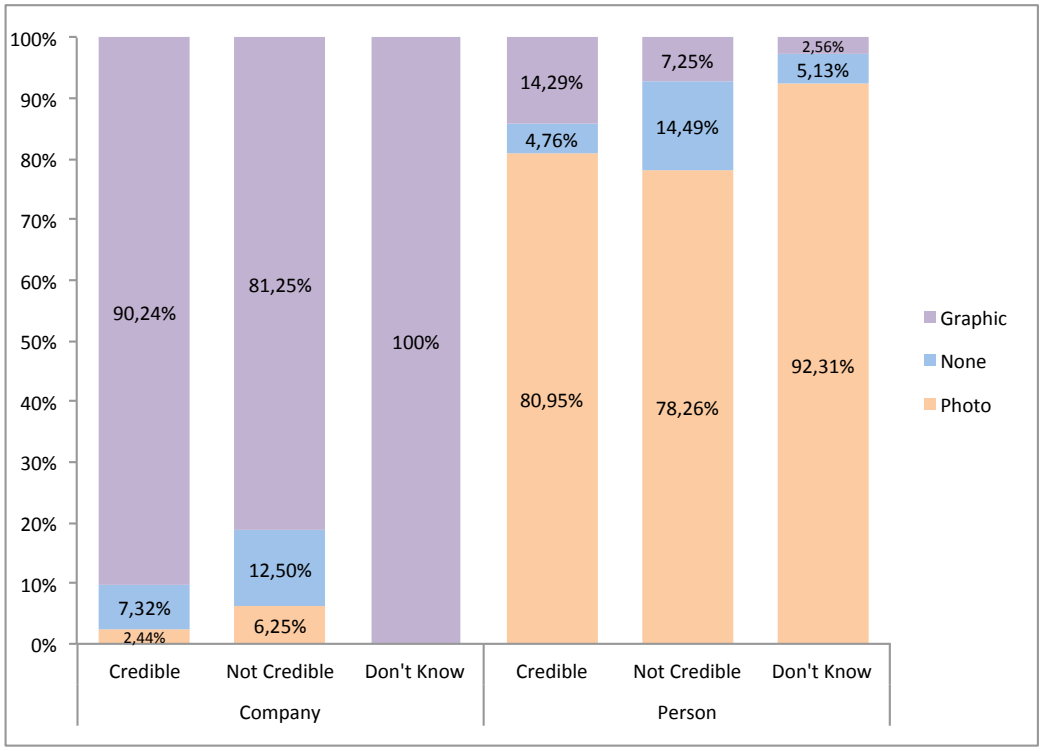

Figure 5.4: Credibility for profile picture type and account type.

Next we analyzed the correlation between credibility and sentiment separated by account type. Due to a bias for company accounts with mostly neutral tweets there is no significant outcome for this account type. However, for account type "person" the credibility rating is highest for neutral tweets (76.19%). Negative tweets are more likely to receive a "Don't Know" rating (see Figure 5.5).

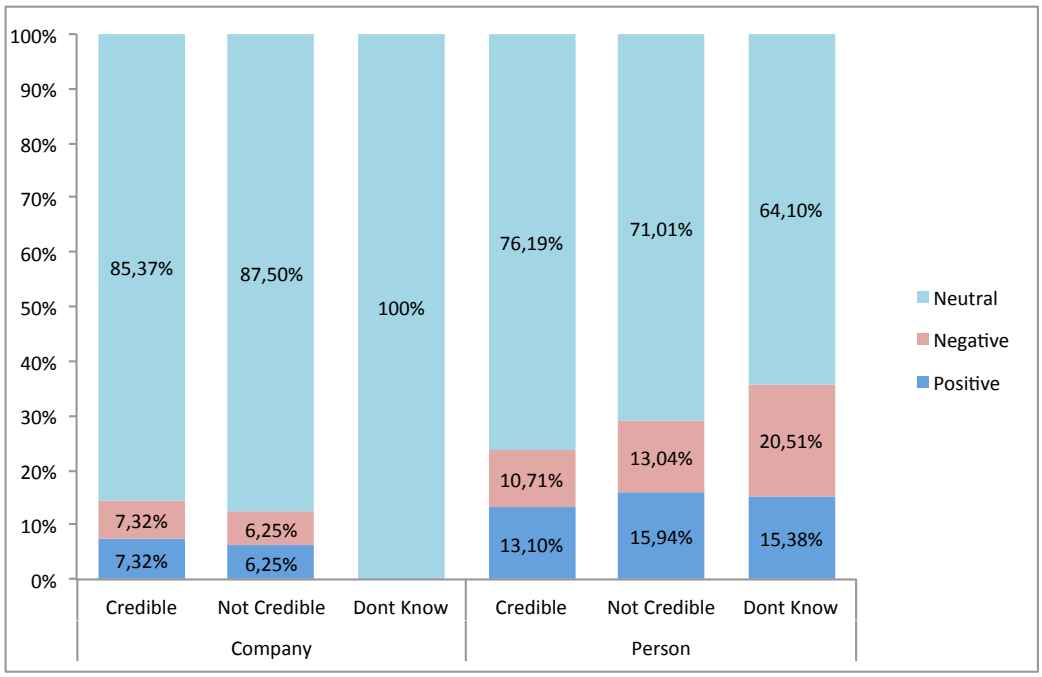

Figure 5.5: Sentiment and credibility for account type.

Finally, we analyzed correlation of sentiment and credibility. The results show that more than a half (50.51%) of all neutral tweets obtained a "Credible" rating. Positive tweets obtained the highest "Not Credible" rating (37.50%) and tweets with a negative sentiment obtained the highest "Don't Know" rating (26.67%) (see Figure 5.6).

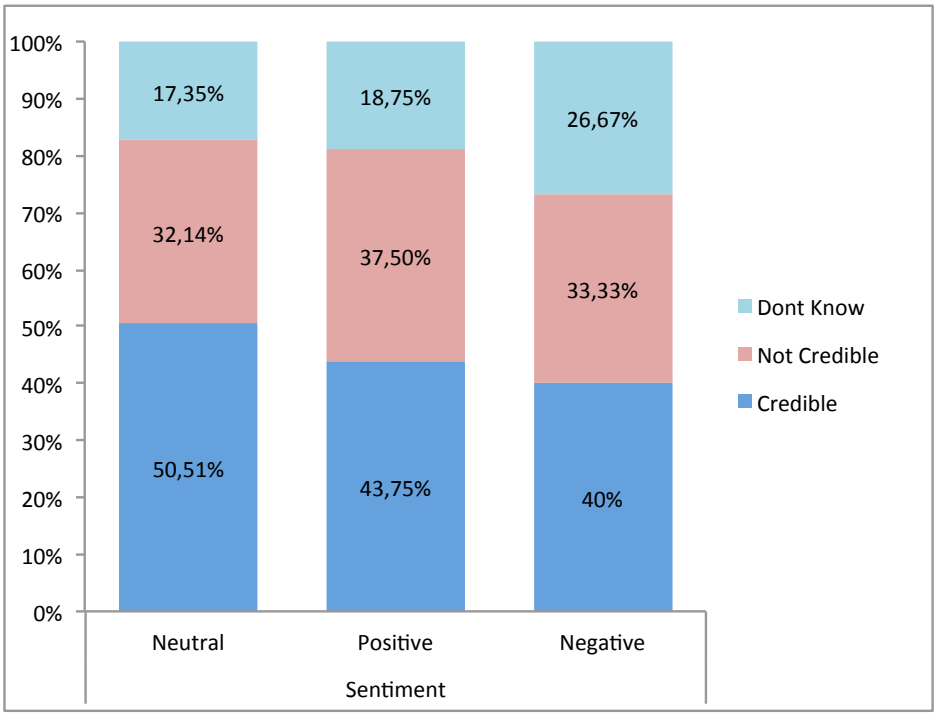

Figure 5.6: Sentiment and credibility ratings.

5.2.3 Feature Definition

One task for the user study participants was to provide textual feedback for the question "What made you choose credible / not credible?". The goal of this feedback was to identify features that can be automatically extracted from tweets.

We analyzed the received user feedback and separated the content into two categories, namely "Credible features" and "Not Credible features". We identified a total of 43 features for both categories. To narrow down the feature count we chose the features which overlapped with the Expert Users and the Regular Users. Additionally we extended the lists with features gathered in Section 5.1.1. The final identified feature count was 25 features.

We separated the features into categories that describe the origin of the feature value:

User Based Features (meta)

Features that can be read on the fly and are available from a tweet author. E.g. number of followers, number following users, number of listed count, author has a location defined.

User Based Features (attributed)

Features that refer to a tweet author and are computed based on available data points. E.g. account type (person or company), has real person name, has real location, profile picture type (graphic or photo).

Text Based Features (tags)

Features that are based on tags contained in a tweet. E.g. number of hashtags, number of mentions, number of links.

