PhD Dissertation

DISI - University of Trento
International Doctorate School in Information and Communication Technologies

LARGE SCALE AGGREGATED SENTIMENT ANALYTICS

Mikalai Tsytsarau

Advisor:
Prof. Themis Palpanas
Università degli Studi di Trento

April 2013

Abstract

In the past years we have witnessed Sentiment Analytics becoming increasingly popular topic in Information Retrieval, which has established itself as a promising direction of research. With the rapid growth of the user-generated content represented in blogs, forums, social networks and micro-blogs, it became a useful tool for social studies, market analysis and reputation management, since it made possible capturing sentiments and opinions at a large scale and with the ever-growing precision. Sentiment Analytics came a long way from product review mining to full-fledged multi-dimensional analysis of social sentiment, exposing people attitude towards any topic aggregated along different dimensions, such as time and demographics.

The novelty of our work is that it approaches Sentiment Analytics from the perspective of Data Mining, addressing some important problems which fall out of the scope of Opinion Mining. We develop a framework for Large Scale Aggregated Sentiment Analytics, which allows to capture and quantify important changes in aggregated sentiments or in their dynamics, evaluate demographical aspects of these changes, and explain the underlying events and mechanisms which drive them. The first component of our framework is Contradiction Analysis, which studies diverse opinions and their interaction, and allows tracking the quality of aggregated sentiment or detecting interesting sentiment differences. Targeting large scale applications, we develop a sentiment contradiction measure based on the statistical properties of sentiment and allowing efficient computation from aggregated sentiments. Another important component of our framework addresses the problem of monitoring and explaining temporal sentiment variations. Along this direction, we propose novel time series correlation methods tailored specifically for large scale analysis of sentiments aggregated over users demographics. Our methods help to identify interesting correlation patterns between demographic groups and thus better understand the demographical aspect of sentiment dynamics. We bring another interesting dimension to the problem of sentiment evolution by studying the joint dynamics of sentiments and news, uncovering the importance of news events and assessing their impact on sentiments. We propose a novel and universal way of modeling different media and their dynamics, which aims to describe the information propagation in news- and social media. Finally, we propose and evaluate an updateable method of sentiment aggregation and retrieval, which preserves important properties of aggregated sentiments and also supports scalability and performance requirements of our applications.

Keywords
Sentiment Analysis, Contradiction Analysis, Dynamics Analysis

Acknowledgements

It is commonly acknowledged that novelty, usefulness and timeliness are the most valued factors of any research. It takes a deep experience to understand the novelty of a problem, a good imagination to see its usefulness and an incredible intuition to predict when it becomes necessary. But above all, it takes a good luck for all these qualities to meet in one person. In this regard I was very lucky to meet my advisor, Themis Palpanas, who first envisioned the idea of diverse sentiment aggregation, and then helped to develop it into a full-fledged analysis of aggregated sentiment. I am especially honored being his first doctoral student, and I would like to thank him for a careful guidance and wise management, and also for a great attention to my scientific endeavors.

My research took off from the problem of sentiment contradiction analysis and evolved to include the problems of demographical and dynamical analysis of aggregated sentiments, inspired by collaborations with the leading researchers in this area. It was my pleasure to work on the demographical sentiments problem with Sihem Amer-Yahia, who suggested many interesting ideas and motivated my research a lot. I am especially grateful for her confidence in usefulness of this research and in our methods. The analysis of sentiments dynamics took a start from the collaboration with Malu Castellanos and Meichun Hsu, and also led to a patent application with Umeshwar Dayal. I would like to thank them for very useful discussions that helped to shape this part of my work and understand its value. Finally, my thanks are due for Aris Gionis and Francesco Bonchi, whom I worked with on the diversity of itemsets. All these collaborations gave me an invaluable experience and I hope that they will continue in the future.

I would also like to thank my friends and colleagues for all their numerous comments and suggestions, that helped to improve my dissertation: Maksim Khadkevich, Siarhei Bykau, Yury Audzevich, Evgeny Stepanov, and everyone I know. My special thanks are to the members of dbTrento group, who were eagerly interested in my work and asked hard questions during our seminars and meetings: Yannis Velegrakis, Oleksiy Chayka, Alice Marascu, Katya Mirylenka, Davide, Dimitra, Kostas, Matteo, Michele and Pavlos.

I would like to complete my acknowledgements by thanking my dear family for their biggest support during my study. Especially I would like to thank my sister Alex, whom I shared with joys and hardships of student life and whose endless energy aspired me all these years to new endeavors. Finally, I thank her little cat Lucy for very active participation in proof-reading of this dissertation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

During the recent years we have been witnessing the Internet becoming an open platform, where people can express their opinions and can be heard. There are many services that allow people to publish information and opinions, such as blogs, wikis, forums, social networks and others. They all represent a rich source of opinionated information on different topics, which can be analyzed and exploited in various applications and contexts. It is therefore not surprising that we witness an increasing interest in the processing and analysis of unstructured data, with a special focus on Web text data. The wealth of information on the Web makes this endeavor not only rewarding in terms of newly produced knowledge, but also necessary, in order to exploit all this available information.

Sentiment Analysis can be used to learn about a customer's attitude to a product or its features, or to reveal people's reaction to some event or their political preferences. Opinions expressed by users are an important factor taken into consideration by product vendors, policy makers and stock analysts. The analysis of sentiments brings significant economic effects to various businesses and therefore we also observe a booming of services that analyze public sentiment on-demand. We believe that the interest in mining Web data would only continue to grow, as new sources of such data emerge and attract more attention from businesses and researchers alike.

However, with the proliferation of social web platforms, where millions of users provide opinions on a wide variety of content, the scale of the problem has also increased and became an obstacle to traditional sentiment mining, aggregation and retrieval methods. Nevertheless, the need to provide fine-grained analytics of social data is growing. Readily available users' demographics along with opinion data constitute a gold mine for extracting insights on what a particular user group thinks and how their opinion evolves over time and compares to opinions of others. This problem demands efficient and scalable methods for sentiment aggregation and correlation, which account for the evolution of sentiment values, sentiment bias, and other factors associated with the special characteristics of web data.

1.1 Thesis Context and Problems

People provide a huge amount of personal experiences and opinions towards different topics in the Web: In reviews, they express their opinions and experiences with a specific product or service. In blog postings, a mixture of information, arguments, and opinions can be found. Forum postings can be seen as online discussions and exchange platform of arguments. Microblog messages bear a reaction of people towards various events and personal experiences. All these expressed opinions are interesting for different applications. Consider the following scenarios:

1. Given a specific medical treatment, a physician might be interested in positive and negative experiences or opinions with respect to this treatment. But, not only a quantitative impression is useful to him, also how the opinion changed over time and which arguments were used in favor or against a specific treatment. Another example of contrasting sentiments comes from political topics, which are extensively discussed in the Web. Being aware of the opinions expressed towards these topics might be crucial to come up with the correct prediction of the impact and support for certain policies or decisions.

2. In some cases we want to detect where opinion of a group of people deviates from the general population or from that of another group with different demographical attributes. For instance, when computer scientists like *sci-fi* movies more than people working in Finance. Another type of biases that we are interested in considers behavioral differences of demographical groups, e.g. when people living in Italy think differently about their local events compared to people from Europe in general.

3. Changes in community's opinion are usually driven by new evidence or by impacting events coming from news sources. However, these are often not mentioned explicitly in texts, and to recognize the cause of sentiment changes we want to navigate to a correlated news trend and analyze the volume of news around a sentiment change. Then, by observing the dynamics of social media and their delayed reaction, we want to predict these changes as soon as we are able to recognize the establishing news trend.

Evidently, such problems require analyzing significant amounts of data to produce a desired output, and special methods that can exploit this volume to improve the resolution and representativeness of sentiment analysis. Conventional sentiment aggregation methods may be inefficient for large scale analytics, especially when subsequent time intervals contain different amount of sentiments and when a simple average of diverse sentiment values is taken. The information about real sentiment values can even be lost, when aggregating opposite values.

The aforementioned problems also require efficient sequential time series access methods with a possibility of hierarchical navigation over time. However, the databases commonly used for sentiment storage and access, are not optimized to track the evolution of sentiments on a large scale or to support fast update rates.

1.2 Thesis Structure and Solutions

The objective of our research is to exploit the current work on sentiment analysis and opinion mining, and extend the state of the art in detecting, explaining and predicting sentiment contradictions. Our additional goal is to provide efficient and scalable sentiment aggregation and storage methods which serve as a basis for the proposed solutions.

This thesis is structured as follows. In Chapter 2 we discuss the related work in Subjectivity Analysis, various trends and directions of research in this area, and identify different problems, which help establishing the scope and aims of the present work. In Chapter 3 we give an overview the particular problems we address and their interaction within the proposed framework. We also outline the most important methods, necessary for tackling the identified problems, and analyze their requirements. Following this, Chapter 4 presents our approach for sentiment aggregation and storage, which supports the outlined requirements and serves as a basis for the solutions we list below.

In Chapter 5, we are focusing on the novel problem of *Contradiction Analysis*, which aims at finding sentiment-based contradictions and opinion shifts at a large scale. First, we give a thorough definition of contradictions, backed up by the analysis of related literature. Second, we evaluate typical sentiment distributions and the human perception of their diversity based on user experiments. Third, we introduce a novel measure of contradiction, which is based on our observations. We study its properties related to our definitions and real data. Finally, we propose a scalable and accurate method for identifying contradictions at different time scales and evaluate its performance using synthetic and real-world datasets, as well as a user-study.

Chapter 6 discusses *Demographics Analysis* of extracted contradictions and concentrates on the detection of correlated user behavior. We propose a scalable approach for correlation detection among various demographic groups, which is suitable for the analysis of their biases and behavioral similarity. The novelty of our method is that it achieves very high efficiency by compressing the top-k correlations, without significantly affecting the quality of the results. We test our approach on both synthetic and real datasets, proving its efficiency and effectiveness.

In Chapter 7, we formulate a problem of identifying news events that cause dramatic changes of sentiments and their *Dynamics Analysis*. We propose a novel framework for a complex news- and social media modeling, which is capable of detecting interesting features of events by observing a time series of news articles publications, search interest or social response, and then correlating these data with time series of sentiment changes detected by various interestingness functions. We also study the differences in response dynamics between various media.

We conclude in Chapter 8, where we discuss the achievements of the present work and interesting findings, and outline future directions of research.

1.3 Innovative Aspects

This work addresses a novel large-scale sentiment analytics problem, focusing on the efficient aggregation of sentiments, detection and explanation of sentiment contradictions and opinion shifts, and computation of significant sentiment correlations for different demographic groups within dynamically determined time intervals. The main contributions of this work can be summarized as follows.

- We describe a novel data structure, which enables sentiment analytics approaches to scale to very large data collections by a careful indexing of time and demographics into hierarchies. It is incrementally maintained in an online environment, and can outperform a relational DBMS implementation by up to 3 orders of magnitude. Moreover, our storage maintains the statistical aggregates of sentiments, sufficient to extract the most important sentiment metrics and assess their significance. In order to enhance the performance of our algorithms, we also describe analytical results that allow to prune the search space, while maintaining quality guarantees on results.

- We formally define the problem of contradiction detection, and further describe two variations of the problem, namely, *synchronous* and *asynchronous* contradictions. We present an approach for sentiment contradiction detection, which is using a novel contradiction measure based on mean and variance of sentiment distribution. We also develop several other measures of aggregated sentiment, which capture the desired behavior. Our approach is uniquely designed to withstand noise and irregularity of online sentiment, thanks to regression analysis and smart thresholding.

- We study various correlation measures and describe their semantics within the scope of sentiment data. Based on a sentiment correlation of demographic groups, and their hierarchical relations, we define the concept of demographics maximality, which is especially useful for automated analysis of meaningful correlations on a large scale. Specific to demographics sentiments and maximality formulation, we describe efficient correlation pruning methods. Furthermore, we introduce two novel methods for correlation compression, which allow for the efficient implementation of our algorithms.

- We also aim at explaining the identified changes in social opinion through analyzing correlations between these changes and news events and understanding event dynamics. First, we propose a specific correlation method for these data. Second, we evaluate the differences in reaction to external events among various social media and develop methods able to reconstruct event importance based on the observed time series. Finally, we develop a framework that analyzes sentiment and news streams and models their causality with the aim to predict future sentiment shifts.

- We conduct an extensive set of experiments on several synthetic and real datasets to validate our problems, and evaluate the performance of our solutions. The experiments demonstrate that contradictions and correlated demographic groups can be identified very efficiently with the help of our specialized indexing storage and effective pruning. Finally, our evaluation provides insights on contradictions for a range of popular topics, and interesting correlations among real demographic groups, which can be of particular interest to social scientists and social recommender applications.

1.4 Related Publications

[–] Tsytsarau M, Palpanas T (2013) Mining Sentiment-based Contradictions at Web scale. InfSys, Information Systems, Elsevier (pending)

[136] Tsytsarau M, Amer-Yahia S, Palpanas T (2013) Efficient Sentiment Correlation for Large-scale Demographics. In: ACM SIGMOD Conference, New York, USA, June 22-27, 2013

[135] Tsytsarau M, Palpanas T, Castellanos M, Hsu M, Dayal U (2012) Identifying News Events That Cause A Shift In Sentiment. US Patent HP-82962988 (pending)

[130] Tsytsarau M, Palpanas T (2011a) Survey on mining subjective data on the web. DMKD, Data Mining and Knowledge Discovery, Special Issue on 10 Years of Mining the Web, pp 1-37, DOI: http://dx.doi.org/10.1007/s10618-011-0238-6, iSSN 1384-5810

[134] Tsytsarau M, Palpanas T, Denecke K (2011) Scalable detection of sentiment-based contradictions. In: First International Workshop on Knowledge Diversity on the Web, Colocated with WWW 2011, Hyderabad, India, March 28-31, 2011

[131] Tsytsarau M, Palpanas T (2011b) Towards a framework for detecting and managing opinion contradictions. In: ICDM Workshops, pp 1219–1222

[133] Tsytsarau M, Palpanas T, Denecke K (2010) Scalable discovery of contradictions on the web. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, USA, April 26-30, 2010

[132] Tsytsarau M, Palpanas T, Denecke K, Brosowski M (2009) Scalable Discovery of Contradicting Opinions in Weblogs. Tech. Rep. DISI-09-038, DISI, University of Trento

For the complete list of our publications with download links please refer to the following page: http://disi.unitn.it/~tsytsarau/index.html#content=Publications

Chapter 2

Background

The purpose of our background study is to highlight the development of the field with a focus on the recent years, examine the main trends that have appeared in the study of the field and their evolution over time, report in a systematic way the performance results of competing algorithms and the characteristics of the available datasets, and discuss some of the emergent topics and open research directions in the area of Subjectivity Analysis.

2.1 Introduction

Since the World Wide Web first appeared two decades ago, it has changed the way we manage and interact with information. It has now become possible to gather the information of our preference from multiple specialized sources and read it straight from our computer screen. But even more importantly, it has changed the way we share information. The audience (i.e., the receivers of the information) does not only consume the available content, but in turn, actively annotates this content, and generates new pieces of information. In this way, the entire community becomes a writer, in addition to being a reader. Today people not only comment on the existing information, bookmark pages, and provide ratings, but they also share their ideas, news and knowledge with the community at large.

There exist many mediums, where people can express themselves on the web. Blogs, wikis, forums and social networks are examples of such mediums, where users can post information, give opinions and get feedback from other users. In their own right, they collectively represent a rich source of information on different aspects of life, but more importantly so on a myriad of different topics, ranging from politics and health to product reviews and traveling. The increasing popularity of personal publishing services of different kinds suggests that opinionative information will become an important aspect of the textual data on the web.

Due to the ever-growing size of the information on the web, we are now barely able to access the information without the help of search engines. This problem gets harder, when we want to aggregate the information from different sources. Multiple solutions have been proposed to solve this problem, and they are mainly specialized in factual information retrieval.

To achieve this, subjectivity filtering is applied [116], in order to remove texts that may provide a biased point of view. These texts can be distinguished by analyzing sentiments expressed by the authors, or by discovering explicit marks of contradiction with other texts [37]. This dimension of web search emphasizes the importance of analyzing subjective data.

We now turn our attention to the following interesting question: whether the subjective data that exist on the web carry useful information. Information can be thought of as data that reduce our uncertainty about some subject. According to this view, the diversity and pluralism of information on different topics can have a rather negative role. It is well understood, that true knowledge is being described by facts, rather than subjective opinions. However, this diversity in opinions, when analyzed, may deliver new information and contribute to the overall knowledge of a subject matter. This is especially true when the object of our study is the attitude of people. In this case, opinionative data can be useful in order to uncover the distribution of sentiments across time, or different groups of people.

It is now becoming evident that the views expressed on the web can be influential to readers in forming their opinions on some topic [57]. Similarly, the opinions expressed by users are an important factor taken into consideration by product vendors [61, 56, 17] and policy makers [97]. There exists evidence that this process has significant economic effects [4, 5, 21]. Moreover, the opinions aggregated at a large scale may reflect political preferences [137] and even improve stock market prediction [13]. These arguments are illustrated in the following examples.

Example 1. Today we can see a growing number of blogs focused on various aspects of politics. They cover the entire spectrum of interested parties: from simple citizens expressing their opinions on everyday issues, to politicians using this medium in order to communicate their ideas (as was best exemplified during the last USA presidential elections), and from journalists criticizing the government to the government itself. It is to the benefit of all the parties mentioned above to follow the opinions that are expressed on a variety of topics, and to be able to identify how these opinions or public sentiments change and evolve across time.

Example 2. Imagine a potential buyer of a digital camera, who is not familiar with the details of this technology. In this case, reading the camera specifications can be an arduous task. In contrast, the opinion of the community that shares the same interests with the buyer, can be very informative. Therefore, a system that accumulates feedback and opinions originating from multiple sources, effectively aggregates this information, and presents the result to the user, can be both helpful and influential.

In this study, we introduce readers to the problems of *Opinion Mining* and *Opinion Aggregation*, which have been rapidly developing over the last decade, as well as with a rather new trend related to these areas, namely, *Contradiction Analysis*. In the rest of this document, we will use the term *Subjectivity Analysis* to refer to all three of the above problems together. We

provide discussions on what the form of the problem that all papers in each one of these areas solve is, and, where applicable, we also include a mathematical formulation.

To the best of our knowledge, only a few systematic surveys were published in this area, which nevertheless played an important role of establishing it as an independent field of research. Out of them we may name the works of Pang and Lee [107] and [125], observing the area from perspectives of machine learning and review mining respectively, as well as the recent work by Liu Bing [82], systemizing existing methods and problems.

Our current study has notable differences to the ones mentioned above, with respect to both new content, and also to the way that some common references are being treated. We provide a more balanced view on Subjectivity Analysis from perspectives of its historic development and evolution of methods. Our special attention is devoted to the interaction between different problems in this area and to a variety of methods applied to tackle with them. We emphasize on the differences between these methods and try to outline their scope and applicability for solving the current as well as the emergent problems. Furthermore, we present novel comparative information (in the form of graphs and tables) on the algorithms in this area, related to the techniques they use, their performance, as well as to the datasets used for their experimental evaluation. Moreover, we include a considerable amount of new information - starting from the newly appeared literature on the topic and our evaluation of existing papers from the different perspectives, to a discussion of absolutely novel trends - Opinion Mining in Microblogs and Opinion Quality and Spam. This information helps the reader form an overall picture for the general area of Subjectivity Analysis: where the past efforts have concentrated on, which the most popular methods and techniques are, and what the current trends are.

In contrast with the first work, we build up our discussion around a classification of the papers into four different approaches (machine learning, dictionary based, statistical, and semantic), also providing formal problem statements and method explanations wherever they were available for each of them. The approaches discussed are evaluated from different aspects, revealing their properties and applicability to Subjectivity Analysis. Similarly, the section on Opinion Aggregation mainly addresses the issues different to the work by [125]: we consider the recent advances, a general workflow, the problems and nuances of the area, rather than its particular methods. The work of Bing Liu [82], which appeared during the preparation of this work, also puts a considerable effort in systemizing the existing methods and providing their theoretical underpinnings. In our survey, we rather aimed at describing the evolution of these methods throughout the past decade and understanding the reasons of their popularity.

The rest of this chapter is organized as follows. In Section 2.2 we provide a general view of subjectivity analysis and outline major problems of this domain. Development, problems, definitions and main trends of this area are described in Sections 2.3 through 2.5. We analyze and discuss the state of the art in Section 2.7. Finally, we conclude in Section 2.8.

2.2 Subjectivity Analysis

Subjectivity Analysis involves various methods and techniques that originate from Information Retrieval (IR), Artificial Intelligence and Natural Language Processing (NLP). This confluence of different approaches is explained by the nature of the data being processed (free-form texts) and application requirements (scalability, online operation). Therefore, Subjectivity Analysis shares much of its terminology and problem definitions with the domains mentioned above. In the following paragraphs, we discuss in more detail the literature on the problems of *Opinion Mining* and *Opinion Aggregation*. We review the recent developments in these areas, and then present the field of *Contradiction Analysis*, which has recently started to attract interest.

2.2.1 Problems and Goals

The Subjectivity Analysis domain is still in the process of being shaped, and its problem statements touch upon different domains. Being originally studied in different communities, the problems of *Opinion Mining* and *Sentiment Analysis* have slightly different notions. Opinion Mining originates from the IR community, and aims at extracting and further processing users' opinions about products, movies, or other entities. Sentiment Analysis, on the other hand, was initially formulated as the NLP task of retrieval of sentiments expressed in texts. Nevertheless, these two problems are similar in their essence, and fall under the scope of Subjectivity Analysis. For the rest of this document, we will use both these terms interchangeably.

At a first level of approximation, the various Subjectivity Analysis techniques can be described as being composed of the following three steps:

1. *identify*; 2. *classify*; 3. *aggregate*.

These steps also implicitly list the most important problems in Subjectivity Analysis. For example, a typical opinion mining process involves the first two steps, and results in producing sentiment values for texts. In opinion aggregation, the third step is involved as well, in order to aggregate these sentiments. Note that even though this aggregation can be considered as a post-processing step, it is no less important than the previous steps. Indeed, the analyst is often times more interested in determining the common features and interesting patterns that emerge through sentiments from many different data sources, rather than in the opinions of particular authors.

2.2.2 Terminology and Definitions

The subjectivity analysis area is a relatively new direction of research. As such, there is no established common framework for describing and modeling the relevant problems. Though, some recent studies have made the first steps towards this direction.

Opinion Mining operates at the level of documents, that is, pieces of text of varying sizes and formats, e.g., web pages, blog posts, comments, or product reviews.

Definition 1 (Document) *D is a piece of text in natural language.*

We assume that each document discusses at least one topic, and not all topics discussed in the same document have to be related to each other.

Definition 2 (Topic) *T is a named entity, event, or concept that is mentioned in a document D.*

Examples of such topics are product features, famous persons, news events, happenings, or any other concept s that may attract our interest. What we are interested in is analyzing these topics in connection to any subjective claims that accompany them. Therefore, for each of the topics discussed in a document, we wish to identify the author's opinion towards it.

Definition 3 (Sentiment) *S is the author's attitude, opinion, or emotion expressed on topic T.*

Sentiments are expressed in natural language, but as we will see below, they can in some cases be translated to a numerical or other scale, which facilitates further processing and analysis.

There are a number of differences in meaning between emotions, sentiments and opinions. The most notable one is that *opinion* is a transitional concept, which always reflects our attitude towards something. On the other hand, sentiments are different from opinions in that they reflect our feeling or emotion, not always directed towards something. Further still, our emotions may reflect our attitudes.

Generally speaking, the palette of human emotions is so vast, that it is hard to select even the basic ones. Most of the authors in the NLP community agree on the classification proposed by Paul Ekman and his colleagues [35], which mentions six basic emotions: *anger, disgust, fear, joy, sadness, surprise*. Although this classification is consistent in itself, it needs to be further extended by antonyms in order to allow capturing positive and negative shifts in opinion. Accordingly, Jianwei Zhang et al. [153] propose to group the basic emotions along four dimensions: *Joy* $\Leftrightarrow$ *Sadness, Acceptance* $\Leftrightarrow$ *Disgust, Anticipation* $\Leftrightarrow$ *Surprise, and Fear* $\Leftrightarrow$ *Anger*. However, such a division requires a rather complex processing and analysis of the input data, which is not always feasible. Therefore, the majority of the authors accept a simpler representation of sentiments according to their *polarity* [107]:

Definition 4 (Sentiment Polarity) *The polarity of a sentiment is the point on the evaluation scale that corresponds to our* positive *or* negative *evaluation of the meaning of this sentiment.*

Sentiment polarity allows us to use a single dimension (rather than the four dimensions mentioned above), thus, simplifying the representation and management of the sentiment information.

De Marneffe et al. [89] introduce a classification of contradictions consisting of seven types that are distinguished by the features that contribute to a contradiction (e.g., antonymy, negation, numeric mismatches). Antonymy are words that have opposite meanings, i.e., "hot - cold" or "light - dark". Antonymy can give rise to a contradiction when people use these words to describe some topic. Negation imposes a strict and explicit contradiction, e.g., "I love you - I love you not". Numeric mismatches form another type of contradiction, which may be caused by erroneous data: "the solar system has 8 planets - there are 9 planets orbiting the sun". Their work defines contradictions as a situation where "two sentences are extremely unlikely to be true when considered together". In other words, contradictions may be defined as a form of textual entailment, when two sentences express mutually exclusive information on the same subject [53].

The works discussed above rely on human-perceivable definitions of contradiction that summarize our expectations about which features contribute to a contradiction. Opposite sentiments are also very common sources of contradictions. However, they may be described in different terms compared to the textual entailment problem. Consider the following example: "I like this book - This reading makes me sick". Both sentences convey a contradiction on opinions expressed about a book, yet they may appear together if they belong to different authors. Therefore, we may relax the 'exclusivity' constraint of textual entailment and propose the following definition:

Definition 5 (Contradiction) *There is a contradiction on a topic T in a document collection $\mathcal{D}$, between two sets of documents, $\mathcal{D}_1, \mathcal{D}_2 \subset \mathcal{D} \mid \mathcal{D}_1 \bigcap \mathcal{D}_2 = \emptyset$, when the information conveyed about T is considerably more different between $\mathcal{D}_1$ and $\mathcal{D}_2$ than within each one of them.*

In the above definition, we purposely not specify exactly what it means for a sentiment value to be very different from another one. This definition captures the essence of contradictions, without trying to impose any of the different interpretations of what might cause a contradiction to arise. For example, if we assume that opinion polarity is the relevant information, then a contradiction would mean that two groups of documents express contrasting opinions on some topic. In Section 2.5 we discuss some of the papers for contrasting opinion summarization, which rely on this principle.

Another interesting application of contradiction analysis is in supplementing information retrieval systems, which in most of the cases are fact-centric. Diverse opinions introduce extra noise to such systems, which are intended to provide a solid and unbiased representation of information about different topics [116]. Understanding contradicting opinions allows information retrieval systems to deal with opinionative data using special methods, for example by extracting the ground truth from different discussions or representing user support against different conflicting topics.

2.3 Opinion Mining

Opinion Mining is the problem of identifying the expressed opinion on a particular subject and evaluating the polarity of this opinion (e.g., whether the expressed opinion is positive or negative). Opinion Mining forms the basis upon which other tasks under Subjectivity Analysis can be built. It provides an in-depth view of the emotions expressed in text, and enables the further processing of the data, in order to aggregate the opinions, or identify contradicting opinions. Evidently, the quality of the results of Opinion Mining is crucial for the success of all subsequent tasks, making it an important and challenging problem.

2.3.1 Problems in Opinion Mining

In the area of Opinion Mining, studies usually follow a workflow consisting of two steps: *identify* (topics, opinionative sentences), and *classify* (sentences, documents).

In the first step, we need to identify the topics mentioned in the input data, and also associate with each topic the corresponding opinionative sentences. During this step, we may also try to distinguish between opinionative and non-opinionative phrases (i.e., perform *subjectivity identification*). This additional task is useful, because not all phrases that contain sentiment words are, in fact, opinionative. The reverse claim is also true: some of the opinionative phrases do not contain positively (or negatively) charged words. Therefore, performing this identification task can be an effective addition to the classification step in order to improve precision [146, 32, 105, 116, 145, 147]. Furthermore, retrieval of opinionative documents evolved into a separate task with many specific algorithms, like in [152, 70, 154, 54].