Text Based Features (words)

Features that describe the words contained in a tweet. E.g number of words, number of prepositions, number of upper case words, number of nouns.

5.3 Experiments and Feature Selection

The goal of the experiments was to find the best performing feature combinations of Section 5.2.3.

We worked with two different data sets to analyze and find the best performing feature combinations. First we used the data set of our user study containing 258 tweets and labels for "Credible", "Not Credible" and "Don't Know". We removed the tweets with the label "Don't Know" to create a binary classification problem. This resulted in a data set with a total of 210 tweets. 125 tweets (59.5%) with the label "Credible" and 85 tweets (40.5%) with the label "Not Credible".

The second data set was from the Verifying Multimedia Use Task [55] of the 2016 MediaEval Workshop[3]. The task addressed the problem of social media posts that contain misleading content (e.g. using photos in the wrong context). The aim of the task was to build and establish automatic ways to classify viral social media content propagating fake images or presenting real images in a false context [56]. The task was separated into "image-only", "text-only", and "hybrid" approaches. The data set contained 2,228 tweets with text, meta information and an attached media item (e.g. photo, video). 1,230 tweets (55.2%) were labeled as "fake" for including misleading content and 998 tweets (44.8%) were labeled as "real" for including correct content. During analysis of the data set we had to remove 265 tweets (11.9%) due to deleted or removed accounts. This reduced the data set to a size of 1,963 tweets (88.1%).

[3] `http://www.multimediaeval.org/mediaeval2016/`

We utilized the WEKA Data Mining Software[4] to employ various feature combinations and machine learning algorithms for classification. We selected the F1-score [13] as evaluation metric and conducted k-fold cross validations [57] with k=10 to receive the scores for our implemented approaches (BST, SND, THD).

	UserStudy data set			MediaEval data set		
Approach	F1	Precision	Recall	F1	Precision	Recall
BST	0.698	0.700	0.705	0.582	0.637	0.625
SND	0.691	0.691	0.695	0.578	0.623	0.618
THD	0.670	0.671	0.676	0.418	0.576	0.567

Table 5.7: Best approaches on user study data-set compared to the MediaEval data-set.

Table 5.7 shows that the best performing approaches on the user study data set perform low on the MediaEval data set. Hence the feature selection was to specific defined for the user study data set. We enhanced the feature selection for the MediaEval data set and evaluated the approaches against the proposed "text-only" approaches from the MediaEval workshop task.

Approach	F1	Precision	Recall
Linkmedia [58]	0.755	0.639	0.922
VMU [59]	0.738	0.587	0.995
MCG-ICT [60]	0.683	0.747	0.629
mBST	0.682	0.683	0.685
mSND	0.677	0.680	0.678
mTHD	0.668	0.675	0.676
MediaLab@DISI [61]	0.617	0.706	0.548

Table 5.8: Approaches on MediaEval data set.

[4] `https://www.cs.waikato.ac.nz/ml/weka`

Table 5.8 shows the results of our three best performing approaches *mBST*, *mSND*, and *mTHD* employed on the MediaEval data set. Additionally the table displays the results of the proposed approaches of the submitted working notes papers for comparison.

The following Table 5.9 displays our developed approaches and lists the contained features.

Approaches	
Name	Features
mBST	*real person name, profile picture type, number of hashtags + number of mentions, number of uppercase hashtags, number of links, number upper case words + number caps lock words.*
mSND	*number of following, number of followers, has location, number of hashtags + number of mentions, number of uppercase hashtags, number of links.*
mTHD	*real person name, profile picture type, number of hashtags + number of mentions, number of uppercase hashtags, number of links.*

Table 5.9: Approaches and implemented features

Chapter 6

Prototype

This chapter covers the design and implementation of the developed prototype. Section 6.1 outlines the conceptional design of the prototype and Section 6.2 covers the implementation details and the employed technologies.

6.1 Conceptional Design

The conceptional design begins with a use-case diagram and covers the individual system components with a component diagram. Next, a class diagram outlines the classes and methods used in the prototype. Sequence diagrams cover the important parts of the system functionalities and highlight the execution flow.

6.1.1 Use-Case Diagram

The following use-case diagram outlines the user interaction option with the prototype.

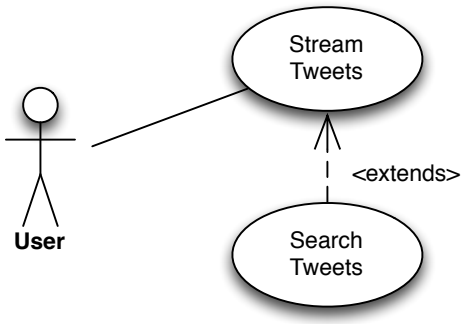

Figure 6.1: Use-case diagram

Use Case 1	**Stream Tweets**
Scope:	Client
Level:	User-goal
Primary Actor:	End-User
Preconditions:	• User has an active internet connection.

Basic flow:

1. The user defines a hashtag.
2. The user defines filter criteria for media type, sentiment and credibility.
3. The user invokes the message retrieval.
4. The system continously displayes retrieved messages based on the hashtag and filter criteria.

Extensions:

3.a Search Tweets

1. The user invokes a search.
2. The system returns a finite set of messages.

4.a No Messages

1. The user starts over with step 1.

4.b Stop Stream

1. The user stops the retrieval.

6.1.2 Component Diagram

The prototype has a client / server architecture with an attached service component that handles the main system tasks. This translates into six components separated within three packages: Client, Server, and Service.

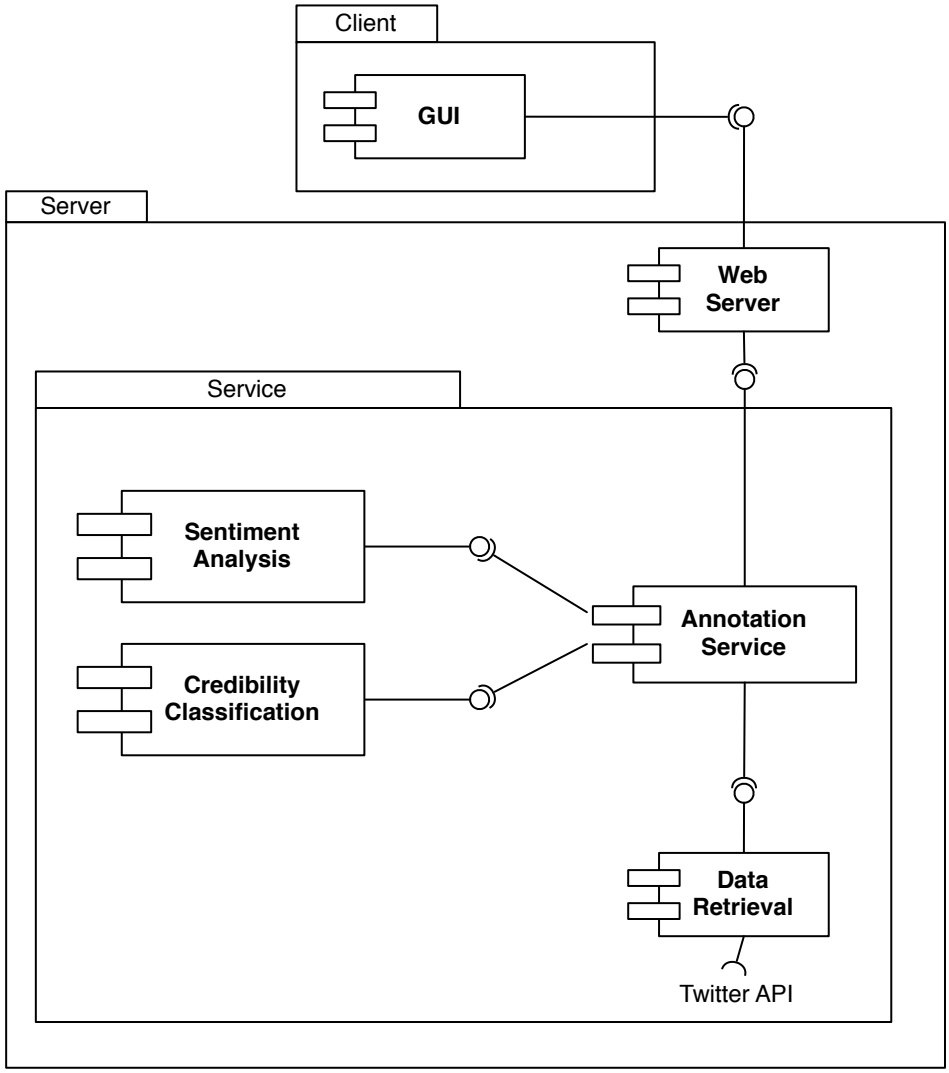

Figure 6.2: Component diagram of the prototype.

GUI: The graphical user interface connects to the web server and provides an interface for the user. The GUI displays the data returned from the web server.

Web Server: The web server provides an interface that returns data sets. It is connected to the annotation service which handles the retrieval and annotation of tweets.

Annotation Service: The annotation service is the interface between the main service components. It annotates the retrieved data from the data retrieval component and provides an interface for the web server.

Data Retrieval: The data retrieval component employs the Twitter API to retrieve tweets. It transforms the tweets into a defined model format that can be shared across the services.

Sentiment Analysis: The sentiment analysis component employs approaches to calculate the sentiment of a tweet.

Credibility Classification: The credibility classification component employs approaches to calculate the credibility value for a tweet.

6.1.3 Class Diagram

The following class diagram represents the packages *Server* and *Service* of the component diagram in Figure 6.2. However, due to the usage of a non object-oriented programming language, the class diagram does not fully comply with the UML[1] standard and is a schematic definition of the main artifacts. The individual boxes represent classes and functions. The title of a box describes the class name or function name and the attributes defined in the bodies of the boxes represent important variables or methods.

[1] `http://www.omg.org/spec/UML/`

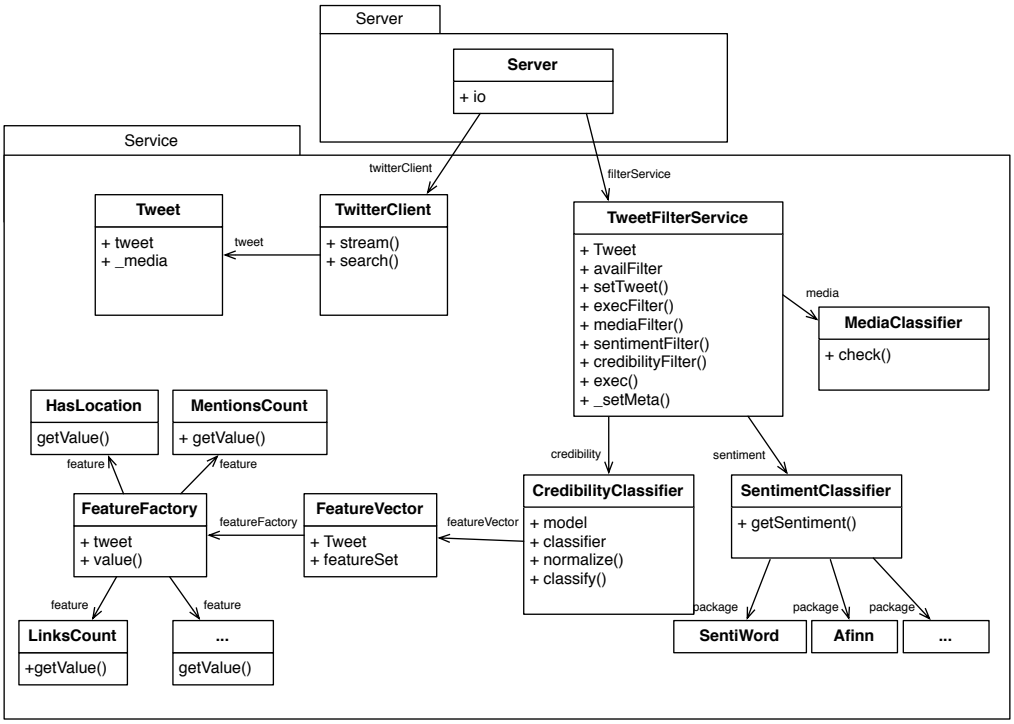

Figure 6.3: Class diagram of the server and service component.

Server: The server provides an endpoint for the user interface. It opens a socket connection and listens to events from the client to stream or search for tweets. The server instances a *TwitterClient* and calls a stream or a search method.

TwitterClient: The *TwitterClient* calls the Twitter API and provides two methods: (1) to stream for tweets or (2) to search for tweets. It returns tweets that match the query parameters as a *Tweet* model.

Tweet: A *Tweet* class represents a model of an incoming tweet from the Twitter API.

TweetFilterService: The *TweetFilterService* receives a tweet and employs different filter on it. If a sentiment filter is set, it instances a *Senti-*

mentClassifier which returns a sentiment value for the tweet. In a next step, it checks if the filter criteria value and sentiment value matches to return or to omit the tweet. The same applies for the *MediaClassifier* and the *CredibilityClassifier*. If a credibility filter criteria is set, it instances a *CredibilityClassifier*, runs a credibility classification and compares the value with the filter criteria value and returns the tweet if it matches.

MediaClassifier: The *MediaClassifier* returns the media type attached to a tweet.

SentimentClassifier: The *SentimentClassifier* returns a sentiment value for a text. It allows to employ different packages for sentiment analysis.

CredibilityClassifier: The *CredibilityClassifier* instances a *FeatureVector* for a tweet to generate feature values. It employs the defined model from the filter criteria and returns a credibility value.

FeatureVector: The *FeatureVector* class returns a feature vector that includes feature values for a tweet. Based on the defined feature set it instances a *FeatureFactory* to access the feature values and create the feature vector.

FeatureFactory: The *FeatureFactory* calculates the various feature values for a tweet. Based on a feature name it calls a new feature instance (e.g. LinksCount, HasLocation, ...) and returns the calculated value.

Feature: A *Feature* (e.g. LinksCount, HasLocation,...) calculates a feature value and returns the value to the *FeatureFactory*.

6.1.4 Sequence Diagrams

The consecutively listed diagrams show detailed sequences of the previously described use-case and the behavior of the components in the *Service* package (see Figure 6.2).

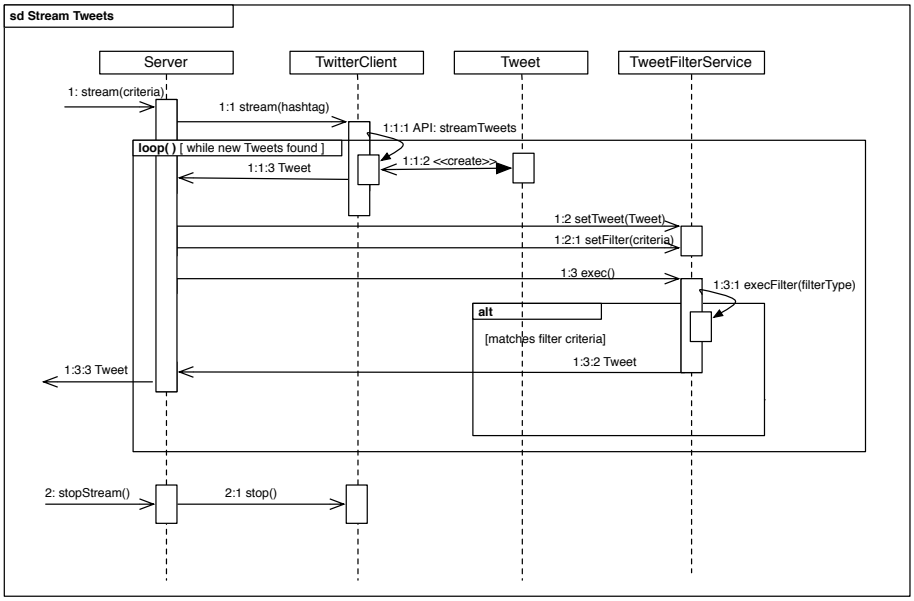

Figure 6.4: Sequence diagram of *Stream Tweets* use-case.

The sequence diagram in Figure 6.4 shows the behavior for the main use-case *Stream Tweets*. The sequence starts with the invocation of a stream and the server responses with tweets that match the user defined hashtag.

In (1:) the function *stream()* of *Server* is called with the user defined filter criteria and the hashtag. The *Server* extracts the hashtag and calls *stream(hashtag)* of *TwitterClient* (1:1). The *TwitterClient* calls the external Twitter API and starts the streaming process to find messages for the defined hashtag (1:1:1). A model for a tweet is generated for every matched tweet from the Twitter API (1:1:2) and returned to the *Server* (1:1:3). The *Server*

passes the *Tweet* to the *TweetFilterService* (1:2) and sets the filter criteria (1:2:1). The *Server* executes the method *exec()* (1:3) of *TweetFilterService* to employ the defined filter(s) on the *Tweet* model (1:3:1). If a tweet matches all filter criteria *TweetFilterService* returns an annotated *Tweet* model to the *Server* (1:3:2). The *Server* passes it to the requesting client (1:3:3). The process of streaming tweets to a client continues until the user defines to manually stop it (2:).