During the second step, the problem of *sentiment classification* is most often a binary classification problem, distinguishing between *positive* and *negative* texts. Nevertheless, additional classes can also be introduced, in order to make the analysis more robust and increase the quality (i.e., granularity) of results. For example, some of the works include the *neutral* or *irrelevant* sentiment categories, which mean that there is no sentiment. By doing this, we can avoid the subjectivity identification task mentioned above, and have the classifier distinguish between opinionative and non-opinionative phrases. There is evidence that this approach has a positive effect on the precision of the final results [68]. Previous work [155] has also tried to improve sentiment classification by running this task separately for each of the topic's features (determined by an ontology) and averaging the output. Though, this step is generally considered as separate from topic identification [107].

In summary, we could argue that Opinion Mining can be viewed as a classification problem, distinguishing between several classes of sentiments (most often, *positive*, *negative* and *neutral*). This division is applicable to some extent even to the methods that produce sentiments on a numerical scale, in which case the division becomes a matter of setting thresholds (between the sentiments classes).

2.3.2 Development of Opinion Mining

Opinion Mining has been studied for a long time. Yet, the research in this area accelerated with the introduction of *Machine Learning* methods and the use of annotated datasets [95, 108, 151, 32]. The evaluations found in [151, 71, 32, 3] demonstrate that opinion data obtained from the web, are represented primarily in discrete or categorical form. This happens mainly because ratings and opinion labels are represented by a limited number of categories on the web. Such availability of categorical training data favors the use of machine learning for such tasks as rating inference or review mining, and made machine learning tools the default choice for solving the Opinion Mining problem. A side effect of the domination of these tools is that the sentiment classification task is mostly considered as a binary- or three-class classification problem, distinguishing among *positive*, *negative*, or *neutral* texts. However, it is not clear whether this approach is the winner. On the contrary, recent studies demonstrate the benefits of employing more complex (detailed) sentiment classifications [127], that provide sentiment values on a continuous scale. Moreover, it is not always possible to use supervised machine learning methods. For example, when there are no annotated training data (like in blog opinion retrieval), other types of approaches, like *Dictionary*, *Statistical*, and *Semantic* become an interesting alternative.

The **Machine Learning Approach** is a sophisticated solution to the classification problem that can be generally described as a two-step process: 1) learn the model from a corpus of training data (supervised, unsupervised), and 2) classify the unseen data based on the trained model.

Below, we provide a formal statement for the (supervised) learning step, adapted to our terminology. We assume training data are documents represented in a space, $\mathbb{D}$, whose dimensions are document features (e.g., frequency of words, bi-grams, etc.). Furthermore, these documents have been assigned a sentiment label from a space $\mathbb{S}$:

$$\text{Given training data } \{(D_i \in \mathbb{D}, S_i \in \mathbb{S})\}, \text{ find } g : \mathbb{D} \rightarrow \mathbb{S},\ g(D_i) = \arg\max_S f(D_i, S_i) \qquad (2.1)$$

The above formulation says that given a set of training pairs, we want to find a function g that maps documents to sentiment labels, according to the best prediction of some scoring function f. This function takes as input documents and sentiment labels and gives a sentiment label probability prediction (using either conditional or joint probability). Without loss of generality, the learning process can be considered as an estimation of the scoring function. Examples of such scoring functions are feature vectors in $\mathbb{D}$, computed relative to class separating hyperplanes, or functions based on decision trees.

The machine learning approach involves the following general steps. First, a training dataset

is obtained, which may be either annotated with sentiment labels (supervised learning), or not (unsupervised learning). Second, each document is represented as a vector of features. We describe examples of such representations further in the text. Third, a classifier is trained to distinguish among sentiment labels by analyzing the relevant features. Finally, this classifier is used to predict sentiments for new documents.

The current popularity of the machine learning approach for opinion mining originates from the work "Thumbs up?" by Pang and Lee [108]. The authors proposed and evaluated three supervised classification methods: Naive Bayes (NB), Maximum Entropy (ME) and Support Vector Machines (SVM). According to their evaluation, SVM showed the best performance, while NB was the least precise out of the three (though, the differences among them were small). Nevertheless, all the algorithms clearly surpassed the random choice baseline, exhibiting an average precision of around 80%. Dave et al. [32] further extended the work of Pang and Lee, emphasizing feature selection. They also used Laplacian smoothing for NB, which increased its accuracy to 87% (for a particular dataset). However, the SVM classifier has achieved similar results, performing below NB only when using unigram features (refer also to Table 2). Pang and Lee [105] also used subjectivity identification as a preprocessing step in order to improve the precision of NB.

The sentiment analysis task is very similar to the rating inference task, in which the class labels are scalar ratings, such as 1 to 5 "stars", representing the polarity of an opinion. The need to provide a finer resolution of sentiments, without affecting the classification accuracy, required different multi-class categorization methods compared to traditional SVM. Although the SVM method has proved its efficiency for binary classification, the new problem demanded more sophisticated solutions.

To address this challenge, Pang and Lee [106] in their study "Seeing Stars" proposed to use SVM in multi-class *one-versus-all* (OVA) and *regression* (SVR) modes, combining them with metric labeling, so that similar classes are positioned closer to each other on a rating scale. Metric labeling is a special case of *a-posteriori* optimization of class assignment with respect to *prior* assignment. This class assignment minimizes the sum of distances between labels of adjacent points, penalized by point similarities. Their results clearly demonstrated that a combination of SVM with other unsupervised classification methods results in better precision. A subsequent work on support or opposition [128] further extended this approach through modeling relationships and agreement between authors in the context of political texts.

The performance of machine learning methods is highly dependent on the quality and quantity of training data, which is scarce compared to the amount of unlabeled data. The paper titled "Seeing Stars When There Are Not Many Stars" [52] proposes a semi-supervised learning technique operating on a graph of both labeled and unlabeled data. The authors represent documents with a graph, where vertices correspond to documents, and edges are drawn between

similar documents using a distance measure computed directly from document features. These assumptions are similar to metric labeling, except that they are used *a-priori*, thus, allowing to use even unlabeled data for training. Although their approach exhibited better performance than SVR, the authors mention that it is sensitive to the choice of the similarity measure, and not able to benefit from the use of additional labeled data.

In the studies discussed above, rating inference tasks have been considered at the document level, thus showing an 'average' precision on heterogeneous reviews, which mention multiple aspects of the product with different sentiments expressed on each one. This brings up the problem of contextual sentiment classification, which requires algorithms not only operating at the sentence level, but also involving the context of each sentence in their analysis [147]. Extending on [106], Shimada and Endo [118] proposed to analyze ratings on the product-feature level, naming their work "Seeing Several Stars". They have demonstrated that SVR, despite being less precise than SVM, produces output labels that are closer to the actual ones. This evidence also supports the claim in [106] that with the use of a "gradual" function in SVR "similar items necessarily receive similar labels".

Topic-dependent sentiment analysis on a sub-document level is also becoming a standard tool for opinion extraction. O'Hare et al. [101] present an approach to topic-dependent sentiment analysis in financial blogs. In particular, positive and negative opinions expressed towards companies and their stock are distinguish by a machine learning approach (Naive Bayes, SVM). Based on topic terms, their approach first identifies text paragraphs dealing with a specific topic. Sentiment analysis is then performed on sub-document level.

Apart from the choice of algorithms and data selection, the performance of machine learning approaches is heavily dependent on feature selection. The most straightforward (yet, in some cases very effective) way is to encode each feature in the set by its presence or absence in the document. In the case of word features, this would produce a simple binary vector representation of a document. Extending this representation, we can instead use relative frequencies of words' occurrence [118]. Though, not all words are equally representative and, therefore, useful for subjectivity analysis. This provides an opportunity to make the learning process more efficient by reducing the dimensionality of $\mathbb{D}$ (refer to Formula 2.1). Osherenko et al. [102] demonstrate that it is possible to use just a small set of the most affective words as features, almost without any degradation in the classifier's performance. Interestingly, the direct use of sentiment values from such dictionaries, instead of binary feature presence values, has shown little to no increase of precision. Therefore, studies usually use frequencies of words instead. For example, Devitt and Ahmad [34] identify sentiment-bearing words in a document by using SentiWordNet, but then use just their frequencies of occurrence for the classification task. This approach is also popular with dictionary methods, which we describe below.

Finally, we should mention that machine learning is used for other problems of opinion

mining as well. For instance, Zhang et al. [154] describe an approach that uses an SVM trained on a set of topic-specific articles obtained from Wikipedia (objective documents) and review sites (subjective documents) to perform subjectivity identification. Another example is the use of machine learning to classify opinion spam [62, 78].

The **Dictionary Approach** relies on a *pre-constructed dictionaries* that contain opinion polarities of words. When it is necessary to process a continuous flow of texts coming at a high rate (which is the case for web-scale analysis), simple implementations of this approach can calculate sentiments with the very high performance while keeping their accuracy still usable for aggregated analytics.

The most popular dictionaries today are the General Inquirer[1], the Dictionary of Affect of Language[2], the WordNet-Affect[3], or the SentiWordNet [39]. For instance, SentiWordNet has been created automatically by means of a combination of linguistic and statistic classifiers and provides a triple of polarity scores (positivity, negativity and objectivity) for each set of synonyms from WordNet. Existing works exploit these resources mainly for identification of opinionative words, although some recent studies showed that it is possible to use polarity scores directly, providing a sentiment value on a continuous scale [40, 132, 94].

Most of the dictionary methods compute expression sentiments using simple rule-based algorithms and then aggregate polarity values for a sentence or a document by averaging the polarities of individual words, without considering sentence's syntactic structure or discourse [156]. We now describe a formula that defines the most basic case of document opinion assignment using a dictionary:

$$S(D) = \frac{\sum_{w \in D} S_w \cdot weight(w) \cdot modifier(w)}{\sum weight(w)} \tag{2.2}$$

In the above equation, S_w represents the dictionary sentiment for a document word *w*, which is being aggregated with respect to some weighting function $weight()$ and modifier operator $modifier()$ (which handles negation, intensity words, and other cases affecting *a-priori* sentiment). Weighting functions may be defined statically for each sentence, or computed dynamically, with respect to positions of topic words. Usually weighting functions represent a window around the topic word, thus, taking into account the sentiments of the words that are immediate neighbors of the topic word in the document. For example, a weighting function can have the value of 1 for two or three words surrounding the topic word, and 0 elsewhere. The weighting of phrase words (reranking) can also be performed by machine learning methods [64], providing

[1] http://www.wjh.harvard.edu/~inquirer/
[2] http://www.hdcus.com/
[3] http://wndomains.fbk.eu/wnaffect.html

yet another hybrid method in addition to using sentiment polarity features in machine learning classifier.

A more sophisticated processing of texts typically involves NLP methods, which we believe to provide the most accurate sentiment values, subject to using a context-dependent dictionary. For instance, the Sentiment Analyzer introduced by Yi et al. [151] the Linguistic Approach by Thet et. al [127], or Live Customer Intelligence by Castellanos et al. [17], extract sentiments precisely for some target topics using advanced methods that exploit domain-specific features, as well as opinion sentence patterns, Part-Of-Speech tags and syntactic parsing. [127] uses a linguistic approach for analyzing sentiments of movie reviews along a scale of -1 and 1 on clause-level. The prior sentiment scores of the words derived from SentiWordNet and used by rules to determine a contextual sentiment score for each clause utilizing grammatical dependencies of words. Different rules are specified for the various parts of a sentence (subject, object, predicate, verb phrase).

We note that relying on the polarity values assigned by a dictionary is not always feasible, as the dictionary may not be suited for use on particular datasets (e.g., may not include some domain-specific lexicons). Furthermore, dictionary methods are usually not able to adapt polarity values to particular contexts. It turns out that words can change their polarity when used in different contexts [40]. Consider the adjectives “cold” (generally regarded as negative), and “warm” (regarded as positive). When these adjectives are used in the phrases “cold wine” and “warm beer”, their polarities change to positive and negative, respectively. Such changes mostly occur with the words with ambiguous meaning or with adjectives, which are usually not present in general (cross-domain) sentiment dictionaries. Nevertheless, sentiment lexicons can be extended or adapted to specific domains to improve the recall of sentiment extraction. For instance, [17] determine aspect-dependent sentiment words by noting their presence within aspect-related clauses and then optimizing their polarity score through a robust regularization framework [87]. In contrast to the dictionary approach, machine learning methods naturally adapt to the corpus they are trained on.

The **Statistical Approach** aims to overcome the problems of the Dictionary Approach mentioned above by using *dynamically-constructed dictionaries*. For example, Farni and Klenner [40] propose to derive posterior polarities using the co-occurrence of adjectives in a corpus. In this case, adaptability is achieved through the construction of a corpus-specific dictionary. Regarding the problem of unavailability of some words, the corpus statistics method proposes to overcome it by using a corpus that is large enough. For this purpose, it is possible to use the entire set of indexed documents on the Web as the corpus for the dictionary construction [139].

We can identify the polarity of a word by studying the frequencies with which this word occurs in a large annotated corpus of texts [76, 93]. If the word occurs more frequently among

positive (negative) texts, then it has a positive (negative) polarity. Equal frequencies indicate neutral words. It is also interesting to mention, that applications working with the Chinese language are able to recognize polarity even for unseen words, due to the fact that phonetic characters determine the word's sense [70, 71]. In this case, we can analyze frequencies of single characters rather than words. Although computationally efficient, the basic method requires a large annotated corpus, which becomes a limiting factor.

The state of the art methods are based on the observation that similar opinion words frequently appear together in a corpus. Correspondingly, if two words frequently appear together within the same context, they are likely to share the same polarity. Therefore the polarity of an unknown word can be determined by calculating the relative frequency of co-occurrence with another word, which invariantly preserves its polarity (an example of such a word is "good"). To achieve this, Peter Turney [139, 138] proposed to use the Point-wise Mutual Information (PMI) criterion for statistical dependence [25], replacing probability values with the frequencies of term occurrence $F(x)$ and co-occurrence $F(x \textit{ near } y)$:

$$PMI(x,y) = \log_2 \frac{F(x \textit{ near } y)}{F(x)F(y)}; \tag{2.3}$$

Sentiment polarity (expressed by *PMI-IR*) for word x is then calculated as the difference between *PMI* values computed against two opposing lists of words: positive words, *pWords*, such as "excellent", and negative words, *nWords*, such as "poor":

$$PMI\text{-}IR(x) = \sum_{p \in pWords} PMI(x,p) - \sum_{n \in nWords} PMI(x,n) \tag{2.4}$$

Along with the formulas above, Turney et al. proposed to obtain the co-occurrence frequencies F by relying on the statistics of the AltaVista web search engine. Extending on this work, Chaovalit et al. [18] used Google's search engine to determine the co-occurrence of words, increasing the precision. Read et al. [115] further extended this approach, employing Semantic Spaces and Distributional Similarity as alternative weakly-supervised methods. A detailed study on constructing dictionaries of this kind was made by Taboada et al. [123], mentioning some problems that occur due to the unavailability of the "near" modifier or non-persistence of the search engine's output. On the other hand, search engines allow retrieving the co-occurrence scores (thus, polarities) not only for words, but also for phrases, which is a useful feature.

The use of statistical methods in computing opinion polarity has found an interesting development in the work of Ben He et al. [54], where they propose to use an opinion dictionary along with IR methods in order to retrieve opinionative blog posts. Their approach first builds a dictionary by extracting frequent terms from the entire collection, which are then ranked according to their frequency among opinion-annotated texts. The sentiment polarity of each document is computed as a relevance score to a query composed of the top terms from this dictionary.

Finally, the opinion relevance score is combined with the topic relevance score, providing a ranking of opinionative documents on that topic.

The **Semantic Approach** provides sentiment values directly (like the Statistical Approach), except that it relies on different principles for computing the similarity between words. The underlying principle of all approaches in this category is that semantically close words should receive similar sentiment values.

WordNet [42] provides different kinds of semantic relationships between words, which may be used to calculate sentiment polarities. The possibility to disambiguate senses of words using WordNet can serve as a way to include the context of these words into the opinion analysis task. Similar to statistical methods, two sets of seed words with positive and negative sentiments are used as a starting point for bootstrapping the construction of a dictionary.

Kamps et al. [65] proposed to use the relative shortest path distance of the "synonym" relation, demonstrating a good degree of agreement (70%) with an annotated dictionary. Another popular way of using WordNet is to obtain a list of sentiment words by iteratively expanding the initial set with synonyms and antonyms [67, 58]. The sentiment polarity for an unknown word is determined by the relative count of positive and negative synonyms of this word [67]. Otherwise, unknown words may also be discarded [58]. However, it is important to know that since the synonym's relevance decreases with the length of the path between the synonym and the original word, so should the polarity value, too. Additionally, the polarity of a word is often averaged over all possible paths to it. Though, as was pointed out by Godbole et al. [51], we should only consider paths that go through the words of the same polarity as initial.

2.3.3 Opinion Mining in Microblogs

In the above paragraphs, we mostly considered static approaches to the problem of Opinion Mining, where the classifier's model does not change after being constructed. However, there exists another class of applications, such as those analyzing messages in microblogging, which require adaptability of the model to changing data during the analysis.

The most prominent example of microblogging platforms, which allows for real-time analysis, is Twitter. Its vast user community, all-around presence and informal style of conversation make Twitter a rich source of up-to-date information on different events and a good indicator of users' moods. Recently, it was demonstrated that sentiments from Twitter messages correlate with political preferences [137, 81], and even improve stock market prediction [13].

Recent works have identified several differences between opinion mining in microblogs when compared to conventional opinion analysis of documents. The most useful feature of short messages is the availability of sentiment or mood annotations in messages, providing a good source of training data for classifiers [50, 9, 104]. Another feature convenient for senti-

ment extraction is the general tendency to mention only one topic and a wide usage of hashtag annotations. However, a fairly short size of messages and the proliferation of internet slang require adaptations of existing sentiment extraction models.

Pak and Paroubek [104] performed statistical analysis of linguistic features of Twitter messages and report interesting patterns which may help distinguish among sentiment classes. They demonstrate that an NB classifier, based on negation extended bi-gram features, achieves good accuracy (albeit, at the expense of low recall) and can be useful to information retrieval applications. Bermingham and Smeaton [7] compared the performance of SVM and Multinomial Naive Bayes (MNB) classifiers on microblog data and reviews, and demonstrated that in most cases these classifiers yield better results on short-length, opinion-rich microblog messages.

However, sometimes the content of short text messages is not sufficient to extract their sentiments reliably. Hu et al. [60] describe a method of sentiment analysis for microblog messages that leverages on users social network in addition to conventional text mining features. In particular, the authors extend a Linear Regression learning method with a sparse regularization component to include optimization factors based on the consistency of sentiments from the same user (sentiment consistency) and from connected users (emotional contagion). Moreover, the proposed method requires smaller amounts of annotated data and overcomes the problem of noisy and sparse training data through the regularization based on the aforementioned sociological theories.

Lin et al. [81] evaluate political sentiments in Twitter during US presidential elections, focusing on aggregated sentiments of pre-selected (partisan) groups of users, who are likely to share the same background or biases. They consider that sentiments in social media are mainly reflecting the reaction towards various events rather than the overall attitude, and should be mined to extract behavioral patterns instead of inferring background preferences.

Since class distributions may vary along the stream of data, there is a necessity to follow these changes and update the classifier's model accordingly. Bifet and Frank [9] studied the problem of using an adaptable classifier with Twitter data and examined relevant evaluation methods. They proposed to use the Stochastic Gradient Descent (SGD) method to learn a linear classifier. Their approach allows specifying the rate with which model's parameters are updated, and to monitor the evolution of the impact of individual words on class predictions. The latter may be used as an indicator of users' support or opposition to particular topics in a stream. In addition, SGD demonstrated an accuracy smaller but comparable to that of MNB (67.41% versus 73.81%). Moreover, the authors proposed to use an updatable baseline statistics (Kappa), when evaluating the classifier performance. The intuition behind this choice is similar to adapting a classifier to streaming data: as class distributions may change, the performance baseline computed on static data may not correspond to local periods with small entropy (high accuracy) and thus the performance can become biased towards smaller accuracy.

2.4 Opinion Aggregation

The analysis of opinions at a large scale is impractical without automatic aggregation and summarization. In this case, we are interested in identifying opinions at a higher level than that of an individual: we would like to identify the average or prevalent opinion of a group of people about some topic, and track its evolution over time.

What distinguishes Opinion Aggregation from other tasks, is the necessity to provide summaries along several features, aggregated over one or more dimensions. Therefore, feature extraction and aggregation appear as the key problems here, and we are going to concentrate our attention on these tasks.

The problem of mining product reviews has attracted particular attention in the research community [95, 32, 83, 15]. This problem imposes certain challenges related to the extraction of representative features and the calculation of the average sentiment or rating. The final goal though, is to determine the overall opinion of the community on some specific product, rather than the individual user opinion on that product.

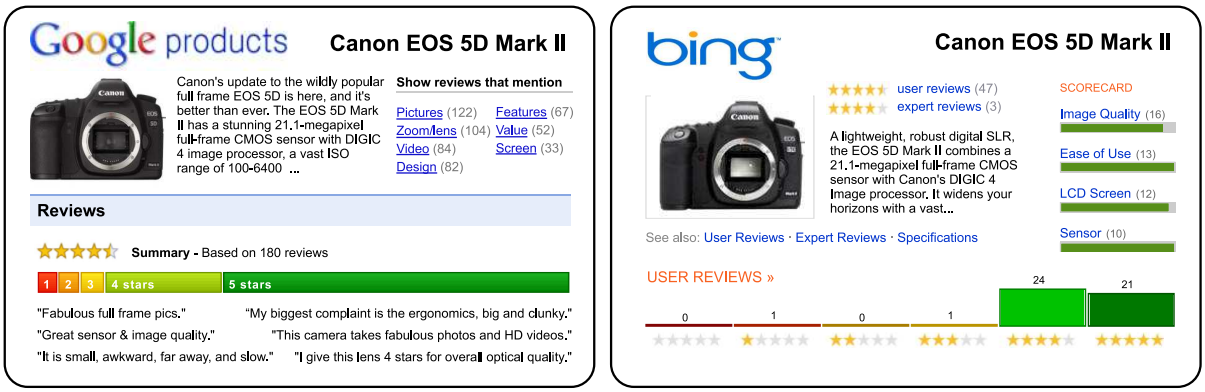


Figure 2.1: An example of Google and Bing review aggregations (actual images and text were arranged for better representation).

Today we can already see working examples of opinion aggregation at several web sites that visualize ratings assigned by a community of users. In Figure 2.1, we depict two examples of opinion aggregation, from the Google and Bing web search engines. Both of them feature images, short descriptions, and aggregate ratings. Additionally, they include statistics for each rating category (number of "stars"). Overall, these two approaches show similar details on the featured product, except that Google offers a representative summary (sentences at the bottom), while Bing displays aggregated ratings for each product feature (displayed on the right).

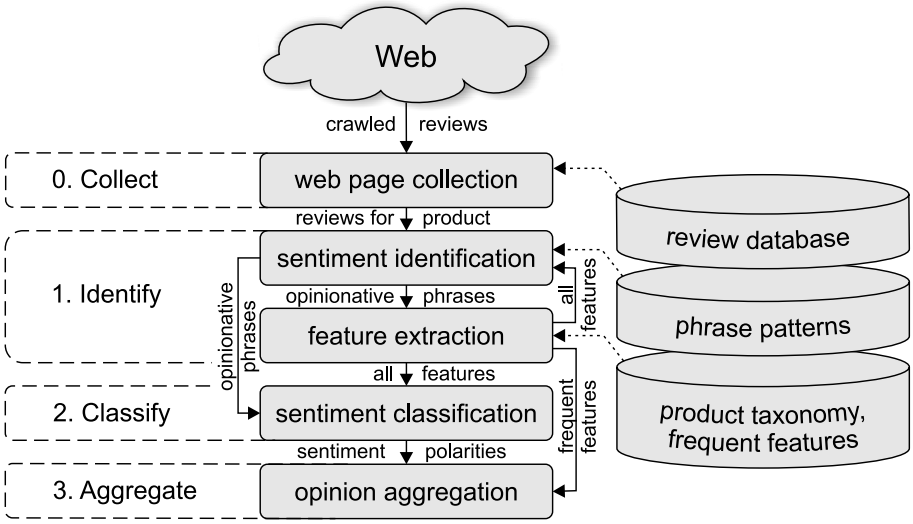


Figure 2.2: An example architecture of product review aggregation.

2.4.1 Problems in Opinion Aggregation

Review mining is the main application domain for opinion aggregation. Therefore, the problems that have been studied in relation to opinion aggregation are mainly formulated around the aggregation of product reviews. They include the processes of collecting, mining and reasoning on customer feedback data, represented in the form of textual reviews [125].

Figure 2.2 illustrates the review mining process. The process starts with the identification of opinionative phrases, which may additionally involve a collection of phrase patterns, or comparative sentences (in this case, sentiments are expressed by means of comparison of an object to another similar object) [82]. Identified phrases are then passed on to the feature extraction step, which may exploit a product taxonomy database [15] in order to improve the results. Features and opinionative phrases are used in the sentiment classification step, which outputs sentiment polarities to be aggregated over frequent features at the opinion aggregation step. This process can be iterative, using the identified features in order to improve the phrase extraction step.

Although Opinion Aggregation is a separate task having its own problems, practical applications also involve information retrieval and sentiment analysis techniques during the data pre-processing. Thus, the Opinion Aggregation techniques have been developing in close connection to other methods, and were subsequently revisited when improved sentiment analysis and feature extraction methods were introduced. Generally speaking, Opinion Aggregation methods are quite modular and may be used with different Opinion Mining algorithms. For example, Carenini et al. [15] describe a system that relies on sentiment extraction only as a preprocessing task, concentrating their attention on the aggregation of user reviews.

Aggregation of opinions for a product, expressed in a document collection $\mathcal{D}$, may be formu-

lated as the problem of determining a set of product features (each labeled with a corresponding sentiment), satisfying certain criteria:

$$\{(f, \mu_S)\} \mid rep(f, \mathcal{D}) > \rho_f,\ \mu_S = agg(S, f),\ \text{satisfying}\, con(S) \tag{2.5}$$

Where f is a product feature that is important for the description of the product in $\mathcal{D}$, according to some representativeness measure $rep()$, and μ_S is the sentiment for f, computed over $\mathcal{D}$ according to some aggregating function $agg()$. During this procedure, we may only consider features with a representativeness measure over some threshold ρ_f, and corresponding sentiments that satisfy some constraints, expressed by $con(S)$. Examples of such constraints are imposing a limit on the sentiment's absolute value (e.g., consider only moderate opinions), or the timestamp (e.g., consider only recent opinions).

We note that Opinion Aggregation is different from Opinion Summarization [129, 66, 112], which is the problem of producing a shortened version of the corresponding text. These problems are complementary to each other in a way that while Opinion Aggregation extracts average sentiments for topic features, the task of Opinion Summarization is to provide exempts from text, bearing or corresponding to these sentiments. In this study we focus on the former since it involves aggregated sentiment.

2.4.2 Development of Opinion Aggregation

A typical method for opinion aggregation was proposed by Hu et al. [58]. They describe a system that aims at discovering words, phrases, and sentiments that best characterize some product. At a high level, their solution follows the steps we listed in the previous section. We note though, that not all studies follow this pattern. For example, Morinaga et al. [95] reversed the ordering of steps 1 and 2, and the experiments revealed that their system achieves a similar performance. By running opinion classification prior to identification of features, we effectively apply some kind of filtering on features: we remove those that were not mentioned in an opinionative phrase (since these are features that are irrelevant for our analysis).

Different approaches to feature extraction have been proposed. Hu et al. [59] identify features by building a list of noun-noun phrases using an NLP parser, and then determining the most frequent ones. Feature frequency in this case corresponds to the $rep()$ function in Formula 2.5. However, their approach outputs many irrelevant words and should be used in conjunction with other methods, as was suggested by Carenini et al. [15]. Accordingly, they introduce a domain taxonomy in the form of user-defined features, which are used to annotate data for training a feature classifier. Opinions are then collected and aggregated based on the full set of features, which consists of features extracted automatically (unsupervised learning) and also through the classifier (supervised learning). Alternatively, Ku et. al. [70] proposed a

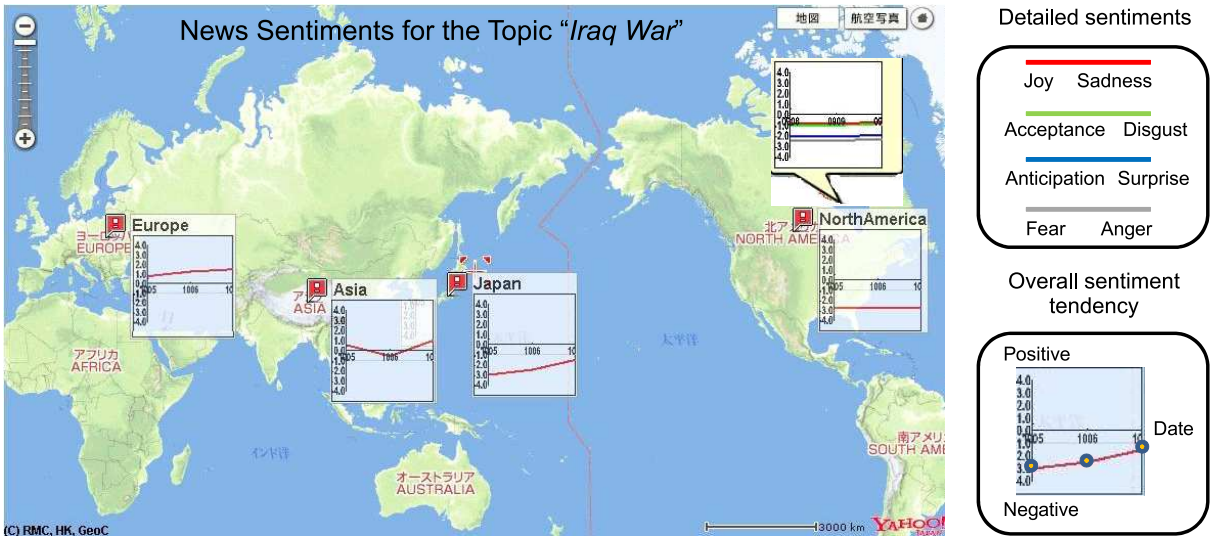


Figure 2.3: An example of geographical sentiment aggregation from [153].

system that identifies features by using information retrieval methods. They use TF-IDF scores per paragraph and per document, and a dictionary to determine polarity. The intuition here is that relevant features appear frequently in few of the paragraphs of many documents, or in many of the paragraphs of few documents. This technique is also efficient for eliminating the irrelevant features described above.

Aggregation of opinions has been traditionally performed over all the documents in some collection. Miao et al. [93] proposed a time-decaying aggregation approach, retrieving only the most recent reviews that were marked by users as helpful. The above constraints are represented by the *con*() function in Formula 2.5. Jianwei Zhang et al. [153] introduced a novel technique, which interactively aggregates and displays sentiments based on different granularities of time and space (geographical location). Moreover, the sentiments are represented by several dimensions, making it the most robust Web-scale application we observed in our study. An example of such an aggregation is shown in Figure 2.3. In this figure, we can see a world map and the time evolution of the sentiments in news articles for different geographical regions of the world. These results are illustrated in pop-up boxes, which report the values of four sentiment dimensions (i.e., joy-sadness, acceptance-disgust, anticipation-surprise, and fear-anger) over time. This system automatically retrieves and displays sentiments around some particular time period for *ad-hoc* queries, aggregating them over different locations as the user navigates the map, or zooms in and out. However, it only targets small-scale data aggregation using a single demographics hierarchy. A recent work of Mandel et al. [88] is a step up in this direction, featuring sentiment aggregation over time for demographic groups formed by gender and location attributes.

2.4.3 Opinion Aggregation and Spam

Aggregation of opinions usually involves a large number of texts, and because of this it has some advantages of scale: resistance to random errors in sentiment identification, more reliable feature identification, possibility to learn or adapt to data. However, it also has some vulnerabilities, such as: loss of deviations in data, accumulation of constant errors in sentiment identification, possibility to manipulate aggregate values by introducing artificially correlated data. This makes opinion quality assessment and spam detection being necessary steps during Opinion Aggregation.

With the rapid growth of web sites featuring product ratings and their increasing impact on users' opinions and/or buying decisions, it comes as no surprise that we observe a significant interest to this area from commercial organizations [56]. These organizations include product manufacturers, marketing and advertising agencies. Opinion Aggregation plays an important role in this field, because it has the potential to capture the opinions of the community. However, it also has some weak points, such as the smoothing of the variances in opinions and the possibility to manipulate the aggregate values by introducing artificially constructed data. This makes opinion quality assessment and spam detection useful pre-processing steps for Opinion Aggregation.

The first problem, opinion quality assessment, aims at determining the quality of opinions expressed in a review. [86] describe an opinion quality classifier relying not only on the review's textual features, but on the reviewer's social context, as well. The authors propose a method that optimizes an error function for training data in a feature space, subject to four regularization constraints. These constraints capture the intuition that the quality of reviews from the same user, as well as from users connected in a social context to that one, should be about the same. The introduced constraints do not employ annotated labels, therefore, may be used to train a model on unlabeled data. The study shows that the proposed method increases the accuracy of identifying reviews of high quality.

At the same time, we observe that the phenomenon of opinion spam (or fake reviews) is also growing [62, 20, 79]. The detection of opinion spam is a hard problem, since spam is targeted to specific products (therefore, resistant to aggregation), and not easily distinguishable from real reviews. This problem had not been studied in depth until recently. Below, we briefly discuss few of the papers in this area that are relevant to Subjectivity Analysis. The aim of these studies is to identify opinion spam in a pre-processing step. Then, the review spam can be excluded from further consideration, thus, resulting in more accurate and truthful Opinion Aggregation.

The work of Lim et al. [79] proposes a method for detecting spammers, rather than individual spam reviews. Each user is attributed with the following statistical measures: *Rating Spamming*, which is the average similarity among the user's ratings for each product; *Review Text Spamming*, which is the average similarity among the user's review texts for each product;

Single Product Group Multiple High (Low) Ratings, which is the number of extremely high (low) ratings posted by the user for each product group in time intervals where such quantity exceeds a certain threshold; *General Deviation*, which is the average deviation of the user's ratings from the mean of each product; *Early Deviation*, which is the same as General Deviation, only weighted according to time. All the individual measures are normalized against all users, and the overall final measure for each user is computed as their weighted sum. To classify a user as spammer, one needs to compare the user's measure against some threshold.

Jindal and Liu [62] classify opinion spam into the following three categories: *untruthful opinions*, *reviews on brands only* and *non-reviews*. The first category, *untruthful opinions*, is represented by intentionally biased reviews, either positive or negative. The second category, *reviews on brands only*, consists of opinions about some brand in general, without discussion of specific product features. The third category, *non-reviews*, refers to explicit advertisement, technical data or off-topic text. If the last two categories can be discriminated from the rest of reviews by a classifier trained on user-annotated samples, the first one is very difficult to classify manually, resulting in no training samples and, hence, impossibility to use machine learning classifier. Therefore, authors exploit different methods for spam detection. Categories 2 and 3 are classified with Logistic Regression model trained on annotated samples and using various features. For this task, the authors report classification accuracy (represented by Area Under ROC Curve) being 98.7%. Category 1 is classified in two steps, according to the observation that most distinctive spam reviews are using repetitive texts and extreme ratings. Correspondingly, during the first step duplicate reviews are identified using bigrams and Jaccard similarity. During the second step, these duplicates are employed as samples of spam for training of the Logistic Regression model. The main purpose of such model, however, is not to discriminate duplicate reviews, but to identify spam reviews which, having no repetitions, are difficult to detect otherwise.

Li et al. [78] combine the two approaches to spam detection, proposing a semi-supervised co-training algorithm for user and review classifiers. Accordingly, the feature set is split into user and review features views, which are used to train the corresponding classifiers in a bootstrapping fashion. In the default implementation, the most confidently classified reviews from either of classifiers are added to the training collection. In the agreement implementation, only the reviews classified by both classifiers as spam are included for the next training iteration. Experiments demonstrate that both methods improve over the performance of a conventional NB classifier, also benefitting from training on a larger set of reviews. Moreover, the agreement implementation yields much better accuracy than the other approaches, indicating that only hybrid methods can be truly effective for spam detection.

2.5 Contradiction Analysis

By analyzing a community's opinions on some topic, we understand how people in general regard this topic. However, people do not always share the same opinions on different topics. We may be interested in focusing on the topics for which conflicting opinions have been expressed, in understanding these conflicting opinions and determining the objective reasons for the observed diversity, and in analyzing their evolution over time and space. Evidently, we need to be able to effectively combine diverse opinions in *ad hoc* summaries, and also to further operate on these summaries in order to support more complex queries on the dynamics of the conflicting, or contradicting opinions. Moreover, performing simple aggregations on such opinions is not enough for satisfying the requirements of modern applications. Opinion aggregation may produce a lossy summarization of the available opinion data, by ignoring and masking the diversity that inherently exists in data. This problem demanded novel sentiment aggregation methods being more robust than their predecessors. In order to find an answer to these interesting problems, we have to employ more advanced techniques. The corresponding problems of Sentiment Analysis, providing methods for this kind of complex analytics form the problem of *Contradiction Analysis*, an emerging research direction under the general area of Subjectivity Analysis, which we discuss in this section.

2.5.1 Problems in Contradiction Analysis

A typical Contradiction Analysis application needs to follow the same steps we identified for Opinion Mining, namely, topic identification and sentiment extraction. For certain techniques of Contradiction Analysis it is possible to rely directly on the output of Opinion Mining, thus simplifying the entire workflow. Then, we need to have a contradiction detection step, where individual sentiments are processed in order to reveal contradictions.

The first step can be accomplished using either IR (TF/IDF topic identification), Probabilistic Inference (latent Dirichlet allocation), or NLP (linguistic parsing) methods. As both these approaches have weaknesses, we believe there is a need for their composition. The opinion classification step may rely on NLP, statistical, or machine learning methods. Again, to obtain high performance on various types of data we need to study algorithms that combine ideas and techniques from all three approaches.

In the contradiction detection step, the goal is to efficiently combine the information extracted in the previous steps, in order to determine the topics and time intervals in which contradictions occur, their composition, level and evolution. In this step, statistical methods can be used, as well as clustering, or other unsupervised methods. The contradiction detection step requires efficient data mining methods, which will enable the online identification of contradictions, and will have the ability to work on different time resolutions.

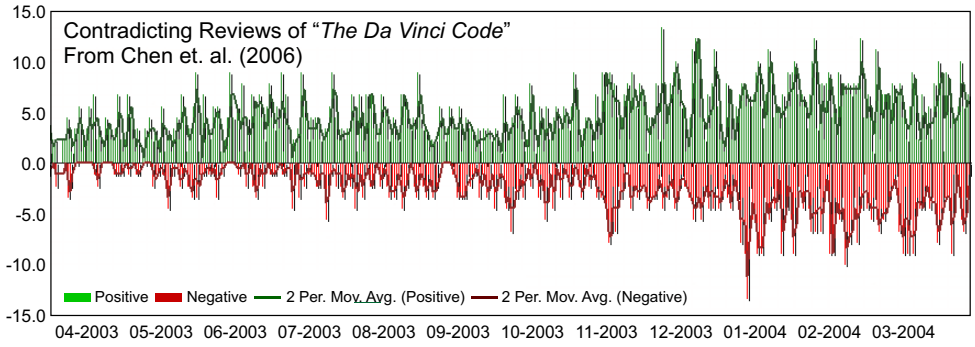


Figure 2.4: Opinion timeline visualization from [19].

As we already mentioned, there exist several problems connected to Contradiction Analysis. One of the problems is the detection of contradicting opinions, which must be approached with either unsupervised or semi-supervised methods to be useful for large-scale application. Another important problem is the automated extraction of contradicting summaries, that can complement to the detection step, providing in-depth overview of the interesting regions, highlighted by the detection. Finally, there exist a problem of management of diverse opinions, requiring special aggregation methods in order to preserve opinion diversity.

2.5.2 Development of Contradiction Analysis

As with all other Subjectivity Analysis problems, research on Contradiction Analysis is under way in different domains. We note that the problems and drawbacks we mention here only reflect that the domain is in the process of formulating its problems and shaping its methods, which further advocates the need for a more detailed theoretical study of the contradiction analysis problem.

A good example of contradicting opinions is presented in the study of book reviews by Chen et al. [19]. The main goal of their work is to classify reviews as positive and negative and to identify the most predictive terms for the above classification task. Apart from this, they aggregate and visualize the contradicting opinions over time, as can be seen in Figure 2.4. The visualization is composed by two trends of opposite (positive, negative) opinions, along with their moving averages. The user can determine contradicting regions by visually comparing these trends. However, such an analysis, which is based on manual inspection, does not scale and becomes cumbersome and error-prone for large datasets.

It is interesting to mention that the identification of contradicting claims first appeared in the speech recognition domain. The works by Hillard et al. [55] and Galley et al. [45] established it as a problem of recognizing agreement (positive) and disagreement (negative) texts, by looking at sentiments and negation. The authors exploited machine learning techniques for classification purposes, combining audio and text features.

Another approach to contradiction detection is to handle it as a textual entailment problem. There are two main approaches, where contradictions are defined as a form of textual inference (e.g., entailment identification) and analyzed using linguistic technologies. Harabagiu et al. [53] present a framework for contradiction analysis that exploits linguistic information (e.g., types of verbs), as well as semantic information, such as negation or antonymy. Further improving the work in this direction, de Marneffe et al. [89] define several linguistic features that contribute to a contradiction. Exploiting these features, supplemented by the sentence alignment tool, they introduced a contradiction detection approach to their textual entailment application [103].

Although the detection of contradictions using linguistic analysis and textual entailment promises more accurate results overall, the current methods do not yet achieve high precision and recall values [141, 47]. For example, Pado et al. [103] report their precision and recall values of contradiction detection at the RTE-4 task as being 28% and 8%, respectively. Therefore, scientists concentrate their efforts in finding contradictions of only a specific type when dealing with large-scale web analysis. In particular, they analyze negation and opposite sentiments.

Ennals et al. [36, 37] describe an approach that detects contradicting claims by checking whether some particular claim entails (i.e., has the same sense as) one of those that are known to be disputed. For this purpose, they have aggregated disputed claims from Snopes.com and Politifact.com into a database. Additionally, they have included disputed claims from the web, by looking for an explicit statement of contradiction or negation in the text. Although this approach would not reveal all types of contradictions, it can help to identify some obvious cases, which can be further used as seed examples to a bootstrapping algorithm.

The problem of identifying and analyzing contradictions has also been studied in the context of social networks and blogs. Relying on the exploited data mining algorithms, scientists proposed different measures for contradiction. Choudhury et al. [24] examine how communities in the blogosphere transit between high- and low-entropy states across time, incorporating sentiment extraction. According to their study, entropy grows when diversity in opinions grows. Their method uses the Mutual Awareness Expansion algorithm in order to extract groups, and defines a custom measure to characterize the social behavior of users, groups, and communities in the blogosphere. In particular, they rely on the two entropy measures calculated over the distributions of group sizes and group topics, respectively.

In some cases it is also interesting to examine how the blog entries of a single user change over time. The study in [90] focuses on the analysis of the sense-of-self sentiments of individual users, and how these change as a function of time. The difference in sentiment is measured as a distance in a two-dimensional space, where the first dimension is represented by the vector of kin words, and the second dimension is represented by sentiment words. However, they do not summarize the detected contradicting opinions or highlight their differences. It is up to the user to visually inspect the results and draw some conclusions.

A large body of work address the problem of contrastive opinion summarization [66, 85, 112, 41], which aims at extracting a short set of representative and diverse opinions from text collections, such as product reviews. Most often these methods consider that documents are annotated with sentiment labels during preprocessing.

A work by Liu et al. [85] introduces a system that allows comparing contrasting opinions of experienced blog users on some topic. They aggregate opinions over different aspects of the topic, which improves the quality and informativeness of the search results.

Kim and Zhai [66] propose to perform a contrastive opinion summarization based on the measures of *representativeness r* and *contrastiveness c*.

$$r = \frac{1}{|X|}\sum_{x\in X}\max_{i\in[1,k]}\phi(x,u_i)+\frac{1}{|Y|}\sum_{y\in Y}\max_{i\in[1,k]}\phi(y,v_i), \qquad c=\frac{1}{k}\sum_{i=1}^{k}\psi(u_i,v_i) \tag{2.6}$$

The first measure is based on the weighted sums of maximal content similarities, ϕ, among positive, X, and negative, Y, sets of sentences and their corresponding summaries, u and v. Representativeness reflects how well the summaries approximate the original text. Contrastiveness captures the similarity between positive and negative sentences in the summaries, but is computed based on the contrastive similarity ψ that is the same as content similarity, except that it is computed without taking into account sentimental words. Elimination of sentimental words results to improved precision for this similarity matching. Both ϕ and ψ rely on similarities among a review's individual words, either restricted to an exact match or a semantic (probabilistic) match. We note though, that extracting the same number of positive and negative sentences k may negatively affect representativeness, because of the different sizes of initial sets of positive and negative texts. Moreover, the contrastiveness function is calculated only on k pairs (first to first, second to second, etc.), instead of the k^2 possible combinations, which makes the default system's output dependent on ordering, thus requiring an optimization on a space of $|X|\cdot|Y|$ to come with k aligned pairs. This reminds us that contradiction measurement is inherently a quadratic problem over the size of a text collection.

Using the same principles, Paul et al. [112] propose to extract diverse opinions for opinion summarization using an extension to PageRank algorithm [14], which favors the transition of a random walker between representative and diverse opinions, improving their score.

Fang et al. [41] propose an opinion diversity metric for text summarization based on the differences in sentiment word distributions for different opinions, measured by Jensen-Shannon divergence. Since the latter is based on the average Kullback-Leibler divergence, which is the entropy of conditional opinion words distribution, it measures the average distance of opinion word distributions to their average.

The above systems operate on a set of sentences that is already divided between positive and negative texts. This may reduce the space of finding the optimal solution, since the interesting

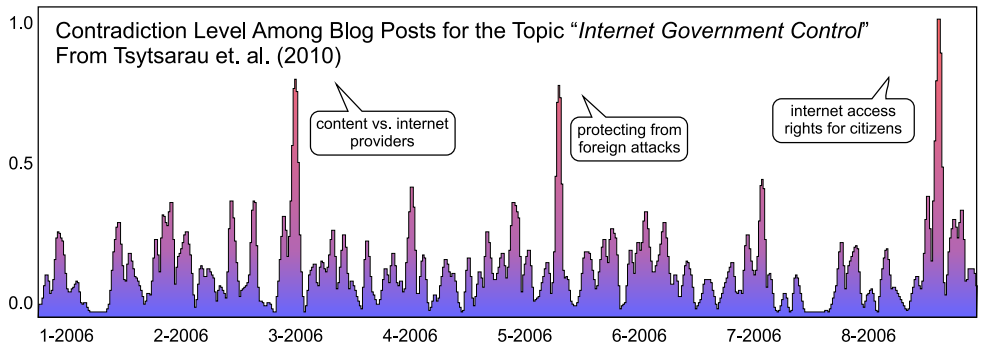


Figure 2.5: Contradiction timeline visualization from [133].

differences among sentences may occur not only at the sentiment level, but also at the levels of sentence structure and factual information.

We propose an automatic and scalable solution for the contradiction detection problem, focusing on the analysis of sentiments [133, 134]. Unlike all other contradiction measures, [134] is based on a joint modeling of positive and negative sentiment distributions, and is computable from aggregated sentiments. The proposed model captures the observation that when the sentiment mean is close to zero, while the sentiment diversity is high, then the contradiction should be high. An example result of such an analysis is represented in Figure 2.5, which depicts the evolution of the contradiction level for the topic "internet government control", covering a time period of about one year. The graph shows the peaks in contradiction for this topic, enabling the analyst to focus on the interesting time points (and the corresponding documents) along the time interval.

Contradictions may occur not only on the sentiment level, but also on the topic level and on multi-polar opinion level. Dealing with this problem requires determining information clusters and their interaction. Following this representation of opinions, we propose a model and a measure for opinion contradictions [131], which relies on aggregate statistics of clusters. In particular, the criteria for contradiction relies on a measure for cluster separation (usually optimized by clustering algorithms), and the measure of contradiction is based on the entropy of cluster size distribution. Similarly, Varlamis et al. [140] propose clustering accuracy as an indicator of the blogosphere topic convergence (Figure 2.6). Clustering accuracy (when represented by the utility function) measures the relative separation of cluster centers with respect to cluster sizes and a number of unclustered blogs (noise). When the clustering is very good, this function reaches its maximum value. It is easy to demonstrate, that divergence in topics leads to greater separation of individual blogs in the feature space and, therefore, less reliable clustering. By analyzing how accurate the clustering is in different time intervals, one can estimate how correlated or diverse the blog entries are. We note that this approach is relevant to the contradiction definition we gave earlier, in the sense that clustering is often defined as the process of finding

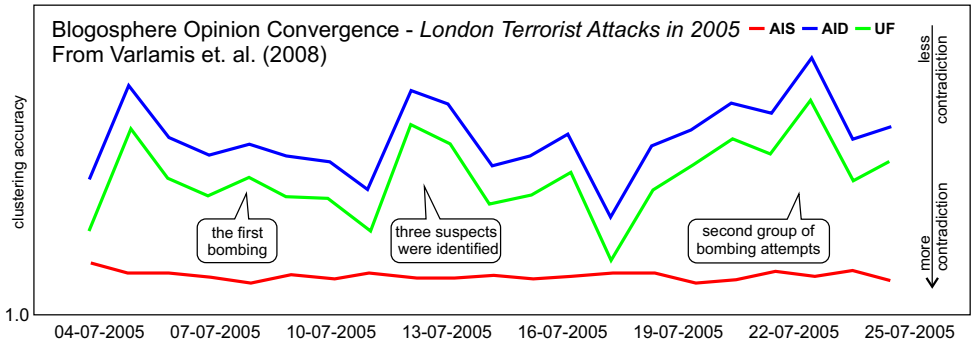


Figure 2.6: Blogosphere topic convergence from [140].

distant (i.e., contradicting) groups of similar (i.e., non-contradicting) items. However, the type of contradictions that this approach discovers depends on the selection of features.

Recently, sentiment change was studied in several publications addressing sentiment analysis in Twitter [113, 126, 88]. Popescu and Pennacchiotti [113] propose a hybrid approach of detecting contradictions in Twitter, which is based on a machine learning classifier trained on a rich set of textual and statistical features. Although this improves the precision over purely textual and purely statistical feature sets, selecting the right combination of features requires numerous training and adaptation stages for the classifier, especially for texts coming from different sources. In contrast, our approach is based on statistical principles and intended for a large-scale operation, where pairwise comparisons or any kind of linguistic analysis of texts may not be computationally efficient. In addition, we are considering a time dimension for contradiction, which allows us to introduce such new types as, for example, change of opinion (asynchronous contradiction).

Thelwall et al. [126] evaluate how twitter sentiment and its volume is changing before and after news events. By analyzing the peaks in sentiment, they show that the volume of negative sentiment is increasing just before an event, while there is an increase of positive sentiment at the event's peak intensity. These results indicate that changes in general twitter sentiment are mainly caused by external events and thus comprise rich data for sentiment analysis. One more observation made by the same authors is that the changes in sentiment are particularly small, making it necessary to apply more sophisticated methods capable of detecting them under high noise conditions. Mandel et al. [88] revealed sentiment differences among demographic groups and a necessity to account for classification errors (sentiment noise) and sentiment biases, thus stressing the need for our work, which addresses both of these problems.

2.6 Topic and Opinion Dynamics

Usually, a particular information source covers some general topic (e.g., health, politics, etc.) and tends to publish more material about this general topic than others. Yet, within a general topic, the author may discuss several more specific topics[4]. Being able to identify the specific topics is vital for the successful analysis of sentiments, because sentiments are attached to them and become their traits.

2.6.1 Topic Identification

Available topic representation and detection methods can be divided into several categories: 1) keyword or keyphrase extraction, 2) lexicon lookup, and 3) topic modeling.

Keywords, phrases and characteristic words used in a document can be considered as topics [144]. There are supervised and unsupervised algorithms available for keyword extraction. Good performing supervised methods are Naive Bayes, Decision Trees and Support Vector Machines. But, these approaches are limited to the extraction of known keywords. The TopCat system [27] exploits natural language processing techniques to identify key entities in texts and then forms clusters with a hyper-graph partitioning scheme.

Most of the works in Subjectivity Analysis assume a set of predefined topics when determining sentiments. These topics are specified either by keywords, or by restricting the collection of documents to only those that mention the chosen topics. In other words, the algorithms operate on the implicit assumption of a *single document - single topic* context. This situation changes when it is necessary to analyze sentiments expressed in free-form texts (e.g., weblogs), which may involve several topics. To solve this new problem, *single document - several topics* context, these methods should be extended with topic identification algorithms. Stoyanov and Cardie [121, 122] present an approach for opinion topic extraction that relies on the identification of topic-coreferent opinions. They use a lexicon look-up to determine product names, person names and the like for topic identification. Nevertheless, determining more general topics, e.g. political topics, requires a more careful modeling of topic - keyword interaction. One of the possible approaches for this kind of topic representation relies on the probabilistic generative modeling, as described below.

Probabilistic topic inference models, such as Latent Dirichlet Allocation (LDA) introduced by Blei et al. [11], consider documents as mixture of topics. Each topic is represented by a set of keywords together with a probability indicating the word's contribution to the topic, and words in a document are sampled in a generative process. LDA has been used in different application scenarios, not limited to topic detection for free text. In [10], LDA is exploited to

[4]From here on, we will refer to specific topics simply as "topics".

distinguish spam Web sites from non-spam Websites. Xing et al. [150] propose an approach to fraud detection in telecommunication based on topic models.

Nevertheless, associating the appropriate semantically consistent topics with the identified sets of keywords is sometimes difficult, as can be seen in Table 2.1. In this table, we demonstrate the topics extracted using LDA for Slashdot and WebMD datasets, when limiting the number of topics to 20 and 200. We observe that LDA-200 is able to identify topics that are more specific, while LDA-20 extracts more general topics, which are sometimes less comprehensible. A more detailed evaluation of LDA accuracy on noisy datasets can be found in[33], while for the purposes of this study we can summarize that LDA-based approaches are more applicable in the context of unknown domains, where lexicons can not be defined in advance or when topic terms are not explicitly mentioned.

Generative topic models have been also applied to Sentiment Analysis [91, 129, 80, 143, 111, 112, 41], intending to extract sentiments or contrasting opinions. For that purpose, topic models were extended with variables that model sentiment words, otherwise unrelated to topic detection. More specifically, these models describe distributions of words which represent probable topics, as well as distributions of words expressing positive or negative opinions towards them. Mei et al. propose a Topic Sentiment Mixture model [91], where every word in a document is considered as being related to one of the topics and to either positive, negative or neutral opinion classes (a word is sampled from topic-dependent and opinion-dependent word distributions). Positive and negative opinion classes can be considered as independent topics which can be additionally attached to a word apart from main topics. Extending this representation with the arbitrary dependency of words on topic, aspect or background, Paul and Girju propose Topic-Aspect Model [111], which they later show to be useful for opinion summarization [112]. Alternatively, Lin et. al. propose a Joint Sentiment Topic model [80], which assumes that word's topics are dependent on its sentiment. Jo and Oh extend this model to work on a sentence level, proposing Aspect and Sentiment Unification Model [63], which demonstrates better accuracy. However, considering that different topics have different (contextual) sentiment lexicons, it is more natural to model sentiment words being dependent on topics, and not otherwise. Accordingly, the Sentiment-LDA model by Li et al. [77] and Cross-Perspective Topic model by Fang et al. [41] represent opinion words as being derived from an opinion word distribution conditioned on a topic. This allows to assign different opinion words for different topics, contextualizing them. Topics, however, are sampled from the distribution for the entire document, rather than for each sentence individually. Li et al. [77] and Titov et al. [129] additionally propose local dependency extensions for their models at word- and sentence levels correspondingly.

Topics for LDA-20 (Slashdot)	
1	china chinese people
2	country american people
3	free market people
4	internet government control
5	companies company money
6	article read gov
7	bush court president
8	day election war
9	people car fact
10	law laws case

Topics for LDA-20 (WebMD)	
1	bad cancer pain
2	clinical drug research
3	child health care
4	medical surgery doctor
5	heart disease blood
6	fat food grams
7	based air advice
8	tea cup green
9	time day sleep
10	body brain cell

Topics for LDA-200 (Slashdot)	
1	companies market internet
2	congress law constitution
3	bush administration clinton
4	argument free copyright
5	anti god evolution
6	access cd music
7	attack iraq war
8	government amendment
9	china economy people
10	people civil democracy

Topics for LDA-200 (WebMD)	
1	knee replacement pain
2	eye surgery vision
3	fat grams calories
4	healthy diet fat
5	blood heart disease
6	breast cancer age
7	depression stress
8	birth control credit
9	ear infections infection
10	coffee caffeine food

Table 2.1: Top ten topics identified for Slashdot and WebMD datasets.