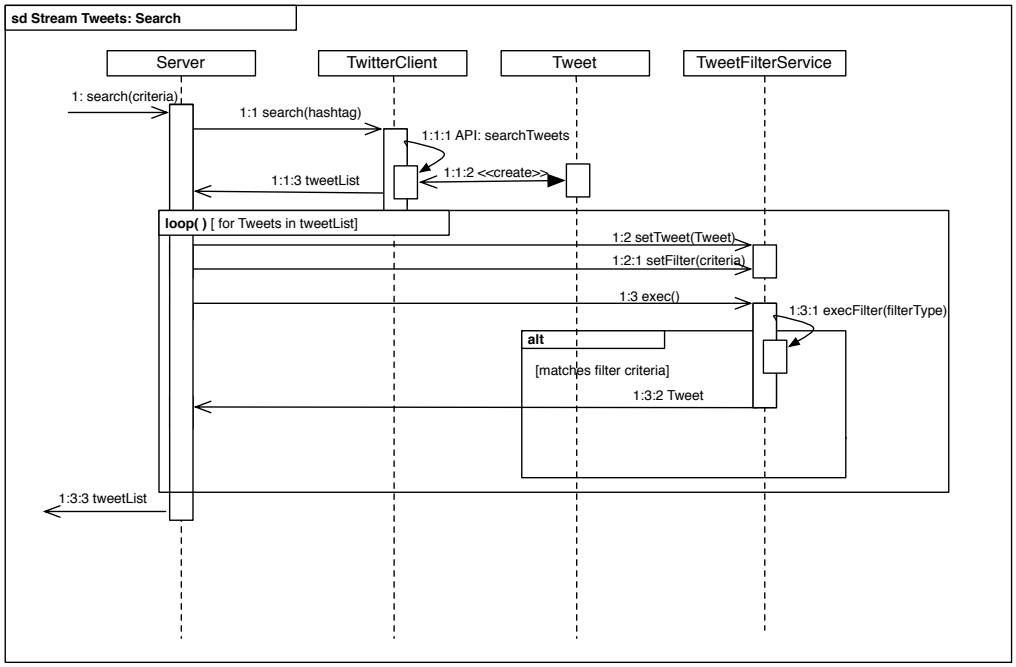

Figure 6.5: Sequence diagram of *Stream Tweets* use-case extended by *Search*.

The sequence diagram in Figure 6.5 shows the behavior for the use-case *Stream Tweets* and the extension *Search*. It is almost identical with the sequence diagram in Figure 6.4. The difference of the extension *Search* to *Stream Tweets* is in the handling of the incoming tweets. Rather than calling the external Twitter API with a stream request, the API call is a request

to search for a hashtag which returns a finite list of tweets (1:1:1). Every Tweet in the list gets passed to *TweetFilterService*. The processes following afterwards are the same as in Figure 6.4.

The following consecutively listed sequence diagrams employ filters and start with the function *exec()* at position (1:3) of the *Stream Tweets* use-case where the execution of the *TweetFilterService* logic begins.

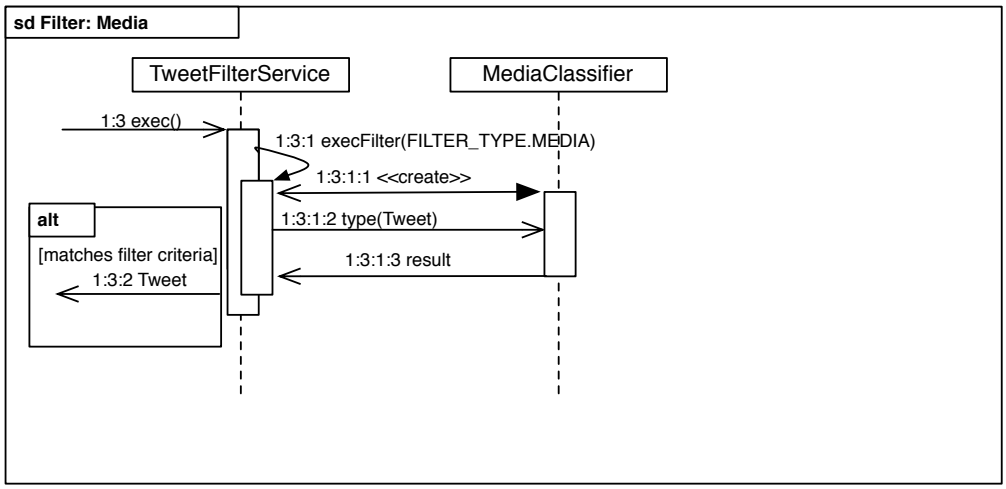

Figure 6.6: Sequence diagram of *Stream Tweets* with a media filter.

The sequence diagram in Figure 6.6 highlights the steps for the employment of the media filter on a Tweet.

TweetFilterService executes *execFilter()* (1:3:1) and instances a *MediaClassifier* (1:3:1:1). *TweetFilterService* calls the method *type()* (1:3:1:2) of *MediaClassifier* to request the attached media type of a Tweet. *MediaClassifier* identifies the attached media type and returns it to *TweetFilterService* (1:3:1:3). If it matches the filter criteria, the Tweet gets returned.

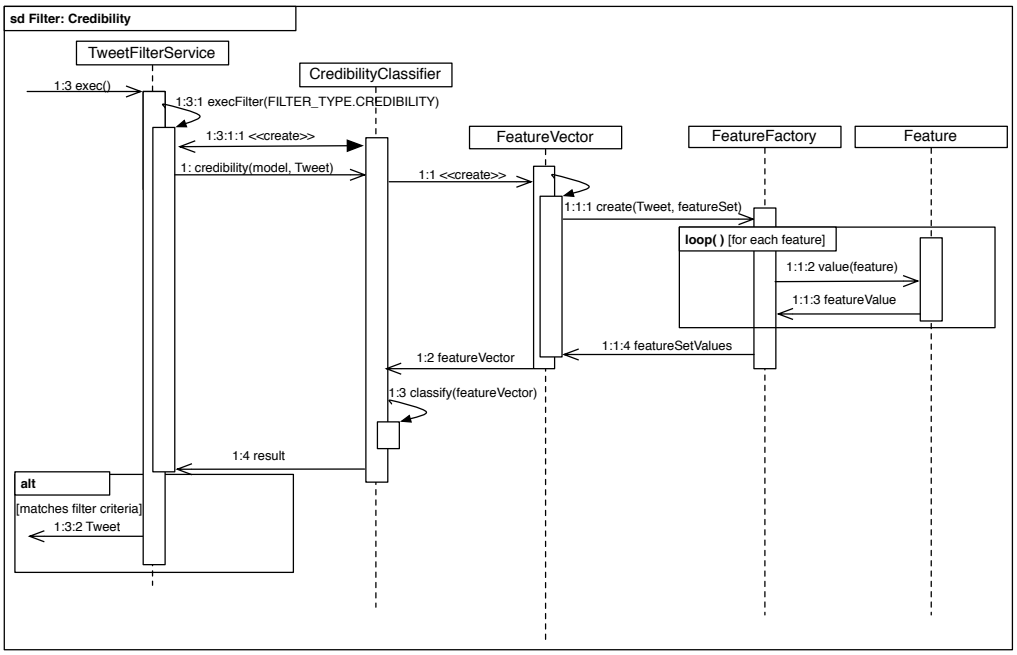

Figure 6.7: Sequence diagram of *Stream Tweets* with a credibility filter.

The sequence diagram in Figure 6.7 highlights the steps for the employment of the credibility filter on a Tweet.

TweetFilterService executes *execFilter()* (1:3:1) and instances a *CredibilityClassifier* (1:3:1:1). For the subsequently method calls the numbering starts with (1:) to ensure readability. The *TweetFilterService* calls the *credibility()* function to start the classification (1:).

The *CredibilityClassifier* instances a *FeatureVector* (1:1). The *FeatureVector* calculates the feature values for a Tweet with the help of a *FeatureFactory* (1:1:1). The *FeatureFactory* loops through a the defined feature set in the model, calculates a value for each feature (1:1:2) and returns the value (1:1:3). The *FeatureFactory* then returns a set of feature values to the *FeatureVector* (1:1:4) which gets passed on to the *CredibilityClassifier* (1:2). The *Credibil-*

ityClassifier executes the classify method based on the feature vector and a model. It then returns the classification result to the *TweetFilterService* (1:4) which matches the result with the filter criteria. If the match is true, it annotates the Tweet with the classification values and returns the Tweet (1:3:2).