2.6.2 Topic Dynamics

A traditional approach in obtaining trends for popular items in blogosphere is to track user support for a set of popular keywords, i.e., measuring the frequency of keywords. Glance et al. describe BlogPulse [49], a system for identifying trends in weblog entries, that relies on the extraction of key phrases, person names and key paragraphs. This method uses frequency as a measure of popularity and relevance, but does not focus on how opinions may vary. Chi et al. [22] introduce a Singular Value Decomposition method for the analysis of trends in topic popularity across time. For example, using this method, we can detect which blogs contribute most of the effort in topic promotion by analyzing the first eigen-value. By using higher-order eigen-values it is also possible to identify blogs that are not affected by the main trends, but rather behave independently. In some cases it is also interesting to examine how the blog entries of a single user change over time [90]. This study focuses on the analysis of the sentiments of individual users, and how these change as a function of time.

2.7 Discussion

In this section, we elaborate on the emerging trends, compare the various methods that have been proposed for Subjectivity Analysis, and list open problems and interesting future research directions.

2.7.1 Analysis of Trends

We now discuss some trends that emerge when analyzing the recent publications on Opinion Mining (for a complete list of these papers, refer to Table 1).

We allocate the papers to several classes under different dimensions: based on the employed algorithms, datasets used for testing, and target domains. In Table 1 we list several of the properties of the papers we used for the above analysis, providing a more detailed view of these studies. Here, opinion classification and opinion aggregation types are denoted by *C* and *A* correspondingly. Column *"Topic"* lists whether algorithm uses topic-specific features, linguistic parsing, domain knowledge or other techniques that allow topic-dependent analysis. Column *"Range"* lists number of the resulting sentiment categories, or *C* in the case of continuous range. Column *"Scope"* represents target domains (and subdomains) for each algorithm, which were either explicitly mentioned by the authors, or inferred from the training and testing data used in the corresponding papers (*M* - movies, *P* - products, *B* - books, *S* - various services, e.g. restaurants and travels, *A* - all or indifferent; we note that for some of the papers we reviewed this detailed information is missing). Column *"Data"* lists one or more used datasets, which are listed in Table 3. Finally, column *"Scale"* represents a characterization of the algorithm (*S* - small, *M* - medium, *L* - large) with respect to its performance and adaptability as follows. Specialized algorithms, or algorithms with high complexity (e.g., sophisticated NLP tools) were classified as small scale. Algorithms, featuring moderate performance were assigned to medium scale. Finally, we classified as large scale those algorithms that are scalable, work in an unsupervised way or may incrementally adapt as they process more data. We note that, even though this classification may not be absolutely objective, it is still useful in order to reveal some interesting trends.

In Figure 2.7, we depict the distribution of papers (using stacked bars) along the most popular types of algorithms and sentiment representations. We observe that the majority of the publications use machine learning methods as the classification tool of choice. Next to them are the dictionary-based methods. Under this category, we also include corpus statistics and semantic approaches. Hybrid methods that combine the above approaches (usually a combination of dictionary methods with NLP tools), are not that popular yet, probably due to their high complexity.

Regarding the representation of sentiments, the alternative approaches are to use a binary representation (i.e., two classes, positive and negative), discrete (i.e., more than two classes; the algorithms we examined used up to six), or continuous (i.e., sentiments represented using scalar values) (refer to Figure 2.7). Most of the approaches in the literature use the binary representation. Though, the other two representations have recently gained in popularity, since they offer finer resolution and level of control. The relatively low amount of studies featuring the discrete sentiment representation for hybrid and dictionary methods can be explained by

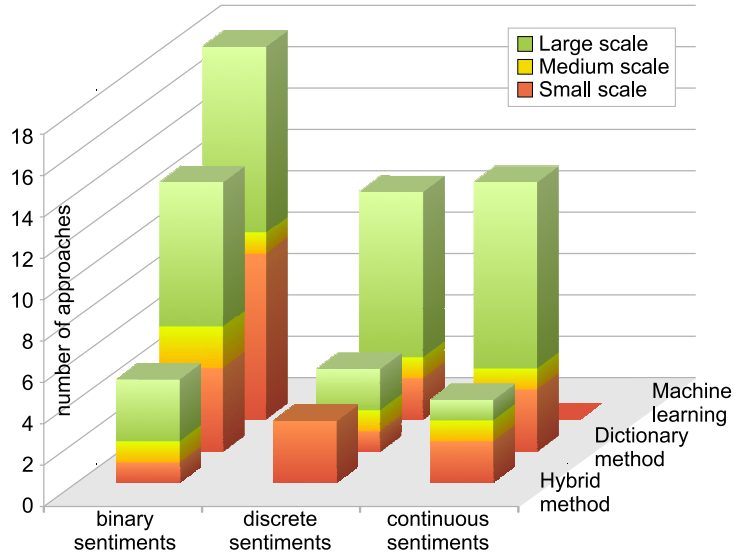


Figure 2.7: Number and scalability of methods over sentiment representation and algorithmic approach.

the availability of the continuous sentiment representation, which offers better precision. These studies use either the binary or the continuous representations, depending on their purpose. On the other hand, the continuous representation is not favored by the classification algorithms, making it a rare choice for the machine learning approaches.

The colors in each bar in the graph correspond to the number of algorithms capable of working with large, medium and small-scale datasets (green, yellow, and red color, respectively). This is directly related to the complexity of the proposed algorithms (e.g., there exist algorithms that operate only in a supervised mode, and evidently cannot scale with the dataset size). The graph shows that there are mainly two approaches that favor large-scale operation, namely, dictionary methods on continuous scale, and machine learning methods with binary and discrete representations. However, their popularity comes from different sources. Dictionary methods have the ability of unsupervised rule-based classification, which is simple and computationally efficient. On the other hand, machine learning methods achieve superior results and domain adaptability by paying the cost of the training phase. Nevertheless, they remain competitive in terms of computational complexity for the inference task (after the classifier has been constructed).

Figures 2.8 and 2.9 show the evolution of the scalability of the approaches proposed in the literature over the last years, as well as the application domains on which these approaches focused. We observe that at the beginning the majority of the studies analyzed review data, mostly at a large scale. As we mentioned above, the machine learning tools were the main contributors to this trend. The use of NLP methods since 2006 opened a new trend of complex review analysis, yet only on small scale datasets, due to the computational complexity of these

methods. At approximately the same time, another interesting pattern emerged, namely, the analysis of news and social media. The current trend shows that social networks and online sources of information are attracting increasingly more interest in the research community.

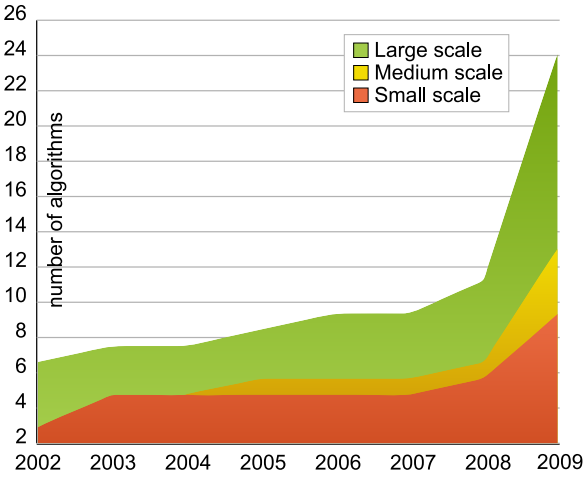


Figure 2.8: Number of algorithms with different scalability levels over the last years.

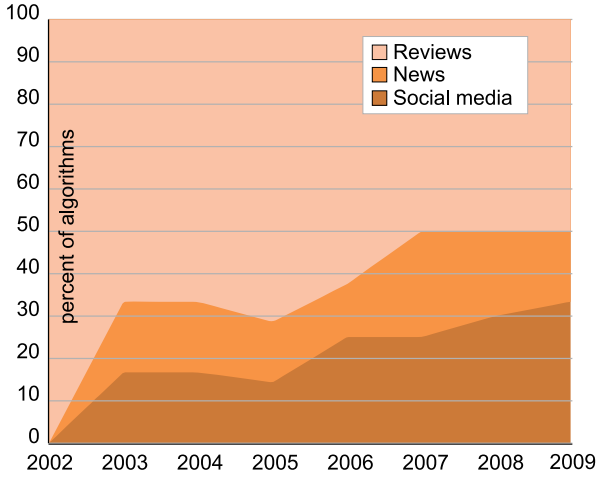


Figure 2.9: Percentage of algorithms targeting different domains over the last years.

2.7.2 Comparison of Methods

As can be seen in Figure 2.7, dictionary and machine learning approaches attract most of the attention in the research community. They have been evolving in parallel since the beginning of this decade, and it comes as no surprise that studies have started to compare their performance on different datasets. Below we present the most interesting comparisons and briefly discuss their results. A complete list of performance evaluations is reported in Table 2.

Dictionary methods are generally considered as inferior to machine learning, mainly due to their frequent use with the simplest bag-of-words models. However, the most robust linguistic-based approaches easily achieve a better precision than Machine Learning, thanks to a proper modeling of opinion patterns and dictionary adaptation. Nevertheless, the recall (the fraction of extracted sentiments) of these methods is lower as they require error-free sentences with recognizable structures. Despite a complex text processing, the computational complexity of such methods can be kept on a reasonable level with the help of efficient keyword-based filtering of phrases that do not contain the analyzed topics and sentiments.

Chaovalit et al. [18] performed an evaluation between the N-gram classifier and statistical approach methods on a dataset of movie reviews. In particular, their study showed the machine learning precision ranging from 66% (on the unseen data) to 85% (with 3-fold cross-validation), making it comparable to the 77% precision achieved with the unsupervised dictionary method.

Gindl et al. [48] compared the precision between various dictionary and machine learning

methods on web datasets (Amazon, IMDb, and TripAdvisor). The results demonstrated the superiority of the machine learning methods over the dictionary methods on all three datasets. The best results were achieved by the ME method, whose precision was in almost every case greater than 80%.

Another comparison between the most popular types of algorithms for sentiment extraction was made by Annett and Kondrak [3], demonstrating that some semantics-based algorithms are able to keep up with machine learning methods in terms of precision, even though they do not require a computationally-demanding learning phase. In particular, a lexical algorithm utilizing WordNet polarity scores achieved a precision close to that of decision trees (60.4% versus 67.4%). Nevertheless, these algorithms do not substitute, but rather complement each other.

As was demonstrated by Prabowo and Thelwall [114], only a combination of different kinds of classifiers is able to achieve a solid performance. In order to build their hybrid approach, they combined several rule-based linguistic classifiers with a statistical approach method and an SVM classifier. Doing so, they achieved a performance ranging from 83% to 91%, depending on the dataset. Another efficient way of collaboration between these approaches lies in using their mutual benefits. For instance, [17] rely on the highly-precise output of a linguistic classifier to train a machine learning method, which is then used to handle texts not recognized by the former.

We also point the interested reader to other studies that compare the performance of various Sentiment Analysis algorithms on different datasets [114, 18, 3]. However, a systematic comparative study that implements and evaluates all relevant algorithms under the same framework is still missing. Note that the performance results reported in Table 2 are not directly comparable to each other, because the evaluation framework and testing methodologies are not the same across the corresponding studies.

2.8 Conclusions

In this chapter, we presented an overview of a special class of web mining algorithms, that of Subjectivity Analysis. This is an area that started developing in the last years, and attracted lots of attention, because of its practical applications and the promise to uncover useful and actionable patterns from unstructured web data.

More specifically, we reviewed the most prominent approaches for the problems of *Opinion Mining* and *Opinion Aggregation*, as well as the recently introduced *Contradiction Analysis*. These have emerged as important areas of web data mining, and the trends of the past years show an increasing involvement of the research community, along with a drive towards more

sophisticated and powerful algorithms. Our survey reveals these trends, identifies several interesting open problems, and indicates promising directions for future research.

The mining and analysis of opinions is a challenging and interdisciplinary task, which requires researchers from different domains to consolidate their efforts. A typical solution in this area requires fast and scalable information retrieval, text preprocessing and topic assignment, in order to run machine learning algorithms supported by the possible use of NLP tools.

We observe that both the performance and resolution of the Subjectivity Analysis algorithms have increased over time. The first algorithms that were proposed in the literature were effective at discriminating between two or among three classes of sentiments. As we mention in Section 2.3.2, switching to several opinion classes required a redesign of the employed machine learning methods [106], while continuous sentiment values are only obtainable by using dictionary-based methods. Based on this, we foresee that the increasing demand for the quality of sentiments will require the development of new methods that will inherit strong features from both the machine learning and the dictionary-based methods.

Nevertheless, we note the lack of benchmarks in this area, which would greatly help its further development. Even though some datasets annotated with sentiments are available, they do not have the required precision and resolution. This problem becomes even more obvious for the most recent algorithms and applications, such as Contradiction Analysis. In Table 3, we list the various datasets that have been used for Subjectivity Analysis (mainly Opinion Mining). Regarding the contradictions between natural-language texts, the research in this direction is supported by the RTE challenge[5], which initiated a three-way classification task in 2008. In addition to the two-way classification between entailment and non-entailment, this task includes detection of contradiction as a part of non-entailment classification.

Finally, we note the need of a consistent framework suitable for working with subjective data, which treats the diversity of sentiment as a first-class citizen. Contradiction Analysis can possibly be the most demanding field for such a framework, as it utilizes most of the opinion mining methods, and at the same time defines its problems on data of various types, ranging from opposite sentiments to conflicting facts. We believe that this problem encompasses most of the challenges relevant to Subjectivity Analysis, and can be used as a reference target for the development of the framework mentioned above.

[5]http://www.nist.gov/tac/2010/RTE/index.html

Chapter 3

Problem Formulation

In this chapter we give an overview of problems connected to large scale sentiment analytics, discuss possible ways to address them and provide a high-level view at major components of our framework. We formalize and solve the particular problems of these components in the subsequent chapters.

3.1 Introduction

Large scale sentiment analytics usually operates with aggregated data. However, traditional approaches of data aggregation are unaware that sentiment data is polarized and thus it can be very sensitive to aggregation. We argue that not only the average sentiment is a poor representative of a real opinion, but also it can be ambiguous. For instance, if the average sentiment has a near-zero level, this can happen either when all documents are neutral, or when all of them are polarized. In other words, aggregated sentiment bears no information about the polarization of opinions. This problem also cannot be addressed by aggregating positive and negative sentiments separately, since their averages do not quantity the number of corresponding sentiments and thus are incomparable. Therefore, there is a need for novel techniques that will summarize and analyze the sentiment information in a principled and systematic way, which preserves and quantifies the diversity of sentiments.

In order to extract meaningful patterns and produce a desired output, sentiment analytics requires processing significant amounts of data and special methods that can exploit this volume to improve the resolution and representativeness of their analysis. Conventional sentiment aggregation methods may not be suitable for large scale analytics when subsequent time intervals contain different amount of sentiments and when a simple average of these sentiment values is taken. We observe the need of methods which can estimate sentiments from noisy and irregular samples, perform a meaningful aggregation with respect to the volume of data, and recover missing values.

Large scale analytics also requires efficient sequential time series access methods with a possibility of hierarchical navigation over time. However, the databases commonly used for sentiment storage and access, are not optimized to track the evolution of sentiments on a large scale or to support fast update rates.

3.2 Problems

In this work, we propose a framework for large scale aggregated sentiment analytics which addresses the problems of detecting interesting contradictions and changes of aggregated sentiment, evaluating their demographical composition and sources, and detecting relevant news events which could have caused these situations. Figure 3.1 outlines the composition of our framework. It consists of the three layers, namely, *Contradiction Analysis*, *Demographics Analysis* and *Dynamics Analysis*, which analyze aggregated sentiment data from different angles and serve the purposes of understanding and, ultimately, predicting sentiment changes.

The contradiction analysis layer takes care of aggregating sentiments for a topic and detecting interesting changes, which can be contradictions, outbursts of sentiments' volume or other changes in sentiment happening over time. For these purposes the layer uses various aggregated sentiment statistics, depending on the demands of the particular analysis.

The event dynamics analysis layer works with time series of news (sentiment) volume to detect various events and analyzes time series of aggregated sentiments for their possible causality. It performs event classification and dependency modeling in order to predict if a given event can cause shifts in sentiment, and for which topics and at what time this may happen.

Finally, the demographics analysis layer is intended to detect which groups of people formed a sentiment trend, by evaluating their biases and behavior with regards to a topic.

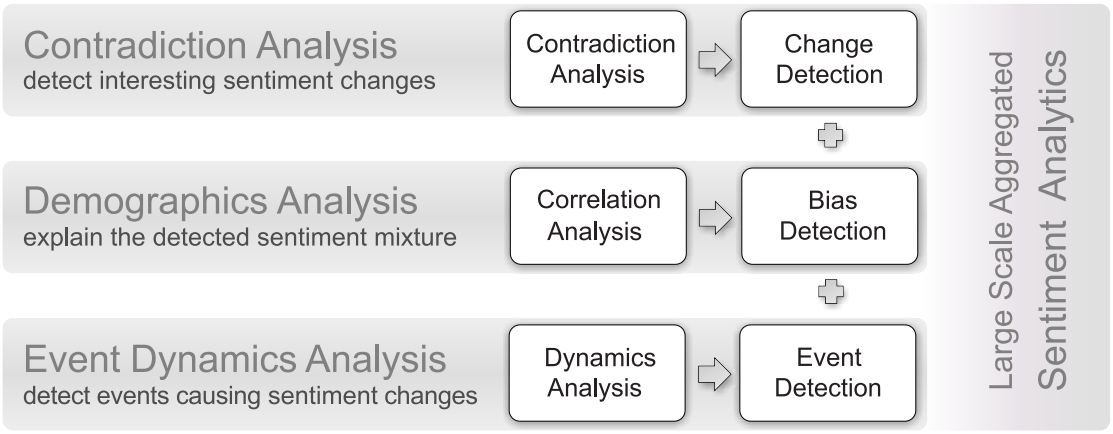


Figure 3.1: Compositional diagram of the proposed framework.

3.2.1 Contradiction Analysis

Although aggregated sentiments do contain some information, it may be incomplete. For example, if two opposite sentiment values are averaged, the result may have a neutral polarity. The information about either sentiment is then lost. The problem of summarizing diverse opinions has been studied within the scope of contrastive opinion mining, which extracts representative sentiments, which best describe opposite opinions. However, these sentiments are likely to capture the meaning of contradiction, but not its level. This problem essentially requires a consistent definition and new methods to deal with it.

The most challenging aspect of this problem is to provide a definition for opinion contradictions, which models an arbitrary number of conflicting opinions, and yet allows their effective comparison as well as the computation and storage of opinion distributions in an updateable and efficient way. A possible solution for this problem lies in adopting one of the existing clustering frameworks. Nevertheless, as we noted in our background evaluation, operating on a level of multidimensional opinions is not yet possible with the current extraction methods. When opinion classes are represented by polarized sentiments, it is possible to come with more efficient measures for contradiction, which allow detecting changes in the aggregate sentiment with suitable accuracy.

The particular problem we tackle is the aggregation of opinion and identification of shifts in sentiment for a given topic. Given the large number of data points available for a (topic, sentiment) pair over time, the key challenge here is to identify the time periods during which sentiment sees the increased diversity or changes its course. That itself will depend on the granularity of time windows during which sentiments are aggregated. When a small size of the window is used, the analysis retrieves local contradictions. By using larger windows, it is possible to identify changes of sentiment that are global with respect to a dataset. This means that in order to detect all possible contradictions, one needs to analyze sentiment time series at different granularities of aggregation. While it is possible to compute higher-level aggregates of the series dynamically from raw values, detecting changes with respect to the global sentiment level in this case can be done only after observing the entire sequence of values, what is not efficient on a large scale. This naturally calls for a solution which will store the *precomputed* sentiment aggregates at all granularities.

The output of this problem for a topic is a set of (time period, contradiction) pairs where time periods can be of different lengths, and even overlapping. For instance, we would like to detect a global shift of sentiment, which occurred during the year, as well as local situations in time which saw the increased diversity of sentiments.

3.2.2 Demographics Analysis

The first component of our framework should be able to perform effective sentiment aggregation either overall, or for particular groups of users. While the ability to determine average sentiment for a demographic group provides insights into a single group, it is not very useful when comparing multiple groups, since their sentiments may coincidentally approach the same level during particular time intervals, while being very different on average. Hence, our second problem considers a topic and a time period and looks for group pairs that are either in agreement or disagreement on the input topic throughout all events within the input period. More specifically, we define two goals: 1) identifying demographic groups with correlated sentiment, and 2) detecting moments in time, when demographic groups deviate from their usual behavior.

For example, this problem applied to the topic "organic farming", would potentially determine that for a while, "French farmers" and "German farmers" agreed then they disagreed on the topic when the French government introduced additional taxes for organic goods in disfavor of French farmers. If we measured these two groups having equal average sentiments for the considered time period, it would be surprising to find this being the result of opposite sentiment deviations from quite different reference levels: "French farmers" were initially overly positive about the topic, while "German farmers" carried on with their more realistic sentiment. Detecting correlations or anti-correlations of this kind can be very useful for policy makers, since changes in opinion of people (as their reaction to various events) reveal more details about their similarities and differences, compared to static sentiment data often found in polls.

One important aspect in this problem is the definition of sentiment correlation as a function of aggregated sentiment over a time period, accounting for inherent biases of groups, and focusing only on their temporal evolution and differences. Therefore, we need to study and formalize various properties of correlation specifically with regard to sentiment data.

In addition, we need a proper way of representing demographic groups and their relations. Available demographical information is usually represented in the form of IP addresses, locations and user profiles, rarely - in the form of their interests or political preferences. This suggests to represent people by sets of attributes, which can be hierarchically organized. Thus, we consider working on a space of possible attribute combinations, forming a *lattice* by parent-children relations of attributes.

Needless to say that demographics lattices can be very large even at the coarsest level of attributes representation. Finding correlations between pairs of groups in this case quickly becomes an intractable problem due to quadratic dependency. Therefore, we need to develop a special way of finding a small but sufficient set of correlations, which explain the behavior of the majority of the population.

3.2.3 Dynamics Analysis

The objective of this part of our framework is the analysis of dynamics of aggregated sentiments and of sentiment volume. Moreover, we want to understand and model the relationships between identified interesting sentiment changes and news events.

Changes in community's opinion are usually driven by new evidence or by impacting events coming from news sources. To annotate sentiment shifts we need to analyze relevant collections of documents for a possible explanation. However, events are often not mentioned explicitly in texts along sentiments, and to recognize the cause of sentiment changes we want to navigate to a correlated news trend and analyze the volume of news around a sentiment change. Therefore, we need to study correlations between shifts in sentiment and events detected in different media.

One of the problems we want to address here is that not all sentiment measures and news trends are linearly correlated, since changes in aggregated sentiment are particularly smooth, while news trends are usually bursty time series. This problem becomes even more exaggerated in the presence of sentiment noise, when some changes are caused by noise rather than events. To address these challenges, we need to develop special correlation measures and perform robust causality analysis.

Moreover, not every kind of news events causes a shift in sentiments, and recognizing relevant events requires a careful modeling of their dynamics. To determine the importance of news to people, it is crucial to consider the publication volume of different social media, rather than only from news agencies or news media. Analyzing the aggregated publication volume on a specific topic over time can yield understanding event's importance. However, social media can contribute to this volume all by itself (without any external stimuli) and also maintain a trending volume growth over long time periods. These effects distract the observed events dynamics and may even make them undetectable. Thus, our main goal here is to detect and differentiate various types of dynamics.

Finally, by observing the dynamics of social media and their delayed reaction, we want to predict these changes as soon as we are able to recognize the establishing news trend.

Chapter 4

Sentiment Aggregation

This section develops on various functions of aggregated sentiment and their properties with respect to statistical significance and robustness to noise. We demonstrate that a fail-safe analysis of aggregated sentiment requires special methods of regression, which depend both on the variance and on the number of aggregated samples. Such measures lead to the necessity of storing multiple statistics of sentiment.

Since our ultimate goal is to support large scale sentiment analytics, we need scalable methods to insert, access and update these data. Therefore, in the second part of this chapter we turn our attention to the problem of organizing sentiment time series in a way that will allow their efficient access, updates and analysis in large collections of data that span very long time intervals.

4.1 Introduction

Efficient sentiment aggregation methods would only be possible if we will manage to introduce a suitable sentiment scale, which is also able to represent sentiments in a compact way. For the sentiment analysis problem, the choice of the continuous scale in the range of [-1,1] seems to be a natural one, as it easily accommodates the discrete opinion categories (-1,0,1), and at the same time provides flexible opportunities for various mappings from the rating scale (e.g., rating stars). However, for conflicting opinions there is no such obvious choice. We need to represent differences in opinions that can not be directly mapped to real values. For example, the pair "the cat is *black* - it is a *white* cat" that features an obvious contradiction, can not be represented using +/-1, as the set containing just two colors (*black, white*) is not complete - there might also be *gray, red* and others.

There exist several indexing methods which work with statistical aggregates of time series. Xia et al. [149] propose MVTree spatial indexing of data streams based on R-tree with mean and variance used to construct an index. Their approach is based on the assumption that mean

and variance of streams are less volatile and thus allow rebuilding an index less frequently compared to raw data. MVTree stores in its leaf nodes only the most recent values of data streams and approximate stream statistics, which are incrementally maintained (updated and devaluated) as new values come in. Our approach is conceptually orthogonal to [149] since it is indexing data streams for topics based on *time dimension*, while mean and variance can be precisely reconstructed from the aggregates stored at every node. In this respect our storage is related to the stream aggregation stage of CluStream [1], which uses hierarchically organized statistics to speedup clustering of evolving streams. However, our implementation is intended for time series storage over multiple topics and gains several important features necessary for fast operation. First, it supports fast linear readout by connecting adjacent nodes at the logical level and storing them sequentially at the physical level. Second, it improves the access performance through the use of paginated I/O and memory buffer management. Third, it applies batched updates and bottom-up update propagation for a more cost-efficient input.

4.2 Aggregation

Aggregated Sentiment We work with aggregated sentiment, which is given in the form of triples (n, M_1, M_2), consisting of the number of aggregated sentiments n, plus the first and the second momentum of sentiments M_1 and M_2, which are defined as follows:

$$M_1 = \sum_{i=1}^{n} S_i \quad \text{and} \quad M_2 = \sum_{i=1}^{n} (S_i)^2 \tag{4.1}$$

Sentiment Mean $s(t)$ and **Sentiment Variance** $\sigma(t)$ are defined as the time series of mean and variance of sentiments, taking values in the range of [-1,1] and [0,1] respectively:

$$s(t) = \frac{1}{n} \sum_{i=1}^{n} S_i \Big| (t) \quad \text{and} \quad \sigma(t) = \frac{1}{n} \sum_{i=1}^{n} (S_i - s(t))^2 \Big| (t) \tag{4.2}$$

Sentiment Volume $s^{+}(t)$ or $s^{-}(t)$ is defined as the relative amount of sentiments of the same polarity expressed in a particular time interval. This measure is essentially a sum (or count) of sentiments of the same polarity, indicating bursts of a particular kind of opinion, e.g. *positive*:

$$s^{+}(t) = \frac{1}{n} \sum_{i=1}^{n} (S_i \,|\, S_i > 0) \Big| (t) \tag{4.3}$$

Relative Sentiment Volume $s^{+}_{\mu}(t)$ or $s^{-}_{\mu}(t)$ with respect to the average level μ is defined as the relative amount of sentiments in a particular time interval which are higher (lower) than μ. This measure takes into account the bias of sentiments μ, which is often observed for sentiments

from demographic groups, where people share the same background or initial views:

$$s_{\mu}^{+}(t) = \frac{1}{n}\sum_{i=1}^{n}(S_i \,|\, S_i > \mu)\,\Big|(t) \tag{4.4}$$

Sentiment Derivatives. Since we analyze changes of aggregated sentiment, it is particularly interesting to consider direct measures of such changes over time. The first derivative of a sentiment time series can be either positive or negative, so it needs to be taken by modulus to derive a useful measure of interestingness. However, the increases of the absolute derivative indicate only one-way changes of sentiment, while in most cases sentiments are just temporarily deviating as a reaction to events, and then go back to their basic level. In such cases, we may consider the second derivative of sentiment.

$$s'(t) = \left|\frac{d}{dt}s(t)\right|, \quad \text{and} \quad s''(t) = \left|\frac{d^2}{dt^2}s(t)\right| \tag{4.5}$$

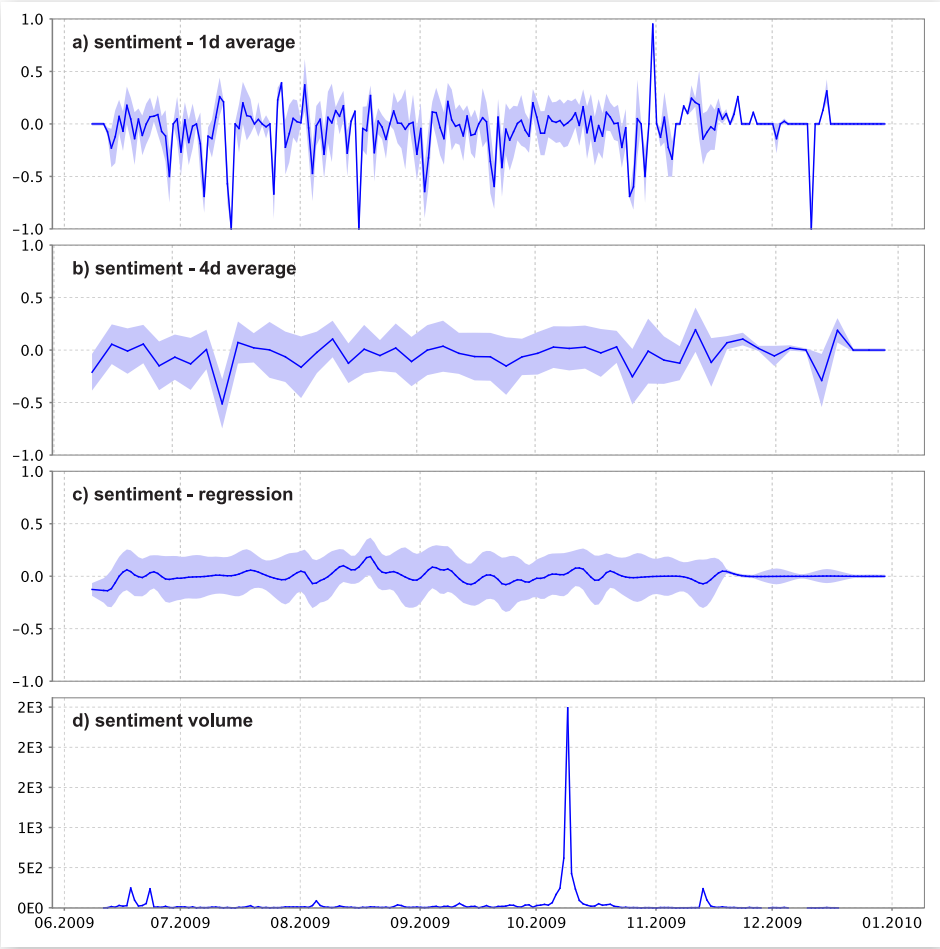


Figure 4.1: A time series from Twitter with various aggregations.

The above functions of aggregate sentiment are all useful as measures of interestingness. However, they do not account for the varying number of aggregated samples n, which is responsible for the significance of aggregated values. Consider for example a time series of sentiments extracted from Twitter, which we demonstrate in Figure 4.1. We observe that the average sentiment is very volatile during the time intervals of slow interest (series a), and even the increased aggregation granularity cannot cope with this problem, featuring several outlier points in the aggregate trend (series b).

The problem of statistical significance, however, is very important for real applications, especially for those working with slow-rate streams or with streams of sentiments with the irregular flow. Nevertheless, the majority of these applications either do not work with the significance directly, or do not able to integrate it into their methods. For instance, such problems as trend analytics, peak detection and shift detection do not naturally adopt the concept of significance.

We propose to cope with this problem by applying to our measures a special weight function W, which approximates the value of significance with respect to the varying number of documents. The weight function is defined as:

$$W = \left[1 + exp\left(\frac{\overline{n} - n}{\beta}\right)\right]^{-1} \tag{4.6}$$

where the constant $\overline{n}$ reflects the expected number of aggregated sentiments, and β is a scaling factor. This weight function provides a multiplicative factor in the range $[0;1]$, indicating the significance of aggregates (Figure 4.2 plots W as a function of n). Using W we can effectively limit our statistics when there is a small number of documents, and also weigh them more when the number of documents is large.

Nevertheless, there is still a problem of temporal irregularity of aggregated sentiments occurring due to sampling variation, which cannot be addressed by means other than sentiment regularization over time. To cope with this problem, we propose to use *local regression smoothing* [26], which computes a smooth regression trend with regard to sentiment observations and their weight. The regression trend ensures the continuity of sentiment values, but unlike sliding window based smoothing, it preserves the sharpness of significant sentiment deviations.

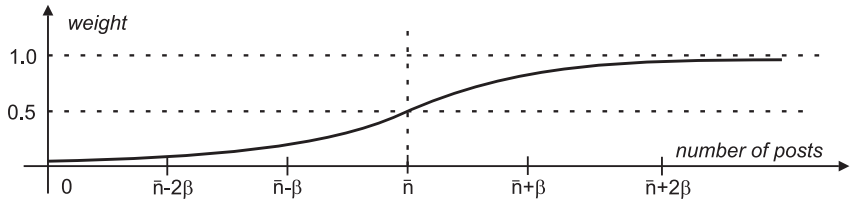


Figure 4.2: The weight function approximating the significance of aggregates.

Moreover, it achieves a more accurate sentiment trend, by considering sentiment variance, thus adding more weight to points with precise and significant values. In particular, we apply the cubic polynomial spline regression from SSJ library[1], which is based on smoothing cubic spline algorithm of Schoenberg. Nevertheless, other smoothing methods and their parameters can be applied depending on the nature of data and processing requirements.

The SSJ regression smoothing for a time series of n points is defined as the optimization problem over $n+1$ cubic polynomials $s_i(t)$, which are agreeing on their values and first and second derivatives at joint points:

$$s(t) = argmin\left[\rho\sum_{i=1}^{n} w_i(S_i - s_i(t_i))^2 + (1-\rho)\int_{t_1}^{t_n}(s''(t))^2 dt\right] \tag{4.7}$$

In the above equation, the regression parameter ρ is typically set around 0.5 (the middle value), and weights w_i of sentiment averages S_i are controlled by their variance and significance as follows: we take the significance weights W_i and sentiment variance σ_i^2 at sample points, and combine them with the global variance σ^2 to come with the following equation:

$$w_i = \frac{2\sigma^2 \cdot W_i}{\sigma^2 + \sigma_i^2} \tag{4.8}$$

Accordingly, this weighting technique favors significant samples with a smaller variance, and gives a zero weight for missing samples, allowing for their interpolation. Still, the right hand component of Equation (4.7) gives a uniform treatment to all samples while constraining polynomials deviation from the average. Nevertheless, we want the computed trend to have smaller deviations during time intervals with uncertain sentiments. To achieve this property, we add a discrete integrable function based on weights to the second part of Equation (4.7):

$$s(t) = argmin\left[\rho\sum_{i=1}^{n} w_i(S_i - s_i(t_i))^2 + (1-\rho)\int_{t_1}^{t_n}(s''(t)/w(t))^2 dt\right] \tag{4.9}$$

Our experiments with real data demonstrate that this method allows fail-safe analysis of sentiment time series even with high levels of noise and irregularity.

4.3 Problems

Our study reveals that recent sentiment aggregation methods are not designed to track the evolution of sentiments on a large scale. However, there is a need to address the problems of aggregating, managing, and analyzing sentiments in a large scale, and in an *ad hoc* fashion, much like the analysis opportunities offered by On-Line Analytical Processing (OLAP) in tra-

[1] http://www.iro.umontreal.ca/~simardr/ssj/indexe.html

ditional data management. However, the design principles of such a storage should answer to the needs of sentiment analytics, which we evaluate below.

To this end, we demonstrated the need to analyze sentiment information on each topic across different time windows. Assuming this requirement, scalability may be achieved by storing pre-computed values for windows of different sizes. The need to analyze time series at different time granularities calls for a hierarchical structure of the storage, like the one illustrated in Figure 4.3. In this example, the time windows are organized using days, weeks, months, and years (though, other hierarchical time decompositions are applicable as well). Using this kind of structure, aggregation queries can be answered on *adhoc* time intervals, by dynamically summarizing the values of covering nodes. The appropriate sentiment storage will need to support various time series access methods:

Single-granularity time series access: Sentiment correlation, evolution and change detection queries require temporal access of time series at different granularity levels.

Multi-granularity time series access: Contradiction, burst and anomaly detection require simultaneous access to child and parent time intervals.

Another important problem is how to efficiently organize the access to different topics. Topic-wise, there is a need to provide fast time series access for different types of queries:

Single-topic queries: This type of queries is necessary for different kinds of trend analysis and burst detection.

All-topic queries: Accessing time series or aggregates for multiple topics simultaneously is useful for correlation analysis and detection of trending topics.

One possible solution for the above problems is to use the proposed time-tree structure for each topic separately. This method allows to achieve scalability on the number of topics, and has a good performance when looking for values within a single topic. However, it involves high update costs, because for each document the data structure needs to be parsed as many times as there are topics in its text. In addition, it renders all-topic queries ineffective, because for each topic we need to navigate through a time structure in order to find the right interval. An alternative solution is to aggregate sentiment values for different topics under the same time-tree structure. This solution does not suffer from the disadvantages mentioned earlier, and is the solution of choice for this study.

In what follows, we describe two approaches that are based on the above observations, and we show how they can be used to identify and manage contradictions. In the first approach, CTree, we introduce a specialized data structure, which can be easily maintained in an incremental fashion when new documents are added to the system. Our second approach, DTree, extends on the previous version, adding the support for external time series storage, and data compression. Finally, we introduce Cdb, a relational database implementation of aggregated sentiment storage, which is used as a baseline for our evaluation.

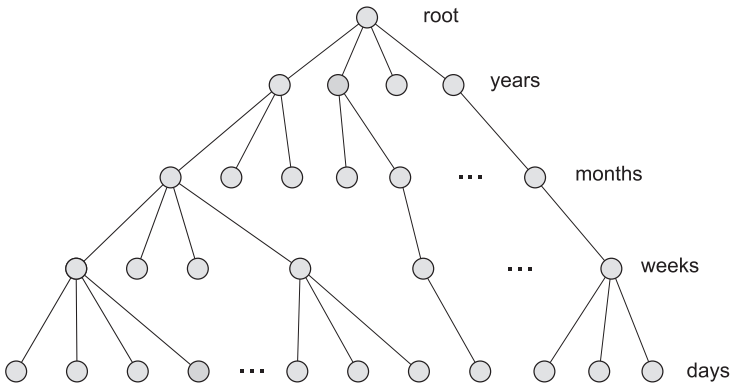


Figure 4.3: Logical structure of CTree.

4.4 CTree

We now introduce the Contradiction Tree (CTree) for managing the information on sentiments and their contradictions, which is incrementally maintained in an online environment, and can outperform a relational DBMS implementation by up to 3 orders of magnitude.

The CTree is organized around the sentiment moments, M_1 and M_2, and a hierarchical segmentation of time, as outlined in Figure 4.3. In this example, CTree has a calendar-like structure (i.e., the top level is years, the next level is months, etc.), while in general the levels of aggregation can be the same (e.g., every interval has no more than 10 children). In the following, we will refer to the levels of the CTree as the different *granularities* of the time decomposition, the root node having granularity 4 and leaf nodes having granularity 0. In the case when aggregation levels are fixed to contain m children, the depth of a tree d (that is granularity of a root node) can be computed as $d = 1 + log_m n$, where n is a number of intervals at the smallest aggregation level.

Each node in the CTree corresponds to a time window, and summarizes information for all documents, whose timestamp is contained in this time window. The internal structure of the CTree nodes is illustrated in Figure 4.4. As the figure shows, a CTree node stores the following information: (a) for each topic, the topic id, tid, the number of documents, n, on this topic that fall in the time window represented by the node (we only store information for topics when $n > 0$), and the sentiment moments, M_1 and M_2; (b) pointers to the children nodes (black dots); and (c) pointers to adjacent nodes, *prev* and *next* of the same level (black diamonds). The adjacent node pointers are used to allow fast sequential access to neighboring nodes in the same time granularity.

In our implementation, we assume that each node fits in a single disk page. This translates to each node being able to hold information for approximately 250 different topics (for the uncompressed implementation). In the case where a node cannot fit all relevant topics, we

can use additional storage, referenced by a special pointer in the CTree node (represented as a white dot in Figure 4.4). This solution allows us to accommodate a large number of topics at a small additional cost. Note that we can significantly reduce the expected cost of accessing this additional storage, by arranging the topics in a way that the most popular ones are located in the original node. For the purposes of this work we do not pursue this direction any further. Though, in the evaluation of our approach we report results with experiments that use this kind of additional storage.

Storage

CTree nodes may have different aggregation types to achieve the desired efficiency, not necessarily the same for the entire tree:

Raw Values. If CTree nodes store direct statistical sums of sentiments at different granularities, an aggregation can be done very quickly since stored triples are additive values. This requires statistical sums of enough precision (to support discrete updates) and capacity (to store large values). It is possible by using floating point numbers with a single or a double precision, which, however, occupy a considerable disk space.

Normalized Values. CTree nodes can store statistical sums of values normalized by volume and not exceeding the sentiment range $[\text{-}1, 1]$, which may have a more compact representation. This allows to store more topics in a single disk page, but requires the updated sentiment statistics being propagated from the bottom level of the tree.

Moreover, depending on application requirements, CTree can be implemented as a standalone storage, or as an index over the existing storage of time series, as demonstrated by the three CTree instances, shown in Figure 4.5:

A) Raw Storage, Raw Aggregates. CTree is implemented as a standalone storage with direct statistical sums of sentiments at different granularities. This is the default implementation, used in our experiments.

B) Raw Storage, Normalized Aggregates. A variant of the standalone storage where raw aggregates are located only at the bottom level. Since normalized aggregates in this case have the same weight, this storage is only suitable for streams with a constant rate of sentiments or with non-stochastic sentiment behavior.

B) Ext. Storage, Normalized Aggregates. This implementation keeps all the raw data in the external storage, which provides aggregates over multiple granularities. CTree maintains only materialized normalized aggregates of that storage, making possible to achieve a very compact index structure.

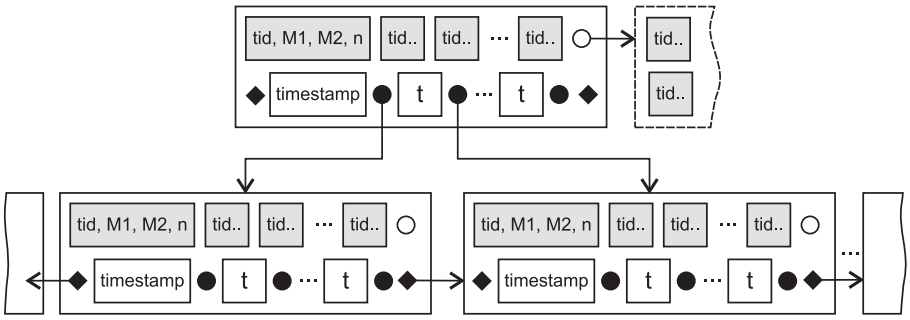


Figure 4.4: Physical structure of CTree nodes.

Table 4.1 summarizes the estimated disk page capacity in the case when stored values are encoded using (in bytes): triples 12 (raw values) and 3 (normalized values), topics - 4, pointers - 4. From the reported numbers it becomes clear that normalized values result in a considerable reduction of a space occupied by aggregates, which is even more pronounced when topic ids are not stored (this is possible when topic values reside in predetermined locations).

Kind of data **Capacity:**	**Raw Values**	**Norm. Values**
triples $(n, M1, M2)$	341	1365
topics + triples	254	407
topics + triples + page pointers	249	398

Table 4.1: Capacity of a 4K disk page for different data.

Updates

As discussed earlier, the additive nature of the sentiment statistics and the hierarchical organization of CTree nodes allow us to incrementally maintain the CTree in the presence of updates. When new collections or individual documents are analyzed, their contribution to the aggregates of the corresponding topics and time windows in the CTree can be easily taken into account by updating the set of relevant $\{n, M_1, M_2\}$ values in the nodes of the tree. However, this process is different for various CTree implementations, shown Figure 4.5. For the implementation A, CTree nodes are updated from top to bottom, as the update is being navigated to its appropriate time interval. For the implementation B, updates first have to reach the raw storage at the bottom, and only then normalized statistics are being propagated upwards. Implementation C, however, updates the external storage at every CTree granularity, and normalized values are obtained straight from this storage.

In order to reduce update costs and lower the possibility of precision loss for large volume statistics, we propose first to accumulate several updates and then submit them in a batch. When new documents arrive, as a preprocessing step, they are aggregated in time windows of the finest granularity of the CTree, by computing their count, as well as the topic sentiment moments M_1^T and M_2^T for each topic. Then, these aggregated values are used to update the counts and topic sentiment moments of all CTree nodes containing respective time windows.

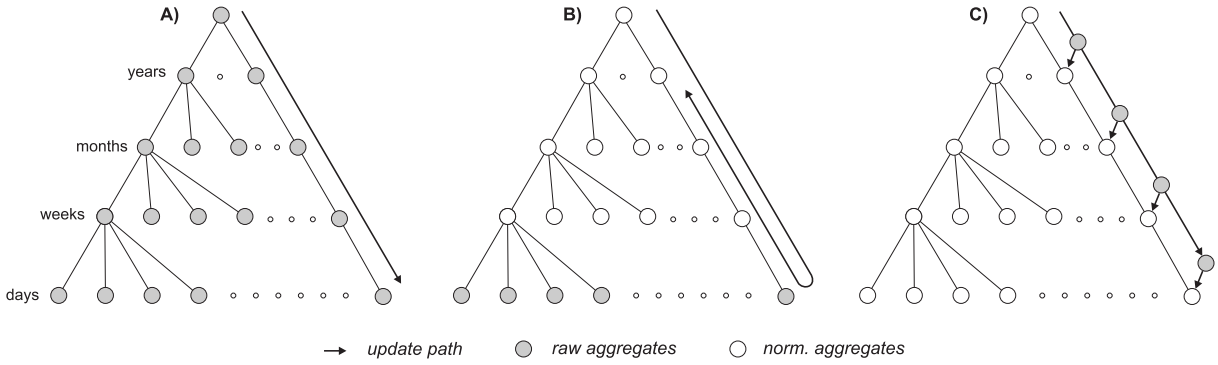


Figure 4.5: Update diagram for different CTree implementations.

The update cost for each batch of aggregated documents is measure by the number of IO operations (disk page reads / writes), which depends on the depth of the CTree, d, and the number of updated nodes, and in the worst case matches $O(d\frac{|T|}{h})$, where h is the maximum number of topics that a single node can hold.

The time complexity of random CTree access linearly depends on the number of nodes accessed to extract time series values. This number, in turn, depends on the size of the time interval, τ, the size of the time windows of the chosen granularity, $|w_l|$. It is also proportional to the number of topics that are stored in time windows of granularity l, which in the worst case is the number of all topics, $|T|$. Therefore, the time complexity is $O(\frac{\tau}{|w_l|}\frac{|T|}{h})$, where h is the maximum number of topics that a single node can hold.

4.5 DTree

To address the problems requiring fast ad-hoc navigation to sentiments for demographic groups, we extend the CTree structure with pointers to a physical sentiment storage, which at its nodes provides access to aggregated sentiment values via the demographics lattice. However, there exist two ways of organizing this storage, which are merely defined by different assumptions on the input data.

The first assumption is that there is a small number of demographics hierarchies, which remain fixed during all time periods - no new nodes being added or deleted. This assumption allows designing a storage with fast ad-hoc access to demographics lattice, since its nodes can be indexed by a simple enumeration. As an additional benefit, a time series of such values can be stored sequentially in one file.

The second way of organizing the data is based around the dynamic allocation of demographics nodes. Not only this allows to accommodate large demographics hierarchies, but also to add new hierarchies at any particular time. However, this approach requires storing a time series for each node in the demographics lattice as a separate sequence. Since the simultaneous

sequential access to time series is not possible in this case, the supported analytics becomes limited to that over time series of individual nodes.

In contrast to the demographics structure, that is being updated very infrequently, the time hierarchy is dynamic, since sentiments can be added to the recent time intervals (online updates) or to the past time intervals (offline updates). Moreover, the time dimension should provide fast sequential access either for computing the correlation or visualizing the time series in an efficient way. Therefore, it makes sense to have a fixed structure for the demographics lattice, in order to fit the requirement of simultaneous random access to individual nodes. It is also important to have a fast navigation between the coherent nodes as well. This can be achieved by using a special encoding of identifiers for demographics nodes, which preserves their proximity in the storage.

Algorithm 1: CTree and DTree Update

Input : Topics $\{T_i\}$, sentiments $\{S_i^T\}$, timestamps $\{t_i\}$
define `update` as a vector: (time interval τ, int n, float M_1, float M_2);
define `updateset` as a set $\{\}$ of `update` vectors;
Aggregate sentiments of each smallest time interval τ:
Let bucket $S_\tau^T = \{S_i^T \mid t_i \in \tau, \tau.granularity = 0\}$
$upd_\tau^T = (\tau, n^T = |S_\tau^T|, M_1^T = \sum S_\tau^T, M_2^T = \sum (S_\tau^T)^2)$;
foreach upd_τ^T **do call** `UpdateNode(`$rootNode$`,` upd_τ^T`)`;

function `UpdateNode(node` r`,` `update` upd_τ^T`)`;
if $r.childNodes \neq \emptyset$ **then**
  Let node $child = r_i \in r.childNodes \mid \tau \in r_i.\tau$;
  Set `update` $old = (child.n^T, child.M_1^T, child.M_2^T)$;
  Set `update` $new =$ `UpdateNode(`$child, upd_\tau^T$`)`
end
if $Norm.Values$ & $r.childNodes \neq \emptyset$ **then**
  //Subtract the old value, add the new value
  $r.M_1^T$ += $new.M_1^T / new.n^T - old.M_1^T / old.n^T$;
  $r.M_2^T$ += $new.M_2^T / new.n^T - old.M_2^T / old.n^T$;
else
  $r.n^T$ += $upd.n^T$; $r.M_1^T$ += $upd.M_1^T$; $r.M_2^T$ += $upd.M_2^T$;
end
return ($r.\tau$, $r.n^T$, $r.M_1^T$, $r.M_2^T$);

We name the new structure *Demographics Tree (DTree)*, and demonstrate in Figure 4.6. Similar to CTree, DTree is a hierarchically organized balanced tree, where each level in the hierarchy stores information relevant to years, month, weeks, and days. Each node in the tree corresponds to one of these intervals, and is connected to the parent and children nodes in the hierarchy, as well as to the adjacent nodes at the same level. Each DTree node stores statistical aggregations of sentiments for different topics for the specific time interval: *(count, sum, sum of squares)*$_t$, where topic $t \in \mathcal{T}$. These aggregations allow us to reconstruct sentiment mean,

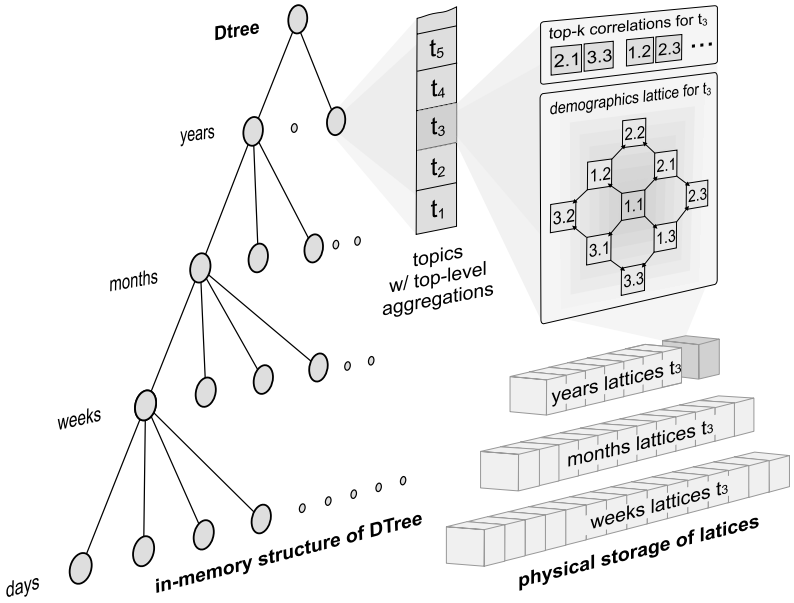


Figure 4.6: The DTree, a time-indexed sequential storage of aggregated demographics sentiments over multiple topics.

variance, volume and their derivatives, and they are also incrementally maintainable, allowing the easy update of the DTree as new data come in. In addition, DTree nodes store some precomputed information (e.g. top-k correlations) for the particular time interval and topic, in order to facilitate query answering. DTree nodes maintain physical aggregations only for the top-level demographic groups for each topic (e.g., only for group (1.1) in Figure 4.6). Detailed aggregations for all individual groups are accessible by following a pointer to a separate structure, the sequential file storage for lattices. This pointer indicates an offset in the file that contains the demographics lattice snapshot with the aggregations for all demographic groups for the particular topic and time interval. By traversing this sequential file storage structure, we can simultaneously reconstruct the sentiment time series for all demographic groups for a particular topic and time aggregation level.

Thanks to this layout, a time index with high-level aggregates and pointers remains compact and can be kept in main memory (Figure 4.6, left), while sentiment time series can be organized as a collection of individual files (Figure 4.6, right). The additional benefit of this organization is that it ensures fast sequential access for time series of sentiments.

The internal structure of DTree nodes is illustrated in Figure 4.7. Same as CTree nodes, DTree stores the following information: (a) for each topic, the topic id, tid, the number of documents, n and the sentiment moments, M_1 and M_2; (b) pointers to children nodes (black dots); and pointers to adjacent nodes (black diamonds). In addition to that, DTree nodes store for each topic a pointer to the top-k page (white diamonds, optional) and a pointer to demographics lattice (white squares).

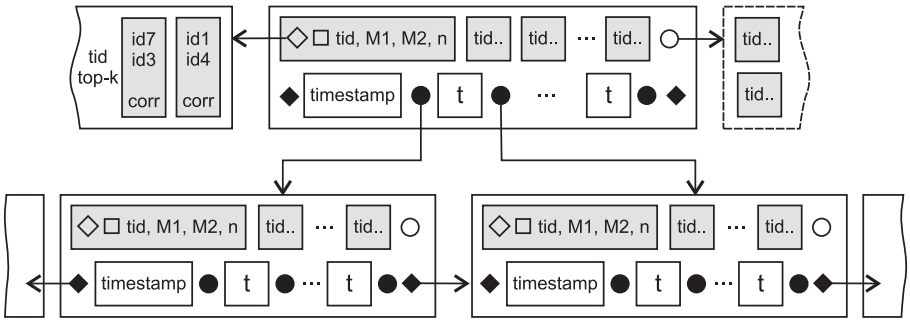


Figure 4.7: Physical structure of DTree nodes.

Modern hard disks can be very efficient at linear reading, and the size of the lattice should not affect reading times, as soon as the controller's capacity of transferring these data is not exceeded. However, sequential storage requires un-fragmented allocation of files in order to reduce disk seek time. Since the variable size of lattices on disk can result in the fragmentation of the time series when new data is inserted in the middle of the tree structure, the maintenance and defragmentation of the storage must be performed regularly using external tools. Nevertheless, the delayed updates are very rare in real-time systems (the data is arriving sequentially) and we can afford compressing the time series and using dynamic lattice storages. Since the number of sentiment values monotonically decreases as we navigate down the demographics lattice and down the DTree levels, many of the demographics leaf nodes will contain zeroes at lower time granularities. This allows storing sentiment values in a more compact way, by storing only non-zero values (e.g., using run-length encoding methods [142]).

4.6 Performance Evaluation

The performance evaluation was conducted on a desktop computer with a dual core CPU. Our algorithms were implemented in Java 1.6.13. The database we used for Cdb was IBM DB2 Express-C 9.5.2.

Baseline

In this section, we describe a baseline database solution for aggregating sentiments, which we call Cdb. The Cdb solution can be seen as a materialized roll-up of OLAP data cube, however the difference is that it is indexed. All the necessary information is stored in a single table, which uses the schema shown in Table 4. We populate this database table for each topic by inserting rows that correspond to time interval nodes of the hierarchical time tree structure (shown in Figure 4.3). Each row contains two timestamps, topic id and pre-computed statistical values for the relevant time interval and granularity. This implementation leads to simple and efficient SQL queries for accessing time series.

We do not use foreign keys to reference parent or children nodes in the tree structure. Instead, we use the combination of attributes *topicId*, *granularity* and time, which serves the same purpose more efficiently as long as we maintain proper indices. We created the appropriate database indices on the first four columns listed above (based on the performance profiling suggested for our queries), and left the logging and transactions functionality turned off. We used the same Java routine as in our application to calculate the contradiction level by attaching it to the database as a stored function. In queries, we refer to this routine as the *contradiction* attribute.