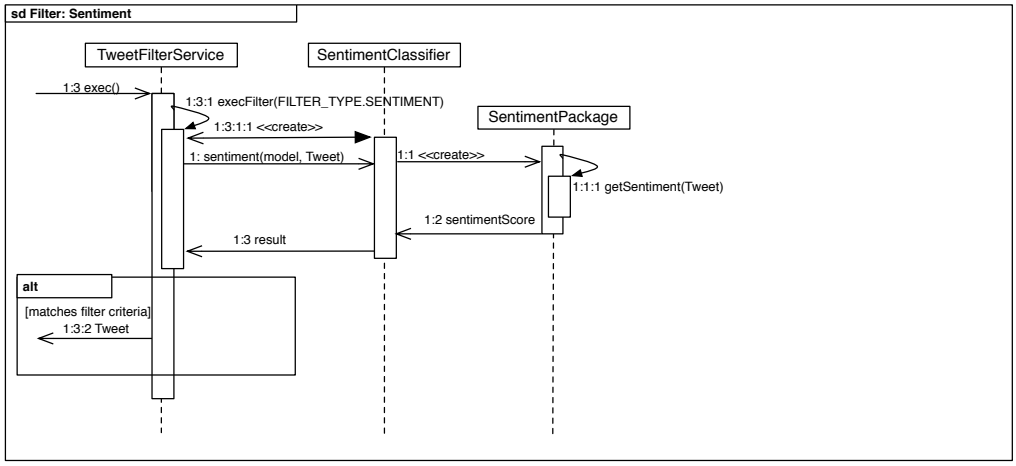

Figure 6.8: Sequence diagram of *Stream Tweets* with a sentiment filter.

The sequence diagram in Figure 6.8 highlights the steps for the employment of the sentiment filter on a Tweet.

The *TweetFilterService* instances a *SentimentClassifier* (1:3:1:1) and requests the sentiment for a Tweet (1:). The *SentimentClassifier* employs a sentiment package and calls *getSentiment()* for a Tweet (1:1:1). The *SentimentPackage* returns the score to the *SentimentClassifier* (1:2) which passes the result to the *TweetFilterService* (1:3). *TweetFilterService* matches the result with the filter criteria and if the match is true, it annotates the Tweet with sentiment scores and returns the Tweet (1:3:2).

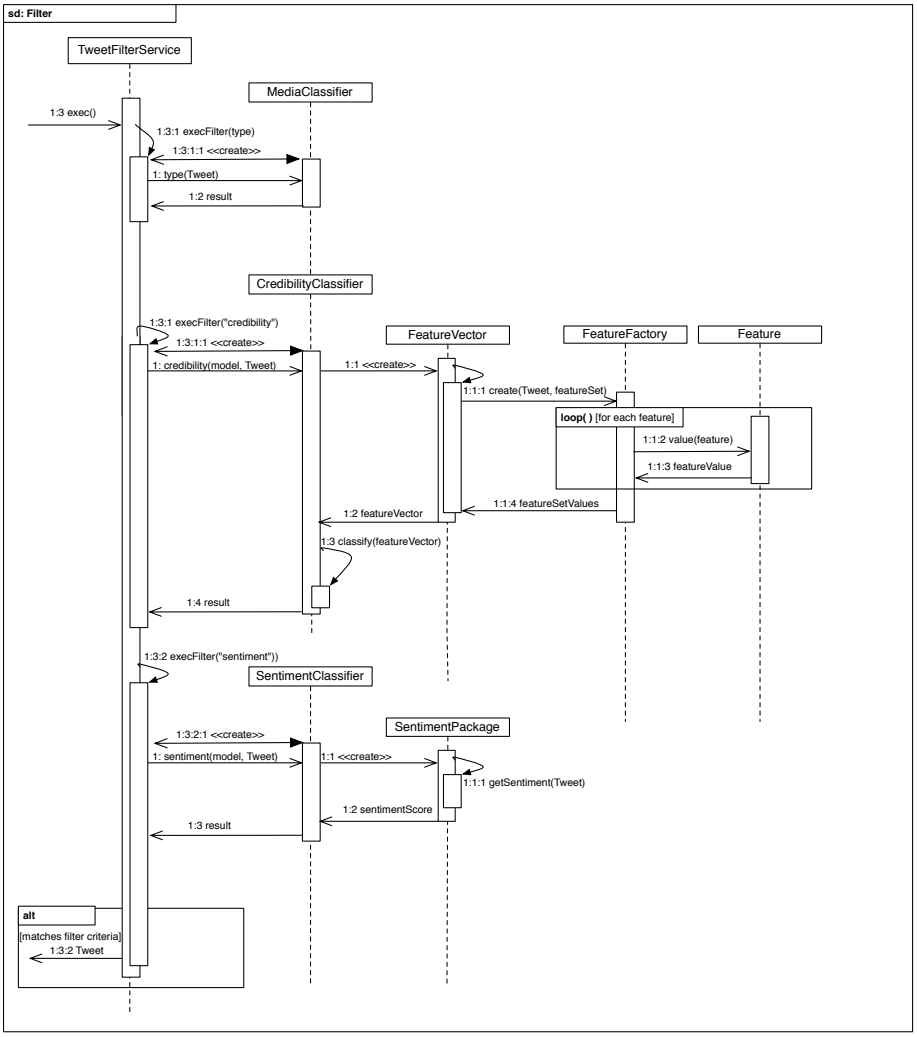

Figure 6.9: Sequence diagram of *Stream Tweets* with a media, a credibility and a sentiment filter.

The sequence diagram in Figure 6.9 shows the previously described filters media, credibility and sentiment employed in sequence. The filters are combined with a logical *AND* to omit a Tweet if one filter criteria does not match.

6.2 Implementation

This section covers the details of the implementation and the employed technologies. The prototype is separated in backend (server, service) and frontend (client). This allows to develop independent components that communicate through a standardized protocol. The major implementation part is in the backend component. This is the main part of the prototype where the logic is located and the sentiment analyis and credibility classification is executed.

We chose JavaScript[2] as programming language to unify the development experience. JavaScript is utilizable for the server implementation and for the client implementation.

6.2.1 Server Implementation

The server side uses Node.js[3] which provides the basis to work with JavaScript outside a web-browser. Node.js is an open source JavaScript runtime built on the JavaScript engine V8. The engine is developed by Google and is also used in the open source web-browser Google Chrome.

The server component provides a Socket.io[4] WebSocket interface that allows clients to subscribe to events or send events to communicate with the server. This allows a client to receive push messages rather than polling for new messages.

```
1 const server = Http.Server(app.callback());
2 const io = new SocketIO(server);
3
```

[2] https://developer.mozilla.org/bm/docs/Web/JavaScript
[3] https://nodejs.org/
[4] https://socket.io/

```
4  io.on("connection", client => {
5    const twitterClient = new TwitterClient();
6    client.on("stream_messages", data => {
7      LogUtils.log("received data", data);
8      twitterClient.stream(data.search, (err, tweet) => {
9        ...
10     });
11   });
12   client.on("search_messages", data => {
13     const matchedTweets = [];
14     LogUtils.log("received data", data);
15     twitterClient.search(data.search).then(tweets => {
16       ...
17     });
18   });
19 });
```

Listing 6.1: Server implementation

The code in Listing 6.1 creates a socket connection and handles incoming clients. When a client connects at line four, the server creates a new *TwitterClient* instance at line five. The server listens for events from the client and executes the desired functions. For example, if the event "stream_messages" arrives at line six, the server executes the stream function of *TwitterClient* at line eight, which starts a stream of tweets.

6.2.2 Service Implementation

The service component implements the functionality to filter for different criteria such as media type, sentiment and credibility. Each filter has a custom implementation and can be adapted or extended.

```
1  const filterService = new TweetFilterService();
2  const filterServiceCallback = (tweet, filter, cb) => {
3    filterService
4      .setTweet(tweet)
5      .execFilter(FILTER_TYPES.MEDIA, filter.media)
6      .execFilter(FILTER_TYPES.SENTIMENT, filter.sentiment)
7      .execFilter(FILTER_TYPES.CREDIBILITY, filter.credibility)
8      .run((err, resultTweet) => {
9        if (err) {
10         cb(err, null);
11       }
12       cb(null, resultTweet);
13     });
14 };
```

Listing 6.2: TweetFilterService

Listing 6.2 shows the code that is executed for every new incoming tweet. At line four a tweet is passed to a *TweetFilterService* instance. Line five to seven shows the register of three filters: Media, sentiment and credibility. The enum *FILTER_TYPE* defines the filter to register and the object *filter* contains the filter criteria value. The filters are connected in series with a logical *AND* to omit non-matching filters. Line eight executes the registered filters and returns an annotated tweet if all criteria match.

6.2.2.1 Media Filter

Based on the attached media entity of a tweet, the filter returns the string *photo* or *video* when compared to the filter criteria. A media filter critiera can be combinations of the values "photo" and "video" or "all".

6.2.2.2 Credibility Filter and Credibility Classification

The implementation of the credibility filter is independent of the employed credibility classifier. Our developed classifier includes models built with feature sets employed on the MediaEval data set (see Section 5.3).

```
1 function credibilityFilter(options) {
2   const clf = new CredibilityClassifier(options.model);
3   const prediction = clf.classify(this.tweet);
4   if (options.values.length === 0
5     || options.values.indexOf(prediction) > -1)
6       return true;
7   return false;
8 }
```

Listing 6.3: Function credibilifyFilter() of TweetFilterService

Listing 6.3 shows the function *credibilityFilter()* of the class *TweetFilterService*. At line two, a *CredibilityClassifier* instances with a model defined in the *options.model* parameter. At line three, the classifier executes the *classify()* function to classify the credibility of a tweet. Line five to seven checks if the prediction of the classification matches with the value in *options.values* which is the user defined filter criteria for credibility. Values can be combinations of "credible" and "not_credible" or "both".

```
1 function classify(tweet) {
2   const fv = FeatureVector(tweet, this.model.featureSet);
3   const clf = new this.classifier(this.model);
4   const prediction = clf.classify(fv);
5
6   return prediction;
7 }
```

Listing 6.4: Function classify() of CredibilityClassifier

Listing 6.4 shows the implementation of the *classify()* function of the class *CredibilityClassifier*. At line two, the *FeatureVector* function calculates a feature vector for a tweet. The returned feature vector gets passed to the *classify()* function on line four to classify the tweet and predict a value.

```
1  function FeatureVector(tweet, set=FEATURE_SETS.EXAMPLE) {
2    const feature = new FeatureFactory(tweet);
3    const allwc = feature.value('allWordsCount');
4    switch (set) {
5      case FEATURE_SETS.EXAMPLE:
6        return [
7        feature.value("followersCount"),
8        feature.value("ratioFollowerFollowing"),
9        feature.value("hasLocation"),
10       feature.value("hashtagsMentionsCount") / allwc,
11       feature.value("uppercaseHashtagsCount") / allwc,
12       feature.value("linksCount")
13       ];
14     default:
15       return [];
16     }
17 };
```

Listing 6.5: Function FeatureVector()

Listing 6.5 shows the *FeatureVector* function with a custom defined feature set. At line two, a *FeatureFactory* is instanced with a tweet. The *FeatureFactory* contains methods to calculate the individual feature values and is extendable. At line five, the feature set named *EXAMPLE* is created. *feature.value()* allows to access individual feature values and to create a feature vector in the form of an array.

6.2.2.3 Sentiment Filter and Sentiment Analysis

The implementation of the sentiment filter is independent of the employed sentiment analysis package.

```
1 function sentimentFilter(options) {
2   const sw = new SentimentClassifier(options.model);
3   const sentiment = sw.getSentiment(this.tweet.text);
4
5   if (options.values.length === 0
6   || options.values.indexOf(sentiment) > -1)
7     return true;
8   return false;
9 }
```

Listing 6.6: Function sentimentFilter() of TweetFilterService

Listing 6.6 shows the function *sentimentFilter()* of the class *TweetFilterService*. At line two, a *SentimentClassifier* instances with a model defined in the *options.model* parameter. At line three, the sentiment value of the tweet is returned by the function *getSentiment()*. Line five to seven checks if the sentiment value matches with the value in *options.values* which is the user defined filter criteria for sentiment. These values can be combinations of "positive", "negative", "neutral" or "all".

The *SentimentClassifier* employs two freely available sentiment analysis packages: AFINN-111 (see Section 3.1.2) and SentiWord[5]. *SentimentClassifier* is a wrapper for the employed sentiment analysis packages and provides a unified interface to retrieve the sentiment of a text. New sentiment analysis packages can simply be added to the *SentimentClassifier* class by extending the *getSentiment()* function with new packages.

[5] http://sentiwordnet.isti.cnr.it/

6.2.3 Client Implementation

The client is a web interface, implemented with HMTL5 and React.js[6]. React.js is an open source JavaScript library for building rich user interfaces. It allows to develop simple encapsulated components with further composition to create complex user interfaces. When data changes, React.js will efficiently update and render the designated components.

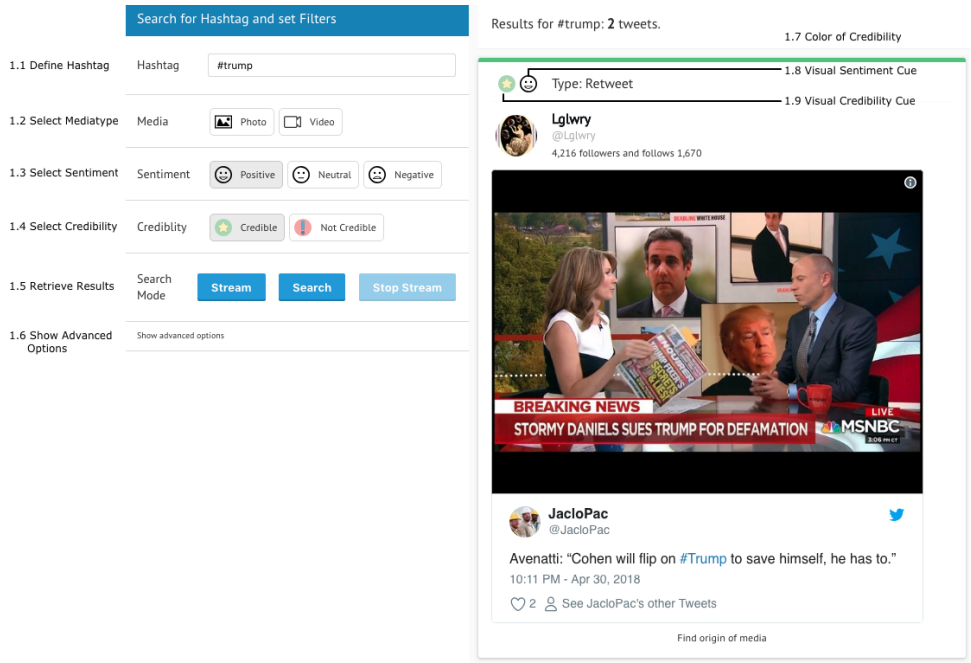

Figure 6.10: Prototype user interface with annotations.

Figure 6.10 shows the interface of the prototype. It consists of a sidebar area on the left side and a content area on the right side. The sidebar is fixed and does not scroll with the content. The sidebar includes the form to control the various filter and to initiate the message retrieval process.

[6] `https://reactjs.org/`

A number above the content sections highlights the result count of tweets for the defined hashtag and filter criteria. Tweets are shown as a list to create a familiar search result feeling with newly retrieved tweets attached on top of previous results.

The following list describes the annotations of Figure 6.10:

1.1 Define Hashtag: Tweets are retrieved based on the given hashtag.

1.2 Select Mediatype: The user can set the media type filter criteria to "Photo" and "Video".

1.3 Select Sentiment: The user can define a sentiment filter criteria and select "Positive", "Negative", and "Neutral".

1.4 Select Credibility: The user can define a credibility filter criteria and select "Credible" and "Not Credible".

1.5 Retrieve Results: The user can start a stream or a search based on the hashtag an the filter criteria.

1.6 Show Advanced Options: Experienced users can expand additional options to select a different crediblity model and a differen sentiment analysis package.

1.7 Color of Credibility: A tweet gets highlighted based on the crediblity value. Green stands for "Credible" and red for "Not Credible". The color matches the icons in 1.4.

1.8 Visual Sentiment Cue: Every tweet gets annotated with a visual cue that matches the selected sentiment filter criteria.

1.9 Visual Credibility Cue: Every tweet gets annotated with a visual cue that matches the selected credibility filter criteria.

6.3 Evaluation

The evaluation of the prototype is based on the "System Usability Scale" (SUS) by Brooke et al. [62]. SUS is a ten-item scale that provides a global view of subjective assessments of usability [62]. The goal was to find out how participants rate the usability of a search interface implementation with filter criteria for media, sentiment, and credibility. Regarding Sauro[63], the SUS score is a reliable metric even with a small sample size.

6.3.1 Setup

We recruited ten participants to evaluate the prototype developed in Chapter 6. We provided a link[7] to an online version of the prototype and attached a brief description of the system use case: "The system provides a method to search for information on Twitter based on a hashtag". Furthermore, we included a link to a trending hashtags site[8] to ensure that the participants work with hashtags that gain results.

Participants of a system evaluation with SUS are generally presented with the ten-item survey after they had an opportunity to use the system and without any further briefing on how to use the system [62]. Questions are evaluated on a 5-point Likert scale from 1 ("Strongly Disagree") to 5 ("Strongly Agree"). The following questions form a SUS survey:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.

[7] `http://eelit.workingtree.at`
[8] `https://trends24.in/united-states/`

4. I think that I would need the support of a technical person to be able to use this system.

5. I found the various functions in this system were well integrated.

6. I thought there was too much inconsistency in this system.

7. I would imagine that most people would learn to use this system very quickly.

8. I found the system very awkward (cumbersome) to use.

9. I felt very confident using the product.

10. I needed to learn a lot of things before I could get going with this system.

Regarding Lewis and Sauro [64], we altered question number 8 and replaced the word "cumbersome" with "awkward". Studies using SUS found that non-native English speakers had problems with the word "cumbersome" and asked for clarification of the word [64].