SQL code for queries that we use for evaluation is represented in Table 5. The queries are constructed assuming an adaptive threshold, therefore they contain self-joins on the table that stores materialized summary values. In our experimental evaluation, we use *Query 1* and *Query 2* to test the performance of Cdb. We do not report results for *Query 3*, since it is a heavy-duty query for a relational engine, involving three self-joins on a large table. (Nevertheless, using the CTree approach, *Query 3* shows almost equal performance to that of *Query 2*.)

Evaluation Methodology

We evaluate the scalability of our solutions, Cdb and CTree, for solving Problems 2 and 3. Remember that in the topic contradiction problem (2) we want to identify the contradictions and corresponding time windows of a *single topic* within some time interval, while in the all topic contradictions problem (3) we are interested in doing the same for *all topics*.

During this study, the parameters of the contradiction formula were at their default values as described in Section 5.3. Changing formula's parameters will enlarge or reduce the number of contradictions being detected, but the computational efficiency will be the same. Performance of our approach does not depend on the value of threshold because we are not storing precomputed contradiction values, and so the database is unable to apply indices or filtering on this parameter. Fixed and adaptive threshold approaches, however, return slightly different sets of contradictions. The first one returns largest contradictions themselves, and the second returns contradictions that are greater than p-times values of their respective parent intervals. The value of p was empirically set at 0.6 to return a result set with an average size equal to the one when using a fixed threshold. This allows us to compare the relative performance of both methods.

To test the performance, we generated sets of 25 queries for Problems 2 and 3, using granularities and topic ids drawn uniformly at random. Since DB2 uses a query cache, all the subsequent query executions are generally faster. In the results we report below, we do not include the "first run" while averaging the results. Thus, we are comparing CTree to the best Cdb performance we were able to achieve on this database.

In the first set of experiments, we measured the time needed to execute single- and all-topic

queries as a function of the time interval, τ, and the granularity of the time windows (Figure 4.9). We report results for both the fixed threshold and the adaptive threshold.

The adaptive threshold queries require in all cases more time since the threshold in this case has to be computed based on the contradiction value of the parent time window, which incurs more computation. This difference is pronounced for Cdb, because it involves an extra join for obtaining the parent time window. On the other hand, the same result in the CTree is achieved by following pointers, resulting in a minimal additional cost.

We observe that both single-topic and all-topics queries (see Figures 4.9(a-b)) scale linearly with the size of τ. This confirms our analytic results, and is explained by the fact that the queries have to return contradictions for all time windows (of a specific granularity) that are contained in τ. For single-topic queries with fixed threshold, the database is able to use all its indices (i.e., on topic, time windows, and granularity) to answer the queries, therefore, achieving fast response times. In all other cases (i.e., all-topic queries, or adaptive threshold), the CTree approach performs up to 3 orders of magnitude faster than Cdb. This pronounced difference is explained by the ability of CTree to access sequential time intervals without having to navigate through the index for each one of them - a situation taking place in the case of Cdb.

Figures 4.9(c-d) depict the time results when we vary the granularity of the time windows specified by the queries. Increasing the granularity translates to larger time windows (i.e., moving up in the time hierarchy) and a smaller number of time windows for the same time interval. Thus, response times get lower. Once again, we observe the same trends in the relative performance between CTree and Cdb as with the previous experiments on varying time intervals.

Performance of Indexing Sentiments

When new sentiments are being inserted into the DTree, they go through a number of levels in the tree index. First, the updates are aggregated in buckets, corresponding to the lowest time granularity. Second, the aggregated values are inserted in a single update, populating nodes of the index from top to bottom. For the DTree, updates are accumulated and inserted for each demographic group individually. Nevertheless, this method still improves the performance since disk pages are being accessed only once for each batch. During every update and at every granularity level, sentiment values are inserted into corresponding demographics lattices, for the specified group and *all of its parents*.

Demographic lattices can be stored on disk using different structures. The simplest of such structures is a *fixed array*, suitable for constant demographics lattices. It allows fast indexed access to the lattice values. The other is a *binary tree* of variable size and structure, which has reasonably fast $\log|L|$ access time and allows demographics lattices to be extended. However, this structure requires more space for storage and extra processing time.

We evaluate update performance for both structures by measuring index throughput versus

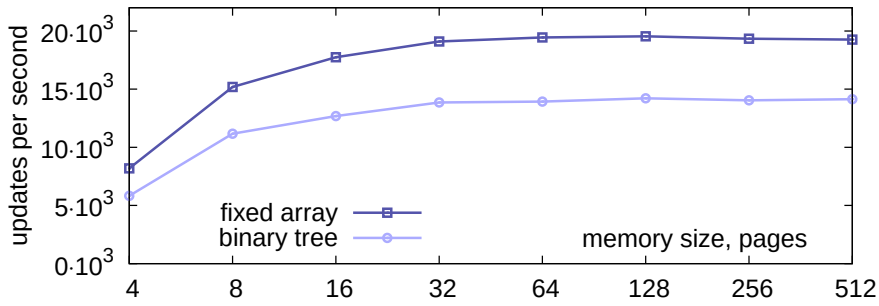


Figure 4.8: Performance of DTree versus memory cache size.

main memory cache size. Smaller cache sizes require disk pages to be flushed more often, while larger cache sizes allow to have a smaller number of sequential updates. With the cache size of one page, the system becomes persistent, i.e. all changes are immediately stored on disk. Larger memory cache sizes demonstrate higher throughput, which is ultimately bounded by disk writing speed, as the processing time is sufficiently smaller than the input-output time. Another advantage of a larger cache is that in-memory updates are extremely fast if a cache has enough space. Nevertheless, the update rate drops back to nominal once the cache is full, and until it is flushed to disk, there is a risk of loosing the data in the case of memory or system fault. The results of our evaluation are presented in Figure 4.8. We observed that disk writes occur every time the DTree is updated up until when the cache size reaches the maximum number of parent nodes in the tree. In our experiment, the updating rate stayed constant until 4 pages. After that, when the cache is further increased, we see a linear improvement in performance, which is then asymptotically reaches the maximum value, bounded by the disk write speed. Binary tree (dynamic) storage features smaller update rate compared to the fixed array storage, since its disk pages occupy twice more space and since a binary tree is dynamically constructed.

Finally, we measured the time needed to update the CTree and Cdb with information from new posts. The updates in Cdb were executed as batch updates, with logging turned off. In Figure 4.10, we report the average time to perform 1,000 updates as a function of the number of topics. Each update operation corresponds to the update of a time window of the finest granularity (and consequently, of all its ancestors as well), or the creation of a new such window (and the update of its ancestors).

The graph shows that there is a linear dependency between the update cost and the number of topics in the system. As we discussed earlier, the increased cost for CTree comes from accessing additional nodes for each time window, when the number of topics do not fit in a single node. Nevertheless, CTree still performs 4 times faster than Cdb.

Performance of top-k topic retrieval

Since the database solution stores information about all the topics in the same table and treats them uniformly, its performance can not be improved for the cases where some topics are more popular (receive more queries) than others. Therefore, the uniform distribution of topic ids used in our experiments favors the database approach. In contrast, CTree can arrange topics using different orderings (e.g., sorted by popularity or contradiction level), and do so independently for each time interval.

To have a notion on how significantly the performance of the CTree at answering top-k queries improves when topics are stored pre-arranged by the decreasing level of popularity, we performed an experiment on a range of "all-topics" queries with random parameters. In Figure 4.11, we plot the average execution times for *Query 2* using a varying limit on the number of returned contradictions. It is clearly visible that a sorted version of the CTree performs on average 6 times faster than the original one. However, this approach reduces performance for the ad-hoc topic access in the case when topics are arranged by contradictions rather than popularity.

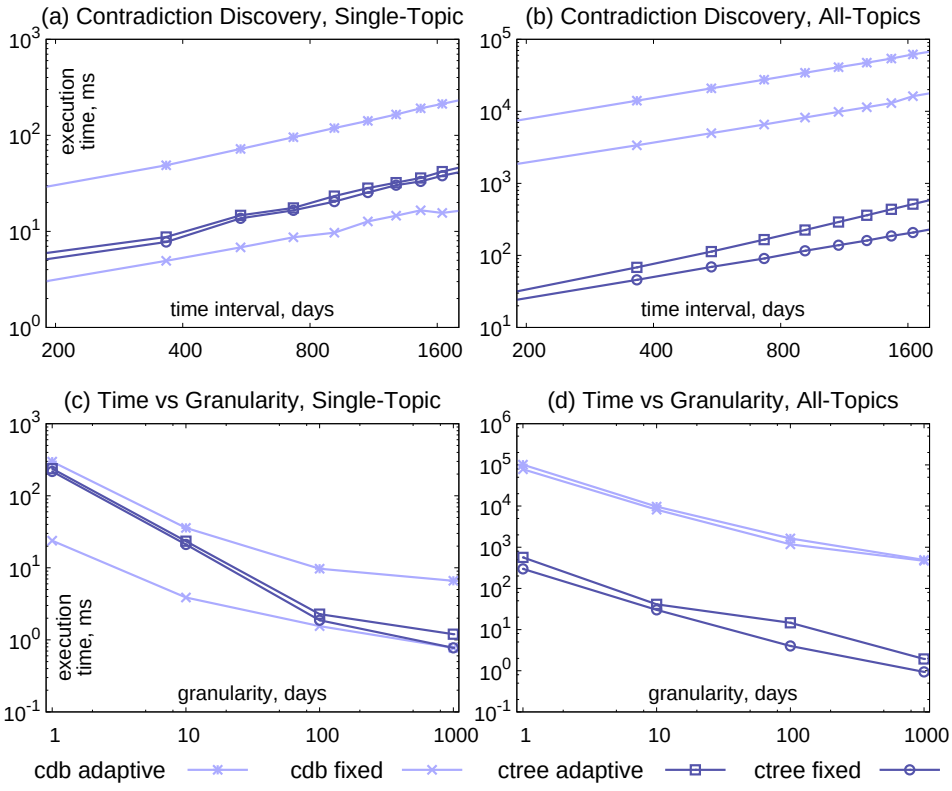


Figure 4.9: Single-topic vs all-topics queries scalability.

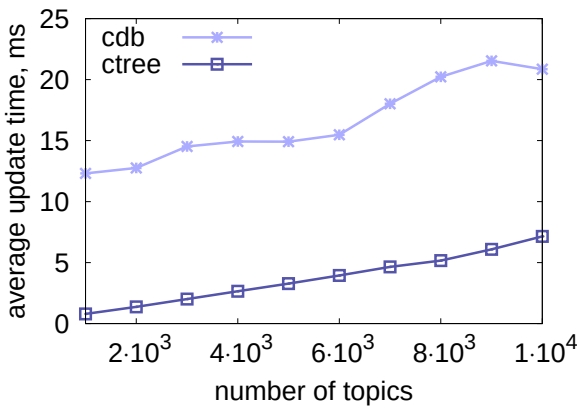


Figure 4.10: Update time vs number of topics.

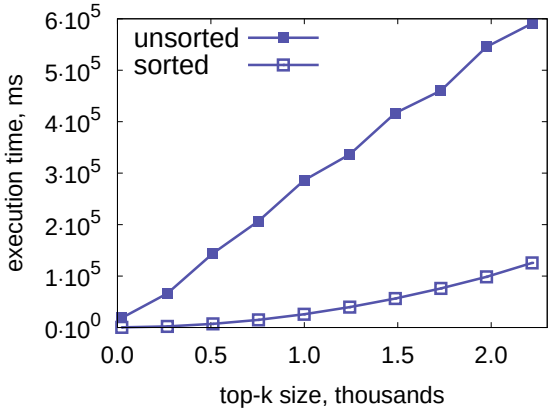


Figure 4.11: CTree top-k queries performance.

Chapter 5

Contradiction Analysis

Diversity is a natural feature of many areas which include a social aspect. As every person is unique, the same are their vision and way of thinking. When people represent information or describe something, it is natural for them to express their attitude or opinion, and it is natural for this opinion to be different not just between people, but also across time.

The analysis of diverse user opinions expressed on the Web is becoming increasingly relevant to a variety of applications. It allows us to track the evolution of discussions in the blogosphere, detect controversial topics, monitor shifts of opinions in relation to news events. The aggregation of diverse sentiments and analysis of contradictions therefore appears a very important application, which becomes effective since we are able to capture the diversity in sentiments on different topics with more precision and on a large scale. Though, the large-scale application dictates a need for an efficient way of sentiment aggregation with respect to the time dimension, that preserves a sufficient enough amount of information, allowing to capture contradictions and perform statistically accurate analysis of sentiment trends and opinion shifts for noisy sentiment observations.

In this chapter, we are focusing on the novel problem of finding sentiment-based contradictions at a large scale, based on data sources that are continuously updated. First, we define two types of contradictions, depending on the distributions of opposite sentiments over time. Second, we introduce a novel measure of contradiction based on the mean value and the variance of sentiments among different texts. Third, we propose a scalable and accurate method for identifying both types of contradictions at different time scales. We evaluate the performance of our method using synthetic and real-world datasets, as well as a user-study. The experiments demonstrate the effectiveness of the proposed method in capturing contradictions in a scalable and incrementally maintainable manner.

5.1 Introduction

The problem of contradictions, or sentiment diversity on some topic, has been studied in the context of different research areas, having a slightly varying notion in each case. For instance, in Information Retrieval opposite opinions and sentiments introduce noise to the fact-centric search and must be avoided [116]. In contrast, conflicting sentiments is one of the desired targets of mining of product reviews. Recently proposed methods can aggregate opinions expressed in customer reviews and extract a representative summary of sentiments on a feature-by-feature basis; or they can capture and aggregate sentiments on some topic among different texts [74].

We say that we have a contradiction when there are conflicting opinions for a specific topic, or sentiment diversity. This kind of contradiction can occur at one specific point of time or throughout a certain time period. Another interesting situation arises when the majority of the texts within some time interval exhibits a positive (negative) sentiment on a particular topic, and this time interval is followed by another one, where the majority of texts exhibits a negative (positive) sentiment on the same topic. A boundary between these time intervals, that contain a change of aggregated sentiment, can also be identified as contradictory, but with a special type of contradiction, which we call *Change of Sentiment*. Furthermore, a contradiction can occur within one text when an author presents different opinions on the same topic, or across texts when different authors express different opinions on the same topic.

In this part of our framework we define the concepts of aggregated opinion, opinion variance and contradiction with respect to the time dimension, and formulate relevant problems. We formally define the problem of contradiction detection, and further describe two variations of the problem, namely, *synchronous* and *asynchronous* contradictions. We present an approach, which solves the above problem for sentiment opinions by using a novel contradiction measure based on mean and variance of aggregated sentiment distribution. Moreover, we draw a mathematical connection between the proposed contradiction measure for sentiments and the more general opinion contradiction framework, so that it is possible to consider one as a restricted version of another, and better understand their properties.

Our method operates on sentence-level sentiments, which are represented in a continuous scale. This allows us to exploit different approaches for sentiment detection, which can be plugged in our framework. The use of mean and variance for contradiction detection allows our method to be fast and linearly scalable on the number of texts, which is an important feature for large-scale analysis. We further extend the performance of our framework by boosting its robustness against sentiment extraction noise and sentiment irregularity with the help of regression analysis and smart thresholding. We design contradiction detection methods to take a full advantage of our CTree storage, enabling them to scale to very large data collections. We experimentally evaluate the proposed approach using several synthetic and real datasets.

The results show its effectiveness and scalability. In addition, we perform a user-study that demonstrates the usefulness of the proposed contradiction measure.

The remainder of this chapter is structured as follows. In Section 5.2 we formally define the problem of contradiction detection. We present our approach for detecting contradictions in Section 5.3, and provide the experimental evaluation in Section 5.4. Finally, we conclude in Section 5.5.

5.2 Problem Definition

The problem we want to address in this part of our framework is the efficient detection of contradicting opinions and opinion shifts (on specific topics) from large-scale, noisy data sources that continuously produce new data. We first turn our attention to the forms of opinion contradictions, and formulate our problem in a more general context. Following that, we consider sentiment-based contradictions.

Usually, a particular source of information covers some general topic T (e.g., *health*, *politics*) and has a tendency to publish more texts about one topic than another. Yet, within a text, an author may discuss several topics. When using the term 'text' we refer either to the entire web document or its individual sentences. With the term 'sentence' we assume a particular piece of text expressing an opinion about a certain topic, which can not be split into smaller parts without breaking its meaning. For each of the topics discussed in some text, we wish to identify the opinion expressed towards it.[1]

Definition 6 (Opinion) *O represents a statement or a claim expressed by an author on topic T.*

The opinion can be either an objective statement, e.g. "car is *black*", or a subjective statement, e.g. "war is *bad*". In fact, there exist a wide range of different types of opinions. In this work, we are interested in contradicting ones, i.e. those that have no sense together. For example, "car is *black* and *white*", or "war is *good* and *bad*". The latter example represents a contradiction between opinions of the evaluative type, which we call sentiments and define as follows:

Definition 7 (Sentiment) *S with respect to a topic T is a multidimensional number that indicates the intensity of the evaluative opinion along basic emotional dimensions, such as Joy $\Leftrightarrow$ Sadness, Acceptance $\Leftrightarrow$ Disgust, Anticipation $\Leftrightarrow$ Surprise, and Fear $\Leftrightarrow$ Anger [153].*

However, extracting precise sentiments (in this multidimensional space) from text is still a major challenge for the sentiment analysis domain, and the majority of methods detect sentiments

[1] Although we assume that each sentence within a text may express a different sentiment for the same topic, enabling to capture contradictions on a sub-document level, for the purposes of large scale aggregation it is more convenient to operate at the level of documents.

projected into a single dimension, that is, polarity [130]. Following these methods, in this study we identify and record the polarity of sentiments, which we represent as real numbers in the range [-1, 1]. Negative and positive values represent negative and positive opinions respectively, while the absolute value of sentiment represents the strength of the opinion. In the following, we refer to sentiment polarity simply as *sentiment*.

For the proposes of building a general contradiction definition, we express the differences between opinions of any kind in a form of a distance function. Much for the same reasons, opinions are compared and evaluated only on a pairwise basis.

Definition 8 (Opinion Distance) *The opinion distance* $d(x,y) = \| x-y \| \in \mathbb{R}^+$ *is a positively-defined (multi-dimensional) function that satisfies to the conditions of semi-metric:*

$$\begin{cases} d(x,y) \geq 0 \\ d(x,y) = 0 \text{ if and only if } x = y \\ d(x,y) = d(y,x) \end{cases}$$

Apart from detecting opinions for individual texts, we also need to measure the aggregated opinion on some topic expressed in a collection of documents (that may span different authors, as well as time periods).

Definition 9 (Aggregate Opinion) *Aggregate Opinion* $\overline{O}$ *is an opinion with the closest accumulative distance to other opinions within a group:*

$$\overline{O} = \arg\min O \sum_{O_i \in \mathcal{D}} \| O - O_i \|^2$$

In the case of sentiment polarity, the aggregated opinion in the definition above can be instantiated using the *sentiment mean*, μ_S, which has the shortest distance to other sentiments.

Definition 10 (Opinion Variance) *Opinion Variance* σ_O^2 *is the average distance between opinions in* $\mathcal{D}$ *and Aggregate Opinion* $\overline{O}$*:*

$$\sigma_O^2 = \frac{1}{n} \sum_{O_i \in \mathcal{D}} \| O_i - \overline{O} \|^2$$

Now we are ready to provide a definition of contradictions, which quantifies the intuitions given in Definition 5. By comparing opinion values of different collections of texts contradictions are identified as follows:

Definition 11 (Opinion Contradiction) *A collection* $\mathcal{D}$ *of texts talking about topic* T*, is considered contradictory, if it can be partitioned into several groups of texts* $\mathcal{D}_i \subset \mathcal{D}$ *such that the distance between aggregate opinions of any two groups is at least* α *times greater than the maximum opinion variance:*

$$\min_{i \neq j} \| \overline{O}(\mathcal{D}_i) - \overline{O}(\mathcal{D}_j) \|^2 > \alpha \cdot \max_k \sigma_O^2(\mathcal{D}_k) \tag{5.1}$$

We define contradiction on a *pairwise* basis, where we evaluate the disagreement between two groups of documents in a collection. In this case, the similarity of information within each group serves as a reference point, providing a basic disagreement level. This definition can lead to different implementations, and each one of those will have a slightly different interpretation of the notion of contradiction. We argue that our definition captures the essence of contradictions, without trying to impose any of the specific interpretations. Nevertheless, in Section 5.3, we propose a specific method for computing contradictions, which incorporates many desirable properties.

This definition allows us to detect contradictions, but does not measure their strength. For this purpose, we define a contradiction measure C based on the number and sizes of contradicting groups: the largest contradiction occurs when there are many groups of equal sizes.

$$C = -\sum_{\mathcal{D}_i \in \mathcal{D}} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \cdot \log \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \tag{5.2}$$

When identifying contradictions in a document collection, it is important to also take into account the time in which these documents were published. Let $\mathcal{D}_1$ be a group of documents containing some information on topic T, and all documents in $\mathcal{D}_1$ were published within some time interval t_1. Assume that t_1 is followed by time interval t_2, and the documents published in t_2, $\mathcal{D}_2$, contain a conflicting piece of information on T. In this case, we have a special type of contradiction, called *Asynchronous Contradiction*, since $\mathcal{D}_1$ and $\mathcal{D}_2$ correspond to two different time intervals. Following the same line of thought, we say that we have a *Synchronous Contradiction* when both $\mathcal{D}_1$ and $\mathcal{D}_2$ correspond to a single time interval, t.

In order to detect contradicting opinions in collections of texts, we first need to group similar opinions and then calculate their relative differences. Whether this problem lies within the scope of clustering algorithms, it is known to be computationally challenging, so many of existing methods provide only approximate solutions.

Problem 1 (Contradiction Detection) *Partition a given collection of documents $\mathcal{D}$ into a minimal number of non-intersecting sub-groups $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$, such that Equation 5.1 still holds, and compute the level of contradiction.*

Depending on the kind of application, the above problem can be formulated for all topics in a collection, or just for a single one.

Problem 2 (Single-Topic Contradiction Detection) *For a given time interval τ, and topic T, identify the time regions, where a contradiction level is exceeding some threshold ρ.*

Problem 3 (All-Topics Contradiction Detection) *For a given time interval τ, identify topics T, which have the highest contradiction level, or the largest number of contradicting regions above some threshold.*

The time interval, τ, is user-defined, whereas the length of a basic window which aggregates the documents can vary depending on the type of contradictions the application is aiming at. As we will discuss later, the threshold, ρ, can either be user-defined, or automatically determined in an adaptive fashion based on the data under consideration.

The latter problem is interesting if we want to consider the popularity of certain web topics. Frequent contradictions may indicate ”hot” topics, which attract the interest of the community. In this work, we focus on the solution to the first problem, since the solution to the second one is its direct extension. Note that the approach we propose in this work is general, and can lead to solutions for several other variations of the above problem, such as detection of topics with periodically repeating contradictions, or with the most frequently alternating *Aggregated Sentiment*.

5.3 Contradiction Detection

Given the problems described before, we propose a four-step approach to contradiction analysis, as demonstrated in Figure 5.1. Steps one and two can be achieved based on existing methods, or adaptations of existing methods. We refer to these steps as ‘preprocessing’ and briefly describe in the following how we have adapted them. The focus of our work is then on the subsequent two steps, namely, the aggregation of extracted sentiments and their analysis in order to identify contradictions. Since we also consider the problem of the management of sentiment information in order to enable fast query answering for aggregated sentiment analytics, we need to address these problems with a family of functions which are based on incrementally updateable statistical values, that can be aggregated to meet the necessity of the hierarchical analysis.

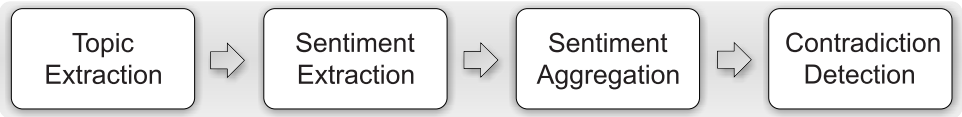


Figure 5.1: Schema of contradiction analysis.

5.3.1 Preprocessing

We determine topics and sentiments at sentence-level to be able to capture changes of sentiment within a single text. It may occur that regarding one topic different opinions are expressed in different sections of the same text. For example, the author of a weblog can collect arguments in favor and against some topic within one post, that may compensate each other during the averaging. By considering sentiment per sentence and relating it to the topic, we are able to detect these different opinions and preserve the contradiction within the post. Nevertheless, for the purposes of large scale analysis, we prefer to average sentiments over text’s sentences having

the same topic, to get one sentiment value for each topic in a text. This is done to equalize the participation of each document in the aggregated sentiment and to prevent the argumentation within some documents from affecting the contradiction level. As an additional benefit, this step reduces the amount of data to process.

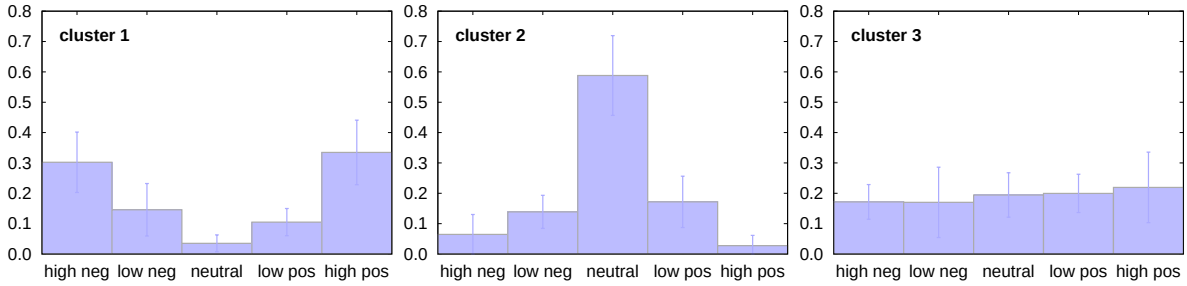
In order to identify topics per sentence and extract sentiments, we apply an existing text mining framework - LK [64]. We note that this framework operates more efficiently on syntactically correct sentences, which are not always observed in web texts. In such cases, we apply the Latent Dirichlet Allocation (LDA) algorithm [11], which was extended to work on the sentence level [33]. Badly structured sentences are considered as input documents for the LDA and assigned with several high-probable general topics. Then, for each sentence-topic pair we assign a continuous sentiment value in the range [-1;1] that indicates a polarity of the opinion expressed regarding the topic. For the sentiment assignment step, we use the LK tool for fine-grained opinion analysis. This tool achieves good results for opinion expressions detection and opinion holder extraction by applying a re-ranking classifier to the output of a conventional syntactic parser. Another feature of this tool is unsupervised domain-independent sentiment assignment, which is rather useful since we need to explore and process opinions for a variety of topics coming from different domains. Nevertheless, this tool can be replaced by any other suitable one that calculates continuous sentiment values at a sentence level.

So far we have described techniques for processing web documents to extract sentiments on various topics, and subsequently to aggregate this information for a document in order to analyze time series of sentiment on a large scale and over different aggregations. Based on the analysis described so far, we can now describe our approach for contradiction detection with respect to different topics. In the following paragraphs, we first propose a novel contradiction measure, and then describe two simple approaches aiming at detecting contradictive periods in time.

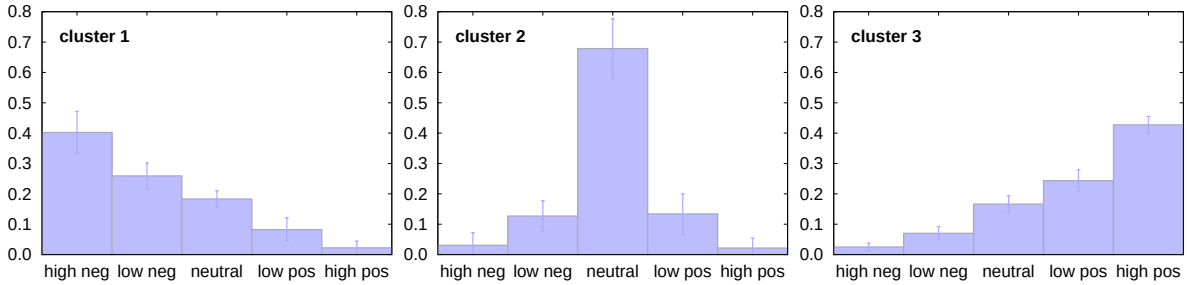
5.3.2 Contradictory Distributions

In order to understand how different sentiment distributions affect the perception of contradiction, we performed a short user study using three real datasets from diverse domains: drug ratings, YouTube, and Slashdot (details in Section 5.4.1). We asked users to select contradicting groups of texts contained in time intervals of pre-determined length (typically, ten days). Texts in these datasets were already annotated with sentiments, allowing us to reliably extract and aggregate sentiment distributions from various time intervals, selected by users, and on which the majority of users agreed. Additionally, we included distributions for such intervals, that were not marked as contradictory by any user. Overall, we selected 128 distributions, half of which were marked as contradicting and another half as non-contradicting.

In order to visualize and study sentiment distributions they were represented in the form



(a) sentiment distributions in contradicting texts



(b) sentiment distributions in non-contradicting texts

Figure 5.2: Typical sentiment distributions.

of histograms. In each time interval, sentiments from texts were aggregated into a histogram composed of five bins: “high neg”, “low neg”, “neutral”, “low pos”, “high pos”. Furthermore, histograms of both types (taken from contradicting or non-contradicting time intervals) were aggregated into the three clusters using *k-means* clustering with euclidean distance metric. In Figure 5.2, we report histograms that visualize the centers of these clusters, as well as the standard deviations of their values.

In this figure, we observe that contradicting groups of texts have nearly symmetrically balanced positive and negative sentiments, while non-contradicting ones have either positive or negative deviations in sentiment. The only exception to the above statement is cluster 2 in Figure 5.2(b), which resembles cluster 2 in Figure 5.2(a), but contains 10% more neutral sentiments and, therefore, may be distinguished from the latter by a smaller variance.

From what we have discussed above it is evident that it is not possible to detect contradictions by simply looking at the average sentiment. Appropriately, we need a model that leads to an efficient solution, and that can effectively account for the different sentiment distributions shown in Figure 5.2. In the following, we introduce such kind of model that is based on statistical data.

5.3.3 Modeling Contradictions

In order to be able to identify contradicting opinions we need to define a measure of contradiction. Assume that we want to look for contradictions in a shifting time window w[2]. For a particular topic T, the set of documents $\mathcal{D}$, which we use for calculation, will be restricted to those, that were posted within the window w. We denote this set as $\mathcal{D}(w)$, and n as its cardinality, $n = |\mathcal{D}(w)|$. We can use the following measures of contradiction:

The *sentiment mean*, μ_S, is calculated as $\mu_S = \frac{1}{n}\sum_{i=1}^{n} S_i$. It can be easily proven that this value has the lowest sum of distances to sentiments in the collection, that is, it conforms to our definition of Aggregated Sentiment. It can be seen, that a value of μ_S close to zero implies a high level of contradiction because of positive and negative sentiments compensate each other. A problem with the above way of calculating the aggregated sentiment arises when there exists a large number of documents with very low sentiment values (neutral documents). In this case, the value of μ_S will be drawn close to zero, without necessarily reflecting the true situation of the contradiction. Therefore, we suggest to additionally consider the variance of the sentiments along with their mean value.

The *sentiment variance*, σ_S^2, is defined as the average of squared distances between sentiments and their mean: $\sigma_S^2 = \frac{1}{n}\sum_{i=1}^{n}(S_i - \mu_S)^2$. According to this definition, when there is a large uncertainty about the aggregated sentiment of a collection of documents on a particular topic, the topic sentiment variance is large as well.

The sentiment mean and variance can be expressed using first- and second-order moments of sentiment $M_1 = \sum_{i=1}^{n} S_i$ and $M_2 = \sum_{i=1}^{n}(S_i)^2$, giving us the following formulas for sentiment statistics:

$$\mu_S = M_1/n; \quad \sigma_S^2 = M_2/n - (\mu_S)^2; \tag{5.3}$$

We demonstrate the effect of outlined measures in Figure 5.3, featuring two example sentiment distributions. Distribution A with μ_S close to zero and a high variance indicates a very contradictive topic. Distribution B shows a far less contradictive topic with sentiment mean μ_S in the positive range and low variance. For example, a group of documents with μ_S close to zero and a high variance (distribution A on the Figure 5.3) will be very contradictive, and another group with sentiment μ_S shifted to negative or positive with low variance is likely to be far less contradictive (distribution B on the Figure 5.3). When assuming a large number of neutral sentiments in the collection, we have two opposite trends: the average sentiment moves towards zero and sentiment variance decreases. If these trends will compensate each other, the neutral documents would not affect the contradiction value much.

[2]Without the loss of generality, in this work we consider windows of days, weeks, months, and years.

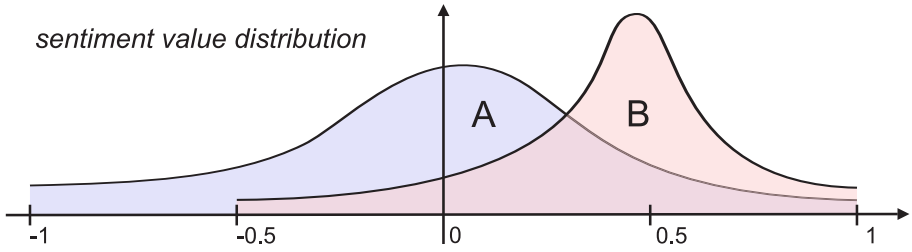


Figure 5.3: Possible sentiment distributions.

Evidently, we need to combine mean and variance of sentiments (expressed in the same units) in a single formula for computing the contradiction value C:

$$C = \sigma_S^2/(\mu_S)^2 \tag{5.4}$$

This formula captures the intuition that contradiction values should be higher for topics whose sentiment value is close to zero, and sentiment variance is large. Moreover, it satisfies the criteria for opinion contradictions, as demonstrated below.

Let's take another look at the two collections of sentiments, A and B, demonstrated in Figure 5.3, and presume that they have n_a and n_b sentiments in each, distributed with parameters (μ_a, σ_a) and (μ_b, σ_b). We can calculate the mean value μ_S and the variance σ_S^2 of their aggregated sentiment distribution, which has $n = n_a + n_b$, $M_1 = M_1^a + M_1^b$ and $M_2 = M_2^a + M_2^b$.

$$\mu_S = \frac{M_1^a + M_1^b}{n} = \frac{n_a\mu_a + n_b\mu_b}{n_a + n_b} \tag{5.5}$$

$$\sigma_S^2 = \frac{M_2^a + M_2^b}{n} - \mu_S^2 = \frac{n_a\sigma_a^2 + n_b\sigma_b^2}{n_a + n_b} + \frac{n_a n_b(\mu_a - \mu_b)^2}{(n_a + n_b)^2} \tag{5.6}$$

Using the formulas of the aggregated values μ_S and σ_S^2 in our measure of contradiction, we obtain the following expression:

$$C = \frac{\sigma_S^2}{\mu_S^2} = \frac{(n_a + n_b)(n_a\sigma_a^2 + n_b\sigma_b^2) + n_a n_b(\mu_a - \mu_b)^2}{n_a\mu_a + n_b\mu_b} \tag{5.7}$$

If our collections satisfy Definition 11, we can remove from the denominator the component depending on the variances, since they are smaller than the distance between mean values:

$$C > \frac{n_a n_b(\mu_a - \mu_b)^2}{n_a\mu_a + n_b\mu_b} \tag{5.8}$$

Now it can be clearly seen that larger separation between sentiment distributions results in higher contradiction value. Taking into account the limited range of sentiment values, this

distance is the largest when sentiment means are of the opposite polarities. In this case, the two sentiment distributions compensate each other and the denominator becomes very small, obtaining $C >> 0$.

Nevertheless, the contradiction values generated by this formula are unbounded (i.e., they can grow arbitrarily high as μ_S approaches zero), and does not account for the number of documents n, that is for the significance of sentiment statistics. For instance, in the extreme case when $\mathcal{D}(w)$ contains only two documents with opposite values, C will become infinitely high, and thus incomparable to the contradiction value of any other set of documents with higher cardinality. While the first problem (of the infinite contradiction scale) can be addressed with the help of a regularizing constant added to the denominator, the second problem (of statistical significance) is important for small-scale applications or for streams of sentiments with the irregular flow. We propose to cope with this problem by filtering on the significance of statistics involved in the calculation of C using the weight function W, defined in Section **??**. Using W we can effectively limit C when there is a small number of documents, and also weigh C more when the number of documents is large, emphasizing the most debated regions in time. What W achieves is essentially a normalization of the contradiction values across sets of documents of different sizes, allowing them to be meaningfully compared to each other.

Incorporating to the contradiction formula the observations made above, we propose the following final formula for computing contradiction values:

$$C = \frac{\vartheta \cdot \sigma_S^2}{\vartheta + (\mu_S)^2} W \tag{5.9}$$

In the denominator, we add a small value, $\vartheta \neq 0$, which allows to limit the level of contradiction when $(\mu_S)^2$ is close to zero. The nominator is multiplied by ϑ to ensure that contradiction values fall within the interval $[0;1]$. Figure 5.4(c) shows how a contradiction value depends on ϑ in the denominator. Smaller ϑ values emphasize contradiction points with μ_S close to zero, for example changes of opinion. Larger ϑ values mask this difference, making levels of contradictions more equal. In this study, we used a value of $\vartheta = 5 \cdot 10^{-4}$, which was effective for its purpose, exhibiting a stable behavior across datasets, without distorting the final results.

An important observation is that the Formula 5.9 that calculates the contradiction values is based on the mean and variance of the sentiment, which can be computed from the first- and second-order moments of sentiments, as shown in Formula 5.3. Based on this representation, we can rewrite Formula 5.9 using the sums M_1 and M_2, as follows:

$$C = \frac{\vartheta(nM_2 - M_1^2)}{\vartheta n^2 + M_1^2} W \tag{5.10}$$

The above form of the contradiction formula gives us additional flexibility, since we can

now compute the contradiction of a large time window by composing the corresponding values from the smaller windows contained in the large one. We can therefore build data structures that take advantage of this property.

Figure 5.4 shows the operation of the proposed contradiction function. To better illustrate this, we use one of the time series from our synthetically generated dataset (described in Section 5.4.1). The graph at the top (Figure 5.4(a)) shows generated sentiments. The bold line in this graph depicts the custom trend, showing an initial positive sentiment that later changes to negative (at time instance t_1), which represents an asynchronous contradiction (change of sentiment) that manifests itself across the entire dataset. There is also a point around time instance t_2, where the sentiments are divided between positive and negative, a situation representing a synchronous contradiction. As can be seen in Figure 5.4(b), a smoothed trend of μ_S (using regression smoothing) captures the aggregated sentiment better than the simple average, effectively reducing noisy fluctuations. The graph in Figure 5.4(c) shows the contradiction value obtained using smoothed mean and variance of sentiments. In this case, C correctly identifies the two contradictions at points t_1 and t_2, where the values of C are the largest. In this case, using simple aggregated values of sentiments μ_S straight away can result in C reporting noisy fluctuations of sentiments as contradictory.

Subjective sentences take a considerably small part in the text when compared to objective statements. So neutral sentiments usually shift the aggregate sentiment towards zero, masking contradictions. Our contradiction formula is designed to compensate such effects by exploiting the sentiment variance. We demonstrate such behavior on another synthetic dataset shown in Figure 5.5. The bottom graph shows that the proposed formula can successfully identify the main contradicting regions, both with or without neutral sentiments. Nevertheless, in their perception of contradiction, people usually account for the relative amount of neutral statements. Hence, they do not consider as contradictory regions containing mostly neutral sentiments (as we observed in Section 5.3.2). This should be taken into account if subjectivity filtering is applied upon sentiment extraction, removing neutral sentiments from the distribution. In such cases, it is possible to tune the sensitivity of our contradiction measure by setting a higher value to the parameter ϑ.

5.3.4 Detecting Contradictions

While weighting on the number of documents addresses the problem of significance of contradiction values in cases when user activity varies over time, there exists another source of sentiment irregularity, which is the result of sampling variation. This type of irregularity can be explained by considering that population samples that contribute to aggregated sentiments are quite different across adjacent time intervals. Indeed, people tend to publish at a particular rate, and the likelihood that they will re-state their sentiment shortly after the first publication is low

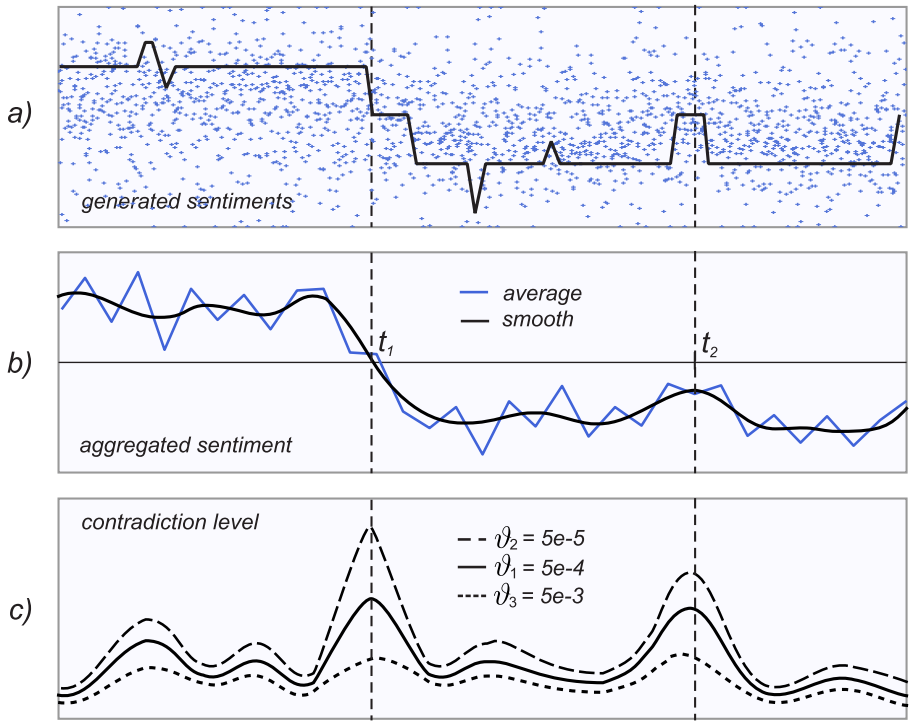


Figure 5.4: Sentiment data with artificial contradictions.

for small aggregation windows. On one hand, increasing the window size at the same granularity level can help reducing such noise, but at the same time it will decrease the resolution of our analysis, allowing to identify only long-lasting contradictions. On the other hand, applying a sliding window of a larger size on a smaller granularity requires substantially more resources for storage and computation. To cope with this problem, we use local regression smoothing as described in Section 4.2, which achieves a more accurate sentiment trend, by considering sentiment variance and number of aggregated sentiments.

In the case of synchronous contradictions, when the community at every particular interval in time has different opinions about the same topic, contradictions can be determined easily with any suitable time window. However, sometimes the community has a solid opinion in one time period, and later changes it, so in another time period it has the opposite opinion, resulting in an asynchronous contradiction. This type of contradiction can only be discovered using a time window large enough to gather posts from the two different periods. Moreover, if for some shifts of opinion there exists a gap in time between positive and negative posts, the detection becomes highly dependent on a time window and on the order in which posts were published. By using a small time window we will likely to get only a small peak of contradiction at the moment, when community has changed it's opinion, because the transition between opposite opinions is slow to result in any significant difference of opinions at any particular time interval. Thus, the

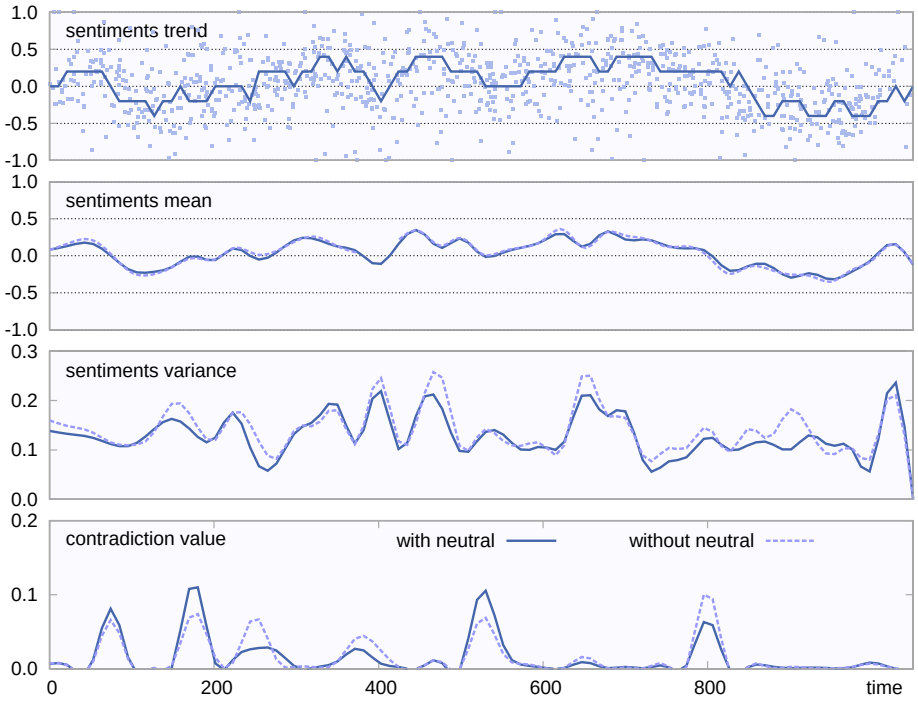


Figure 5.5: The effect of neutral sentiments on Formula 5.10.

hierarchical refinement of time intervals from larger ones to smaller is particularly important for the discovery of asynchronous contradictions.

When trying to detect contradictions, we would like to identify those that have a contradiction value above some threshold. The intuition is that these contradictions are going to be more interesting than the rest in the same time interval. An obvious solution in this case is to define some fixed threshold, ρ, and only report the contradictions above this threshold. We refer to this solution as *fixed threshold*. However, by adopting the above solution, we cannot normalize the threshold to better fit the nature of the data within each time window (that may vary over time and across topics). In order to address this problem, we propose an *adaptive threshold* technique, which computes a different threshold for each topic and time window as follows. The adaptive threshold ρ_w for a topic T in time window w is based on the contradiction value C_{w_p} that has been calculated for T in the parent time window of w, w_p, and is defined as $\rho_w = p \cdot C_{w_p}$. In our experience with real datasets, p values between $0.5 - 0.7$ work well. In this work, we use $p = 0.6$.

Adaptive threshold helps to detect interesting contradictions that occur in different time granularities and across topics, even if these contradictions do not have the largest values overall. This is particularly important when a single, fixed threshold value cannot detect all contradictions across time, or when the user is unsure about which threshold to choose. Note that

we cannot achieve the same result by using *top-k* queries (though, they can be complementary to our approach). The reason is that the adaptive threshold is changing as we navigate the timeline, and it provides even discrimination of peaks of contradiction both in highly- and lowly-contradicting regions. Moreover, it does not impose a strict limit on the number of contradictions in the result, and can thus report the entire set of interesting contradictions within some time interval.

We are now ready to present the algorithm we use to retrieve topic contradiction values using time windows of a given granularity. Listing 2 outlines the algorithm that uses the adaptive threshold. The algorithm needs a single pass over the collection of pages of the specified granularity, l, that fall inside the time interval, τ of the query. In line 6, we check if a contradiction value (for a specific topic and time window) is above the adaptive threshold. Note that contradiction values, C^T are computed from the information stored in the node using Formula 5.10. The type of contradiction is identified in lines 4-6, by comparing signs of sentiments for adjacent nodes. In our implementation, we additionally do not visit children nodes whose parents are not contradictory (we omit this detail from the algorithm for ease of presentation).

Algorithm 2: CTree Access

Input : Topic T, Time interval τ, Granularity l
Output: List of contradictions
$\mathcal{C} = \{(time\ window, contradiction\ value, type)\}$

```
1  forall the nodes r ∈ τ, r.granularity = l − 1 do
2      forall the nodes r_i ∈ r.childNodes do
3          if r_i ∈ τ and r_i.C^T > p × r.C^T then
4              if r_{i−1}.S^T × r_i.S^T ≤ 0 then
5              type = "asynchronous contradiction";
6              else type = "synchronous contradiction";
7              C = C ⋃ (r_i, r_i.C^T, type);
8          end
9      end
10 end
11 Arrange C by topic contradiction count or level;
```

5.4 Experimental Evaluation

In this section, we report the results of our experimental evaluation on synthetic and real datasets. The objectives of the experiments we conducted were to: analyze the quality of the approach; study its usefulness from a user perspective; study the scalability of our solution.

5.4.1 Datasets

Synthetic Dataset

Specifically for the evaluation of accuracy and performance of our method, we generated a synthetic dataset containing time series of sentiments with artificial opinion shifts, contradictions and a controlled amount of noise. To create this dataset we generated and then inserted in the CTree a large volume of sentiments with time stamps following the Poisson distribution with the parameter λ ranging from 1 to 10 sentiments per day, and with values following the normal distribution. Moreover, a particular fraction of sentiments followed a planted trend with dispersion 0.125, while the rest, controlled by the noise parameter, were distributed randomly with dispersion 0.5 and median 0.0. We have chosen these distributions because they are simple , and still resemble the real data. Noise parameter varied from 0.0 to 0.4 with a step of 0.1. [3] Overall, we generated 1000 topic time series, and stored them in the CTree, making independent copies for each of the above noise parameters.

Real Dataset

We study the usefulness of our algorithms on a data set of drug reviews collected from the DrugRatingz website[4], a data set of comments to YouTube videos from L3S and a dataset with comments on postings from Slashdot, provided for the CAW2 workshop[5].

The first dataset contains 2701 positive, 352 neutral and 1616 negative reviews for 477 drugs. These reviews are provided by persons that took a specific drug. They describe their personal experience with the drug, including side-effects that occurred.

The second dataset contains approximately 6 million comments to YouTube videos, with an average of 500 comments per video. These comments feature a lot of argument on various topics, which are often different to topics mentioned in videos.

Our third dataset, Slashdot, is from a popular website for people interested in reading and discussing about technology and its ramifications. It publishes short story posts which often incite many readers to comment on them and provoke discussions that may trail for hours or even days. It contains about 140,000 comments under 496 articles.

We conducted an evaluation of the precision of our approach on the user-annotated dataset of sentiment distributions from DrugRatingz, pictured in Section 5.3.2. In this particular sample of our dataset, we limited the number of false positive annotations by including as non-contradictory only the intervals marked as such by *all* users. Additionally, we balanced the number of positive and negative samples in this dataset to include 64 of each, so that the baseline precision became 50%.

[3] We note, that even at 0.0 noise setting a constant amount of noise is present in the time series itself.

[4] http://drugratingz.com

[5] http://caw2.barcelonamedia.org/

Method	CTree	SVM-hist	SVM	LR	LR-hist	Baseline
Accuracy	**82.0**	79.7	78.9	68.8	66.4	50.0
Precision	**93.6**	93.2	91.1	72.2	66.2	50.0
Recall	**68.8**	64.1	64.1	60.9	67.2	100.0
F-Measure	**79.3**	75.9	75.2	66.1	66.7	66.7

Table 5.1: Performance evaluation of synchronous contradiction detection.

5.4.2 Accuracy

Synchronous Contradiction Detection

We evaluate the accuracy of our approach for synchronous contradiction detection on the annotated dataset of contradicting sentiment distributions described in Section 5.3.2. More specifically, we compare the accuracy of our method to several supervised machine learning classifiers, available in Weka data mining tool. We used the same dataset both for training and testing, and the classifiers used feature vectors either based on mean and variance (same as our approach) or on histograms (reported as *hist*).

As the main alternative to our method, we chose an SVM classifier (nu-SVC type using radial kernel), since its performance characteristic is well known. In addition to SVM, we used the Logistic Regression (LR) classifier. Parameters of machine learning methods and the CTree threshold were optimized for the best overall accuracy (for both positive and negative classes). We report average statistics for 10-fold cross-validation, where 90% of data were used for training and 10% for testing on every iteration.

The results of our evaluation are shown in Table 5.1, where we report the overall accuracy (the number of instances correctly classified as contradictory or non-contradictory) and contradiction detection precision and recall (according to instances correctly classified as contradictory). Reported F-Measure is the harmonic mean of precision and recall, indicating the overall contradiction detection performance.

The best results were achieved using CTree, which was 3% more accurate than SVM and close followed only by histogram-extended version, SVM-hist. Logistic Regression method has demonstrated significantly worse results and was not able to benefit from using the histogram data, much likely due to the impossibility of separating classes in the linear space. We should note, that in this experiment SVM methods were rather good at classifying contradictions mainly because of the cross-validation and exhaustiveness of the evaluation dataset. While the first circumstance alone required training on the 90% of whole data, in combination with the second one it made most of the testing samples similar to those used for training. This makes the reported precision values for these methods reading as an optimistic estimate, rather than the actual performance. On the other hand, the optimal value of the CTree threshold used in these experiments was most often equal to 0.5 of the average contradiction level across all

tested samples, indicating that our approach is very effective even with the default setting of the adaptive threshold.

We further note that our model uses only statistical moments as input data, which are less descriptive than histograms. Furthermore, our method in this case did not not apply the significance-based weighting, since we used annotated data based on text collections of the same size. Because SVM utilizes normalized values, it cannot automatically handle situations when statistical values are not significant due to a small number of sentiments. Even when the number of sentiments is added as an additional feature, pruning on this feature can be either uncontrollably biased by other features, or very strict, depending on training data.

Even though SVM methods have demonstrated a comparable performance, they fail at delivering several other important properties that are relevant for our problem and supported by our method:

- SVM provides no measure for the level of contradiction and cannot filter or rank the result set appropriately. Neither it can flexibly change between high precision and high recall, as it is possible using a simple threshold in our method.
- SVM cannot automatically handle situations when statistical values are not significant due to a small number of sentiments, since it relies on normalized values.
- SVM methods are not able to adapt to different kinds of sentiment biases in real datasets without re-training. In contrast, we can control sensitivity of our method using ϑ parameter, and additionally it is possible to compensate biased sentiments by appropriately adjusting μ_S, without modifying the actual values stored in the CTree.
- Finally, sentiment histograms occupy more space than the aggregated statistics used in our method, and thus require more processing time.

Asynchronous Contradiction Detection

In order to evaluate the accuracy of asynchronous contradictions, we manually identified changes of opinion in 10 time series randomly picked from our synthetic dataset. An example can be seen in Figure 5.4, where time point t_1 corresponds to an asynchronous contradiction. To simulate a real-world application, we labeled only the most prominent asynchronous contradictions, which are important to analysts. We did not consider as asynchronous contradictions changes of opinion, where the sentiment time series shortly crosses the zero line and then goes back. (We purposely did not annotate synchronous contradictions, as their perception is subjective and largely depends on annotator's experience.)

We evaluate the accuracy of our method with and without smoothing by measuring precision and recall of extracting contradictions for varying noise levels (ranging from 0.0 to 0.4).

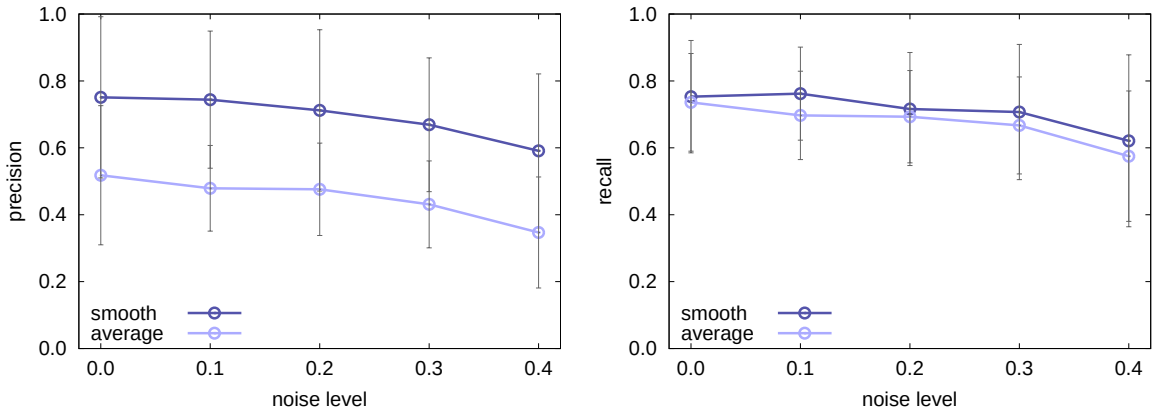


Figure 5.6: Accuracy of asynchronous contradiction detection with or without regression smoothing.

For these experiments we used a constant 0.05 contradiction threshold, and 0.5 coefficient for time series smoothing. Precision is computed as the percentage of the extracted contradiction intervals, which match to the ones manually annotated in the dataset. Recall is computed as the percentage of the annotated contradiction intervals, which were actually extracted.