6.3.2 Results

SUS yields a single number ranging from 0 to 100 and represents a composite measure of the overall usability of the evaluated system [65]. Bangor, Kortum, and Miller [66] proposed scales to interpret the SUS score (see Figure 6.11).

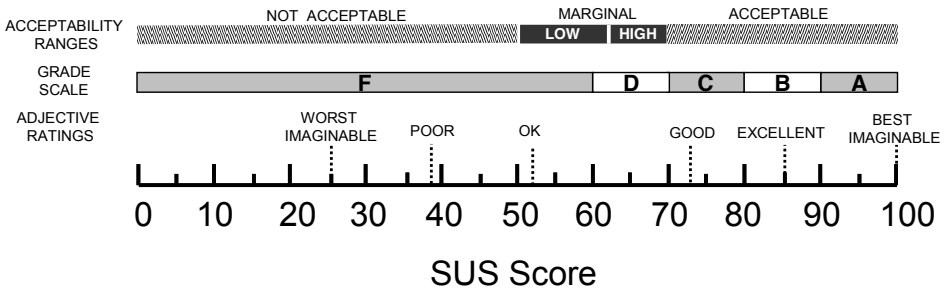

Figure 6.11: Acceptability scores, adjective ratings, and school grading scale in relation to the average SUS score [66].

The calculated SUS score for the prototype is 80. This translates into an adjective rating between "Good" and "Excellent", an acceptability range of "Acceptable", and a school grade of "B".

Some participants provided valuable feedback. One participant found that filtering should take place after the results appeared: *"After the results appeared I applied several filters but it seems it's necessary to set the filters in advance not afterwards ('onclick' change)"*. This is a valid point. Currently the default state of the system is to retrieve any message with an attached media type and to annotate each tweet with the respective sentiment and credibility value. The thinking in this case was to allow users to instantly search for a hashtag without setting any filter. Two participants mentioned the "Advanced Search Options". It was unclear for them what the options represented: *"After clicking "show advanced options" I can choose a package and a model but I don't know what those options mean or how it affects the search result"*, *"The difference Sentiword and Afinn need to be explained somewhere."*. This needs to be explained better or removed as too much options confuse users. One user stated, that he/she is currently not a Twitter

user, but found it useful to search for information with this prototype: *"I hardly ever use Twitter therefore I am not that familiar with hashtags. But I played around a bit and it was easy to use and helped searching information"*.

Overall, the prototype usability is in an acceptable range and further development based on the user feedback can be applied.

Chapter 7

Conclusion

7.1 Summary

Social media networks provide users the ability to share content and search for information. However, even faithful users can distribute wrong information unintentionally. To overcome this challenge, we developed a prototype that allows to search for Twitter messages and filter regarding credibility, sentiment and media type.

We derived knowledge from a user study and analyzed user feedback to identify features for credibility assessment. Experiments with different feature set combinations and test data allowed us to develop credibility models. An evaluation showed that the results of this models where aligned towards approaches that employ the same test data.

We implemented a prototype to demonstrate the developed models and an integration of sentiment analysis approaches. The focus of the proposed prototype was on a simple and extensible design to allow further development

and integration of new scientific knowledge. The separation of the prototype into a server/client structure allows to develop individual clients for different systems. An evaluation of the prototype shows that the participants are satisfied with the usability of the prototype.

7.2 Limitations and Future Work

We found that working with pre-compiled Twitter data sets can be challenging. In the course of time, tweets and user accounts can get removed from the Twitter platform and are therefore no longer accessible. The data set employed in our feature evaluation was reduced by 265 tweets (11.9%) from the original data set. Therefore a comparison with approaches using the original data set was limited. However, the magnitude of the limitation on the final results is tolerable.

The prototype is limited to the free Twitter API and only handles an input with hashtags. Another limitation is the non existence of a Twitter login for the prototype. Such a login allows individual users to use the prototype with their own credentials.

The following ideas beside the already stated from the feedback could be implemented in future work: Currently, the prototype works on desktop browsers only and does not adapt to smaller screens (e.g. mobile devices). A responsive design implementation is encouraged to allow mobile usage of the prototype. A filter for additional media types could be implemented to allow a search not only for photos and videos but documents of different types. Implementing methods to highlight (spam) bots would add a broader use case to the prototype.

List of Figures

List of Tables

Listings

Bibliography

[1] N. B. Ellison *et al.*, "Social network sites: Definition, history, and scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[2] M.-A. Abbasi and H. Liu, "Measuring user credibility in social media," in *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. Springer, 2013, pp. 441–448.

[3] A. Zubiaga, M. Liakata, R. Procter, K. Bontcheva, and P. Tolmie, "Towards detecting rumours in social media," *arXiv preprint arXiv:1504.04712*, 2015.

[4] B. News, "Reddit apologises for online boston 'witch hunt'," 2013, accessed: 2016-12-09. [Online]. Available: http://www.bbc.com/news/technology-22263020

[5] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proceedings of the 26th Annual Computer Security Applications Conference*. ACM, 2010, pp. 1–9.

[6] B. Kollanyi, P. N. Howard, and S. C. Woolley, "Bots and automation over twitter during the first us presidential debate," COMPROP Data Memo, Tech. Rep., 2016.

[7] D. Jurafsky and J. H. Martin, *Speech and language processing*. Pearson, 2014.

[8] J. F. Allen, "Natural language processing," in *Encyclopedia of Computer Science*. Chichester, UK: John Wiley and Sons Ltd., pp. 1218–1222.

[9] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT Press, 1999, vol. 999.

[10] N. Indurkhya and F. J. Damerau, *Handbook of natural language processing*. CRC Press, 2010, vol. 2.

[11] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: The penn treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[12] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 417–424.

[13] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[14] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[15] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[16] M. De Mey, *The cognitive paradigm*. University of Chicago Press, 1992.

[17] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[18] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752. Citeseer, 1998, pp. 41–48.

[19] T. Joachims, "Making large scale svm learning practical," Universität Dortmund, Tech. Rep., 1999.

[20] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *IJCAI-99 workshop on machine learning for information filtering*, vol. 1, 1999, pp. 61–67.

[21] M. Seeger, "Learning with labeled and unlabeled data," Tech. Rep., 2000.

[22] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," *Mahway: Lawrence Erlbaum Associates*, vol. 71, p. 2001, 2001.

[23] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining." in *LREC*, vol. 10, 2010, pp. 2200–2204.

[24] M. M. Bradley and P. J. Lang, "Affective norms for english words (anew): Instruction manual and affective ratings," Technical report C-1, the center for research in psychophysiology, University of Florida, Tech. Rep., 1999.

[25] F. Å. Nielsen, "A new anew: Evaluation of a word list for sentiment analysis in microblogs," *arXiv preprint arXiv:1103.2903*, 2011.

[26] X. Zhu, "Semi-supervised learning," in *Encyclopedia of machine learning*. Springer, 2011, pp. 892–897.

[27] B. Liu, "Sentiment analysis and opinion mining," *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[28] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[29] M. Ganapathibhotla and B. Liu, "Mining opinions in comparative sentences," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 241–248.

[30] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 675–684.

[31] B. Fogg and H. Tseng, "The elements of computer credibility," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 1999, pp. 80–87.

[32] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, and F. Menczer, "Detecting and tracking political abuse in social media." 2011.

[33] M. R. Morris, S. Counts, A. Roseway, A. Hoff, and J. Schwarz, "Tweeting is believing?: understanding microblog credibility perceptions," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, 2012, pp. 441–450.

[34] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *arXiv preprint arXiv:1407.5225*, 2014.

[35] A. Bessi and E. Ferrara, "Social bots distort the 2016 us presidential election online discussion," *First Monday*, vol. 21, no. 11, 2016.

[36] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, p. 12, 2009.

[37] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle," in *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, 2012, pp. 115–120.

[38] G. Paltoglou and M. Thelwall, "Twitter, myspace, digg: Unsupervised sentiment analysis in social media," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 4, p. 66, 2012.

[39] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010.

[40] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *LREc*, vol. 10, 2010, pp. 1320–1326.

[41] G. Paltoglou, M. Thelwall, and K. Buckley, "Online textual communications annotated with grades of emotion strength," in *Proceedings of the 3rd International Workshop of Emotion: Corpora for research on Emotion and Affect*, 2010, pp. 25–31.

[42] M. Thelwall and D. Wilkinson, "Public dialogs in social network sites: What is their purpose?" *Journal of the American Society for Information Science and Technology*, vol. 61, no. 2, pp. 392–404, 2010.

[43] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, 2005, pp. 347–354.

[44] C. Biever, "Twitter mood maps reveal emotional states of america," *New Scientist*, vol. 207, no. 2771, p. 14, 2010.

[45] S. Sikdar, B. Kang, J. ODonovan, T. Höllerer, and S. Adah, "Understanding information credibility on twitter," in *Social Computing (SocialCom), 2013 International Conference on*. IEEE, 2013, pp. 19–24.

[46] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, "Tweetcred: Real-time credibility assessment of content on twitter," in *International Conference on Social Informatics*. Springer, 2014, pp. 228–243.

[47] N. Diakopoulos, M. De Choudhury, and M. Naaman, "Finding and assessing social media information sources in the context of journalism," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 2451–2460.

[48] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[49] D. Metzler and W. B. Croft, "Linear feature-based models for information retrieval," *Information Retrieval*, vol. 10, no. 3, pp. 257–274, 2007.

[50] J. Xu and H. Li, "Adarank: a boosting algorithm for information retrieval," in *Proceedings of the 30th annual international ACM SI-*

GIR conference on Research and development in information retrieval. ACM, 2007, pp. 391–398.

[51] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *Journal of machine learning research*, vol. 4, no. Nov, pp. 933–969, 2003.

[52] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 133–142.

[53] M. Hao, C. Rohrdantz, H. Janetzko, U. Dayal, D. A. Keim, L.-E. Haug, and M.-C. Hsu, "Visual sentiment analysis on twitter data streams," in *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*. IEEE, 2011, pp. 277–278.

[54] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "Twitinfo: aggregating and visualizing microblogs for event exploration," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2011, pp. 227–236.

[55] C. Boididou, S. Papadopoulos, D. Dang-Nguyen, G. Boato, M. Riegler, S. E. Middleton, A. Petlund, and Y. Kompatsiaris, "Verifying multimedia use at mediaeval 2016," in *Working Notes Proceedings of the MediaEval 2016 Workshop, Hilversum, The Netherlands, October 20-21, 2016.*, 2016.

[56] "The 2016 verifying multimedia use," 2016, accessed: 2017-09-07. [Online]. Available: http://www.multimediaeval.org/mediaeval2016/verifyingmultimediause/

[57] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," in *Encyclopedia of database systems*. Springer, 2009, pp. 532–538.

[58] C. Maigrot, V. Claveau, E. Kijak, and R. Sicre, "Mediaeval 2016: A multimodal system for the verifying multimedia use task," in *Working Notes Proceedings of the MediaEval 2016 Workshop, Hilversum, The Netherlands, October 20-21, 2016.*, 2016.

[59] C. Boididou, S. Papadopoulos, S. E. Middleton, D. Dang-Nguyen, M. Riegler, A. Petlund, and Y. Kompatsiaris, "The VMU participation @ verifying multimedia use 2016," in *Working Notes Proceedings of the MediaEval 2016 Workshop, Hilversum, The Netherlands, October 20-21, 2016.*, 2016.

[60] J. Cao, Z. Jin, and Y. Zhang, "MCG-ICT at mediaeval 2016 verifying tweets from both text and visual content," in *Working Notes Proceedings of the MediaEval 2016 Workshop, Hilversum, The Netherlands, October 20-21, 2016.*, 2016.

[61] Q. Phan, A. Budroni, C. Pasquini, and F. G. B. D. Natale, "A hybrid approach for multimedia use verification," in *Working Notes Proceedings of the MediaEval 2016 Workshop, Hilversum, The Netherlands, October 20-21, 2016.*, 2016.

[62] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

[63] "10 things to know about the system usability scale (sus)," 2013, accessed: 2018-05-13. [Online]. Available: http://www.measuringu.com/blog/10-things-SUS.php

[64] J. R. Lewis and J. Sauro, "The factor structure of the system usability scale," in *International conference on human centered design*. Springer, 2009, pp. 94–103.

[65] M. Suominen, "Evaluating usability in video conferencing service in metso," 2013.

[66] A. Bangor, P. Kortum, and J. Miller, "Determining what individual sus scores mean: Adding an adjective rating scale," *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, 2009.

Appendix A

Zusammenfassung

Social Media Plattformen ermöglichen es ihren Nutzern Informationen zu verschiedenen Themen zu suchen und veröffentlichen. Der Inhalt solcher Informationen kann aus Fakten, Statemens und Meinungen mit unterschiedlichstem Emotionsgehalt bestehen sowie mit diversen Medientypen untermauert sein. Auf den Plattformen fehlt größtenteils die Verfizierung der Benutzerdaten, was ein anonymes erstellen und verteilen von Inhalten ermöglicht. Während Anonymität in Sachen Meinungsfreiheit wichtig ist, kann sie missbraucht werden um absichtlich falsche Inhalte zu verbreiten. Für den durchschnittlichen Nutzer kann es schwierig werden, zwischen falschen und korrekten Informationen zu unterscheiden. Im Rahmen dieser Arbeit haben wir eine Nutzerstudie durchgeführt, um die Glaubwürdigkeitswahrnehmung von Twitter-Nachrichten herauszufinden. Wir identifizierten Merkmale für die Glaubwürdigkeitsbewertung und bauten Modelle, um die Glaubwürdigkeit solcher Nachrichten automatisch zu klassifizieren. Um diese Klassifizierung zu demonstrieren haben wir einen Prototype entwickelt, der das Suchen nach Twitter-Nachrichten sowie filtern nach Glaubwürdigkeit, Emotionsgehalt und Medientyp erlaubt. Das System sucht nach Nachrichten, stellt sie

dar und fügt visuelle Hinweise für berechnete Emotions- und Glaubwürdigkeitswerte hinzu. Eine Bewertung des Prototyps hinsichtlich der Nutzbarkeit zeigt eine Akzeptanz innerhalb der Teilnehmergruppe. Ferner erlaubt das modular aufgebaute System eine Weiterverarbeitung und Einbeziehung von neuen wissenschaftlichen Erkenntnissen.

Appendix B

Prototype User Guide

B.1 Requirements

The minimum requirements to run the prototype locally are Node.js[1] (includes npm[2]), git[3] and working API keys for Twitter API and GateCloud[4].

B.2 Installation

Use your operation system specific terminal to access the system console (e.g. Terminal on OS X) and run the following commands.

```
1 git clone https://gitlab.com/simbra/thesis-application.git
```

Listing B.1: Prototype download from the repository into /thesis-application

[1] https://nodejs.ors
[2] https://npmjs.com
[3] https://git-scm.com/
[4] https://cloud.gate.ac.uk/

The next command can take some time as it loads 3rd party libraries from the npm registry.

```
cd thesis-application && npm install
```

Listing B.2: Install packages

Adapt the config files and add the api keys for the Twitter API and GateCloud service (rename config files in the config directory and remove ".sample" postifx).

```
cd server/src/config
```

Listing B.3: Config directory

```
npm run dev:server
```

Listing B.4: Start local server instance

The console should output: *"[INFO] Listening on *:3000"*.

Open a new terminal window and navigate into the folder *"/thesis-application"*.

```
npm run dev:web
```

Listing B.5: Start local web interface

The frontend is now accessible at `http://localhost:8080`.

A working prototype can be found at: `http://eelit.workingtree.at`

Appendix C

SUS Survey Results

The following page displays the individual ratings of the participating users from the prototype evaluation survey. To calculate the SUS score a Google Spreadsheet document template was used.

The template was found on: `https://community.uxmastery.com/t/system-usability-scale-calculator/3347`. The credit goes to `https://community.uxmastery.com/u/annabelle_andre` for implementing it.

SUS Scoring Sheet & Reliability Test

SUS data (scored 1=Strongly disagree; 5=Strongly agree)

Average score is 68

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Participants	I think that I would like to use this system frequently	I found the system unnecessarily complex	I thought the system was easy to use	I think that I would need the support of a technical person to be able to use this system	I found the various functions in this system were well integrated	I thought there was too much inconsistency in this system	I would imagine that most people would learn this system very quickly	I found this system very akward to use	I felt very confident using the system	I needed to learn a lot of things before I could get going with this system
1	4	2	4	2	4	2	4	2	4	2
2	4	2	4	2	5	2	5	1	5	1
3	4	1	5	2	4	1	5	1	5	1
4	3	2	5	2	3	2	5	1	5	1
5	3	2	5	3	4	2	4	2	4	3
6	3	2	5	2	5	2	4	2	4	3
7	4	2	4	1	4	1	4	1	5	2
8	4	2	5	1	4	1	4	1	4	2
9	3	2	4	2	5	2	4	2	3	2
10	4	1	4	2	3	2	4	2	3	1

Scales			
Odd items	Even items	SUS score (/100)	Grades
15	15	75	C
18	17	87.5	B
18	19	92.5	A
16	17	82.5	B
15	13	70	C
16	14	75	C
16	18	85	B
16	18	85	B
14	15	72.5	C
13	17	75	C
	Average scor	80	B

Grading SUS Key	
92	Best imaginable
85	Excellent
72	Good
52	OK/Fair
38	Poor
25	Worst imaginable
90-100	A
80-89	B
70-79	C
60-69	D
Less than 60	F