The graph in Figure 5.6 shows the accuracy of extracting opinion shifts with and without regression smoothing applied. We observe that both methods correctly identify a large fraction of the contradictions at all noise settings. The recall is varying from above of 75% to about 60%, and ranging around 75% (for the smoothing version of our method) for low to mid noise levels, meaning that the method is applicable to and useful for information retrieval purposes. The fact that recall values never reach 100% reveals that some of the opinion changes can not be detected at the granularity of 10 days. Precision values are significantly better for the smoothing version of our method (75% versus 50%). Applying adaptive threshold instead of a constant threshold in this case should not yield a dramatic improvement of precision of asynchronous contradictions, since their detection mainly depends on variance (sentiment mean being zero at a change point), which remains almost constant at a 10-day aggregation granularity.

5.4.3 Correctness

We now apply the introduced contradiction analysis approach to real datasets, aiming to assess the correctness of identified contradictions by navigating to the relevant collections of text and extracting opposite points of view. Figure 5.7 depicts highly irregular subset of sentiment values for the topic “internet government control” taken from the Slashdot dataset, for the time interval September 2005 to September 2006. This dataset reflects a typical situation in Opinion Aggregation, when the irregular flow of values results in a time series with variable significance of aggregates or even missing values. In the upper graph, we plot the average sentiment time series, and the contradictions that were extracted based on this series, when there is no

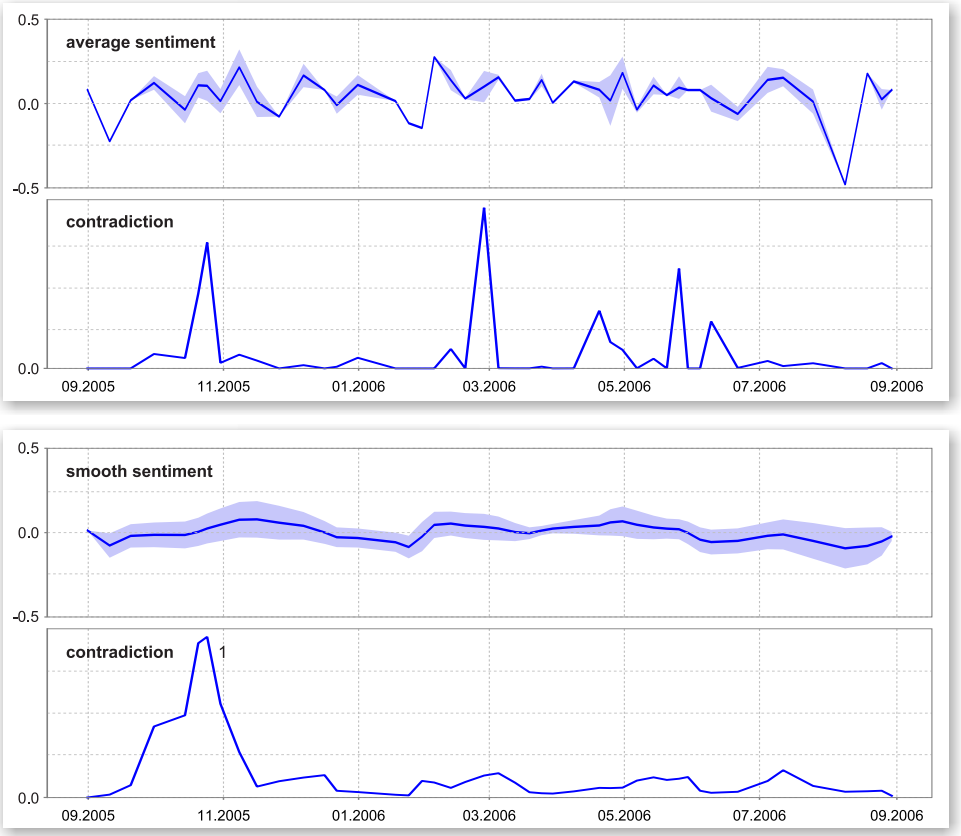


Figure 5.7: Average and smooth sentiments for “Internet government control”.

smoothing applied to the sentiment series. In the lower graph, we plot the same average sentiment time series, where we have additionally applied our regression smoothing step, and the corresponding detected contradictions. In both graphs we plot sentiment and variance and the corresponding contradiction values , calculated using a time window of ten days. We observe that the trend in the upper graph, computed by simply averaging sentiments, results in contradiction values that do not accurately reflect the evolution of sentiments. For example, the second and the subsequent peaks of contradiction caused by spontaneous fluctuations of aggregated sentiment towards zero. On the other hand, the regression-smoothed trend in the lower graph demonstrates a more stable behavior. Note that contradiction values are high for the time windows where topic sentiment is around zero and variance is high, which translates to a set of posts with highly diverse sentiments. These situations are not easy to identify with a quick visual inspection of the raw sentiments.

Analysis of the regression time series shows that there is one major contradiction (marked 1 in the bottom graph of Figure 5.7). This contradiction discusses the pros and cons of a law that would give the government more power in controlling the internet traffic, especially personal correspondence. By taking a closer look at the corresponding weblog posts (reported

in Table 5.2), we find out that the main discussion is about restricted internet access and its advantages, while other contradictions contain a general discussion on the possibility of organizing the content by several top-level domains and restricting access to them, and of a possible transfer of jurisdiction and control over top-level domains to United Nations.

In Table 5.2, we also report additional examples of contradictions identified by our analysis. For the topic "iraq war", the related posts discuss pros and cons of the US strategy on this issue. A more detailed study of the corresponding posts shows that the discussion is about the deployment of US Army troops in the Iraq. On the one hand, people discuss how different media influence people's opinion on the Iraq war and the US strategy. On the other hand, people discuss about the US Iraq strategy itself. The extracted posts correspond to an asynchronous contradiction that our algorithm identified in the time period of May 2006. This particular change of sentiment was from positive to negative. Interestingly, it coincides with a surge of bomb attacks in Iraq, which claimed many US lives. The next example of contradictions comes from the WebMD dataset, where posts discuss treatment of AD/HD[6]. One group of posts speaks in favor of a specific brand name drug, which is an antidepressant, while others indicate the disadvantages of this drug, and suggest a different drug. Evidently, these are all very relevant discussions that express different points of view on the same topic, and having an automated way of identifying them can be very useful.

5.4.4 Usefulness

In the following paragraphs we describe a user study which we conducted in order to evaluate the effectiveness and usefulness of our approach for the task of contradiction discovery.

In our usefulness evaluation, we used four datasets corresponding to opinionative posts for four topics extracted from three diverse real datasets (refer to Table 5.3). For each topic, we selected a varying number of posts, spanning in time from one to almost three years. The shortest list contained 60 posts, and the largest about 480. Moreover, the quality of posts for topics also differed a lot. The drug review datasets contained primarily brief and concise opinions about drugs; Slashdot topics featured large and detailed comments, with an average size of several paragraphs; YouTube comments were, on the contrary, short and often off-topic.

The group of users consisted of eight persons (PhD students of various disciplines at the University of Trento), and the experiment was conducted as follows. Participants were asked to detect groups of contradicting posts for each of the topics in the above datasets (and label the positive and negative posts). We provided them with a web application that featured two approaches to help them identify time-intervals with potentially contradicting posts (see Figure 5.8) and digest their content: The first approach (marked as "stage 1" in the figure), based

[6]Attention-Deficit Hyperactivity Disorder is a commonly diagnosed psychiatric disorder in children.

topic "Internet government control", Slashdot (contradiction 1 in Fig. 5.7)
PRO: It would be helpful for restricting the flow of information, which is a double edged sword.
PRO: I suppose we better wrap a firewall around our country and not let those damn foreigners access to our internet.
PRO: "A slew of Chinese web portals have pledged to self-police even more, after signing on to a Beijing plan to 'clean up the internet'." if it's not a government doing it, it's not censorship.
CONS: And what exactly does a neutral Internet do? It takes away the right of anyone who lays down the wires or installs the access points to control what goes through their network. don't complain about taking rights away when you advocate to take rights away.
CONS: While it sounds like a decent idea, I'm really all for the whole uncensored and unregulated internet.
CONS: Sure, they can ruin Internet inside USA, but the rest of the world couldn't care less.
CONS: We don't need the FCC regulating the Internet. Not for "neutrality" or any other excuse someone can think of.
topic "Iraq war", Slashdot
PRO : You are fortunate to live in a country unencumbered by an ongoing threat of terrorism and I respect your governments decision to oppose the U.S. attack in Iraq.
CON : Unfortunately, that happened to many Americans during the run-up to the ongoing war in Iraq. Most Americans didn't investigate the claims made by politicians and the media, and thus were ignorant to the fact that they were being seriously mislead.
topic "adhd child", WebMD
PRO : I have seen antidepressants make a huge positive difference lifting a child's mood and improving the quality of his/her life.
CON : Stimulants treat symptoms of ADHD in a greater percentage of people than [Drug], and often treat inattention and destructibility more robustly than [Drug]. [Drug] isn't safer than a stimulant, and if effective a stimulant alone would be a far better choice.

Table 5.2: Examples of contradicting posts.

on the visualization method proposed by Chen et al. [19], displays to users the intensity over time of the positive and negative sentiments expressed in the posts (Figure 5.8(a)). The second approach (marked as "stage 2" in the figure) is based on the method proposed in this study, and displays to users a graph that marks the time points at which contradictions were automatically detected (Figure 5.8(b)). Using our tool, the users could see the time intervals that our tool had identified as contradictory, and could therefore, focus their exploration in these regions. Figure 5.8(d) shows some posts in a time-interval, which have been marked with positive (green) and negative (red) sentiments. These sentiments values are also illustrated in the overall time-line, depicted in Figure 5.8(c). In order not to favor any of the two approaches, in our experiments we alternated the approach required to be completed first.

For both approaches, we measured the average time, T_1 and T_2, and the average number of time-intervals examined by the users during the search, N_1 and N_2, needed to identify a single contradiction. Additionally, we asked users to rate the overall difficulty, D_1 and D_2, of completing the task when using each one of the two approaches, according to the following scale: 1- very difficult; 2 - somewhat difficult; 3 - normal; 4 - somewhat easy; 5 - very easy.

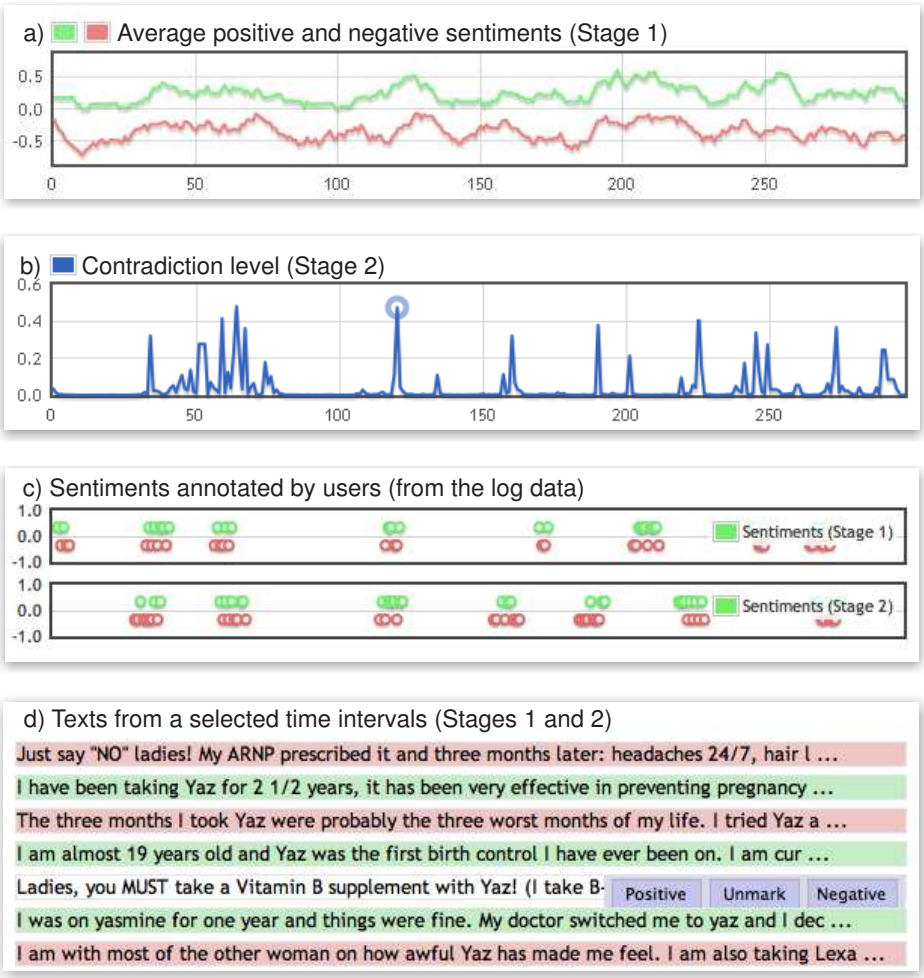


Figure 5.8: Annotation page for the dataset “Yaz” demonstrating opposite opinions.

The aggregate results (averaged over all the users) of our evaluation are reported in Table 5.3. We report the relative improvements we measured when our approach was used (stage 2), compared to the alternative approach (stage 1). With the p-level being below the value of 0.20 for all measures, we may report these relative improvements as being moderately significant. Still, the obtained results demonstrate that our approach can successfully identify contradictions in an automated way, and quickly guide users to the relevant parts of the data.

We observe that when users employed our approach in order to detect contradictions, they were able to identify contradictions faster, requiring 23% less time on average (ranging between 7% and 40%). The biggest improvements were for the topics “Ambien” and “Zune HD”, which had a few contradicting posts visible using our approach, but otherwise hard to discover. Our approach also led to a reduction by 28% of the number of time-intervals examined in order to identify contradictions (ranging between 12% and 42%). The largest reductions were observed for the topics “Zune HD” and “Internet Control” (38% and 42%, respectively), which contained many posts that did not take a position, or were off topic. For such topics, the helpfulness of

Dataset	**Topic**	D_1	D_2	**ΔD**	T_1	T_2	**ΔT**	N_1	N_2	**ΔN**	P_1	P_2	**ΔP**
Drug-Ratingz	Ambien	2.7	3.7	**50%**	142	87	**40%**	1.6	1.4	**12%**	0.70	0.81	**20%**
	Yaz	2.8	4.2	**58%**	85	79	**7%**	1.4	1.1	**22%**	0.75	0.95	**32%**
Slashdot	Int. Ctrl.	1.5	1.7	**17%**	283	218	**11%**	3.8	1.8	**42%**	0.37	0.63	**114%**
YouTube	Zune HD	1.4	2.9	**107%**	143	85	**32%**	3.7	1.9	**38%**	0.36	0.61	**109%**
Avg. improvements		2.1	3.1	***58%***	163	118	***23%***	2.6	1.5	***28%***	0.55	0.75	***69%***

Table 5.3: Performance of contradiction detection aided by our approach versus a baseline.

our tool mostly depends on the nature of texts, yet it was able to demonstrate time improvement in all cases. The average difficulty ratings were also favorable for our approach, which was consistently being marked as more helpful. This difference was most pronounced for the "Zune HD" topic (107%), which involved many posts. In this case, going through the posts was not easy, and our approach allowed users to focus their search and identify the contradicting posts.

It is interesting to mention, that for review datasets, such as "Ambien" and "Yaz", our tool was very helpful for users. However, the time improvement for the latter was considerably smaller, since it is more balanced between positive and negative opinions and manual discovery of contradictions in such dataset is relatively easy. The topic from Slashdot dataset featured the largest contradiction discovery time due to the size and complexity of texts, so it was rated as "very difficult" to work with. Still, taking into account the improvement in number visited regions, we may conclude that our tool is rather useful for the analysis of contradictions within massive amounts of data, featuring infrequent sentiment contradictions.

We report an additional measure of usefulness in Table 5.3: since both approaches aim at guiding the users to the time-intervals that are most promising for containing contradictions, we computed the percentage rate of success, P_1 and P_2, of the examined time-intervals that led to the identification of a contradiction, as well as the relative improvement, ΔP. We note that the above evaluation reports averages of precision values measured on per-user basis, when individual users were annotating the regions in time they navigated to according to their own analysis of trends (stage 1) or by our measure (stage 2). Assuming the normal distribution of errors, we can estimate that our tool provides a precision improvement greater than 0% at significance level of 0.02%. Other than that, it provides a 20% improvement at significance level of 5%, what we assume as rather good result.

The detailed results show that our approach was always more successful in suggesting to users time-intervals that contained contradictions, improving precision by nearly 70% on average (even more for the Slashdot and YouTube datasets). Our approach resulted in an average success rate of 75%, and was as high as 95% (for topic "Yaz"). Even though the approach by Chen et al. [19] (stage 1) was not designed with the contradiction detection in mind, it is still a good baseline for this task since it demonstrates two trends of opposite sentiments rather than the commonly used single average.

5.5 Conclusions

In this chapter, we formally defined the problem of detecting sentiment contradictions in texts with respect to the time dimension and formulated the two variations of it depending on the target analysis: single-topic and all-topics contradiction detection. Also, we introduced yet another specification of the above problem - synchronous and asynchronous contradiction types.

Within the scope of sentiment polarity, we proposed an approach of detecting sentiment contradictions for large-scale and noisy data sources, which is the first general and systematic solution to the problem. Our approach relies on a sentiment analysis technique that assigns a continuous sentiment value to each relevant topic of a text. Subsequently, the sentiment values are aggregated for each topic and across different time windows, organized in a tree structure, which can be efficiently queried to report contradictions according to a novel contradiction function.

We conducted an experimental evaluation with synthetic data, as well as three diverse real-world datasets and evaluated the usefulness of our approach with a user-study. The results demonstrate the applicability, usefulness, and efficiency of the proposed solution. In particular, the precision of our approach for contradiction detection reaches 80% in all our test scenarios, demonstrating that an unsupervised approach, designed specifically for large-scale datasets, can be effective. Again, the above result is comparable with the one from our user evaluation, although in this case our approach was tested against the "gold standard" of user annotations, rather than individual evaluations of each particular user.

While the contradiction function is based on the mean and variance, allowing us to compute it using incrementally updateable measures, it is not obvious that in a more general case of opinion contradictions the hierarchically aggregated clusters will still satisfy to Equation 5.1. For this task, we must consider metric functions, which not only allow computing Equation 5.1 from aggregated data, but also provide guarantee of this inequality. Provided that the above properties hold, opinion contradiction level C can be efficiently computed from the aggregate group size counts using Formula 5.2. Most of the proposed methods can be seamlessly applied to solve similar problems for opinion contradictions (e.g. adaptive threshold, synchronous/asynchronous contradiction types). Nevertheless, there are some technical aspects (CTree aggregating storage, updateable measures, aggregate opinion), which require a more careful modeling of opinions. In our further investigation, we are going to address the above problems and refine the proposed framework.

Chapter 6

Demographics Analysis

Aggregating sentiments for ad-hoc user groups is becoming necessary on the Social Web, where millions of users provide opinions on a wide variety of content. While several approaches exist for mining sentiments from product reviews or micro-blogs, little attention has been devoted to aggregating and comparing extracted sentiments for different demographic groups over time, such as "Students in Italy" or "Teenagers in Europe."

Nevertheless, the need to provide fine-grained analytics of social data is growing. Readily available users' demographics along with opinion data constitute a gold mine for extracting insights on what a particular user group thinks and how their opinion evolves over time and compares to opinions of others. This problem demands efficient and scalable methods for sentiment aggregation and correlation, which account for the evolution of sentiment values, sentiment bias, and other factors associated with the special characteristics of web data.

We propose a scalable approach for sentiment indexing and aggregation that automatically detects the right time granularity for computing meaningful sentiment correlations among various demographic groups. Furthermore, we describe methods for compressing the top-k correlations, leading to improved performance without significantly affecting the quality of the results. In addition, the data structures we use are incrementally updateable, making our approach suitable online. We present an extensive experimental evaluation with both synthetic and real datasets. Our experiments show the efficiency of our sentiment aggregation and demonstrate the effectiveness of the proposed algorithms.

6.1 Introduction

Today, sentiment analysis has become a platform that provides valuable information on people's opinions regarding different topics, and is widely used by businesses [61] and social study institutions [137]. Sentiment extraction and aggregation has been applied in various domains, from *movie reviews* to product *reputation management*. While multiple efforts focused on develop-

ing machine learning and statistical methods for characterizing sentiment within large bodies of text or for brief opinions and tweets [113, 126], not much attention has been devoted to analyzing the sentiment diversity observed in a large scale. Several studies indicate that the observed diversity of sentiment can be the result of demographic groups reacting differently to external events [126, 88]. The study of Thelwall et al. [126] indicates that changes in general sentiment are mainly caused by external events and thus are likely to be reflected synchronously in sentiments of various demographic groups. One more observation made by the same authors is that the changes in sentiment are particularly small, making it necessary to apply more sophisticated methods capable of detecting correlations under high noise conditions.

Evaluating the aggregated sentiment along users' demographics is a challenging task, which requires both precise sentiments (due to smaller aggregation levels) and efficient methods (due to increased complexity). Studies along this direction have traditionally focused on an offline analysis and aggregation of polling data for pre-determined demographic groups. Polling requires long-term monitoring of a large sample of the population in order to allow for a meaningful comparison of sentiments among demographic groups. However, it is difficult to organize polling on a large scale and conduct it regularly enough to analyze trends and correlations over time [100, 88]. Therefore, many scientists look towards evaluating online sentiments, especially considering their existing correlation with actual opinions.

Online sentiments monitoring has been approached by scientists using a variety of data mining algorithms, although these studies were not specifically accounting for relationships between demographic groups, their sentiment's correlation and hierarchical nature of demographics. Nevertheless, some recent studies have already made a step along this direction, uncovering interesting problems and observations which we addresses in our analysis. For instance, "A Demographic Analysis of Online Sentiment during Hurricane Irene" [88] revealed dynamic (temporal) sentiment differences between Southern USA and New England, and at the same time a constant (inherent) difference in the sentiments expressed by males and females, referred to as *sentiment bias*. Their study suggested a necessity to account for classification errors (sentiment noise) and sentiment biases, which we describe below.

Different demographic groups may have different points of reference when they express their sentiments for different topics. For example, while youngsters tend to prefer relatively cheap restaurants and are comfortable with a certain level of noise, pensioners generally prefer quieter and moderately priced restaurants. This problem has been studied in the literature and there exist methods which aim at extracting sentiment biases. Choudhury et al. [24] examine sentiment biases in blogosphere's communities, relying on entropy measure as an indicator of the diversity in opinions. The work of Das et al. [30] introduces complex mining of sentiment data in the form of ratings, where the authors aim at extracting meaningful demographic patterns, that describe groups with biased sentiments. Our work differs from the above since we

study the complementary problem of extracting groups with correlated sentiments over time, that is, groups that react similarly over time to external events. Nevertheless, this kind of analysis can also provide a more meaningful interpretation for the biases observed in the sentiments expressed by the different groups.

In addition, sentiments of demographic groups may evolve differently over time. For example, "French farmers" and "German farmers" had initially positive sentiments for the topic "organic farming", but later disagreed when the French government introduced additional taxes for organic goods superseding laws set by the European union and in disfavor of French farmers. In the above example, if the two groups have equal average sentiments for a specific topic during some time period, it will be hard to say if they really have the same attitude towards the events in that period, or if their equal sentiments are merely the result of an instantaneous convergence of otherwise diverse sentiments. This issue makes a straightforward estimation of inherent demographical opinions error-prone, and commands approaching the problem by analyzing the dynamics of sentiments [81].

Our examples suggest the need for sophisticated methods that can compare and correlate sentiments of demographic groups over time regardless of their inherent biases. Problems related to the identification of correlations among multiple time series have been studied by the data streams community, using a variety of techniques. These techniques focused on the efficient computation [157, 96], hidden variables [109], local correlations [110], pruning of candidate pairs [28], and lagged correlations [117]. Moreover, it is important to use efficient correlation methods, which allow online updates. From the above methods, StatStream [157] is the one that is closest to our work from the perspective of time series handling. StatStream computes correlations using sliding time intervals of specified sizes, composed of a number of sub-intervals of fixed length. It employs the *Discrete Fourier Transformation (DFT)* to compute correlations in an approximate and incremental manner. Our solution is different from the above works in a number of ways: (a) it analyzes time series using multiple aggregation granularities and detects correlations on ad-hoc time intervals; (b) it applies effective top down pruning both on time and demographics hierarchies; and (c) it uses correlation compression techniques to achieve efficiency and scalability.

One important aspect in the design of our methods is the definition of sentiment correlation as a function of aggregated sentiment over a time period. We explore Pearson's correlation using several variations of the average sentiment. In addition, we evaluate several ways of constructing a time interval of sentiment correlations and show that correlations remain meaningful and robust to noise when time intervals are assembled from smaller ones, which allows to apply efficient top-k and windowed correlation methods. Moreover, there are two computational challenges when implementing our methods. First, finding demographic groups requires the exploration of all possible combinations of values for demographics attributes - a task, that becomes

quickly intractable, especially for pairs of demographic groups. Second, in order to find correlations between pairs of demographic groups, one potentially needs to explore all sub-intervals of the input time interval. We show that both challenges render traditional database approaches inefficient, and describe algorithms that exploit the *lattice structure induced by demographics attributes* in order to prune the search space. Our algorithms also make use of *hierarchical time aggregation* to achieve efficient and scalable indexing and retrieval of aggregated sentiments.

In this part of our work we are focusing on the efficient aggregation of sentiment and computation of significant sentiment correlations for different demographic groups within dynamically determined time intervals. We base our algorithms for sentiment aggregation and correlation detection on a careful indexing of time and demographics into hierarchies. In order to enhance the performance of our algorithms, we describe analytical results that allow to prune the search space, while maintaining quality guarantees on results. Furthermore, we introduce two novel methods for correlation compression, which allow for the efficient implementation of our algorithms. We conduct an extensive set of experiments to validate our problem, and evaluate the performance of our solution. We use synthetic datasets, which contain large-scale artificial correlations with added noise, and the MovieLens real dataset, which comes with rich user demographics. The experiments demonstrate that correlated demographic groups can be identified very efficiently with the help of our specialized indexing storage and effective pruning. Finally, our evaluation provides interesting insights on correlations among real demographic groups in MovieLens.

The rest of this chapter is organized as follows. Section 6.2 defines our framework and the problem we tackle, while Section 6.4.1 develops the properties of correlation with respect to our problem. Section 6.4 describes our algorithms for correlated groups and correlation compression. Our user study and performance experiments are reported in Section 6.5. We conclude in Section 6.6.

6.2 Problem Definition

We are given a database $\mathcal{X}$ of records $x = (u,t,s,p)$ where $u \in \mathcal{U}$ denotes a user expressing sentiment $s \in$ [-1,1] on a topic $t \in \mathcal{T}$ in a time period p characterized with a start and end timestamps.

In our definitions we assume that sentiments are extracted for a given topic. For example, the record $x_1 = (u_1, Politics, 0.8, p_1)$ means that user u_1 expressed a positive sentiment (i.e., +0.8) for "Politics" during time period p_1. Such information can be extracted from the tweets of user u_1 during that time period. The record $x_2 = (u_2, Drama, \text{-}0.5, p_2)$ expresses a negative sentiment for "Drama" movies by user u_2 during time period p_2. This information can be computed from movie rating datasets such as MovieLens.

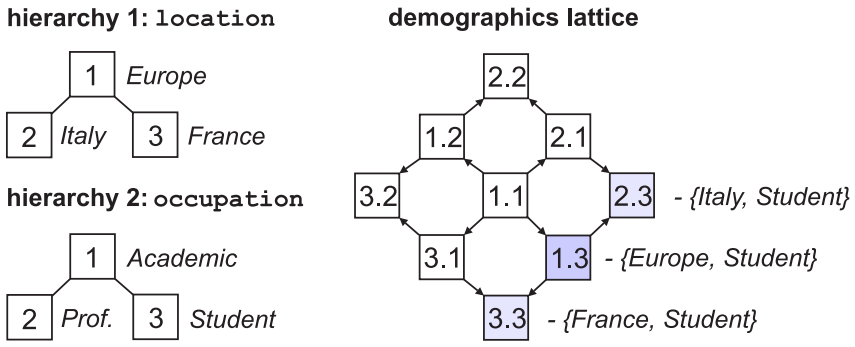


Figure 6.1: Two demographics hierarchies forming a lattice.

Definitions

We assume that each user $u \in \mathcal{U}$ is associated with a collection of values for a set of demographics attributes $\{a_i\}$. For example, *25* for attribute a_1:`age`, *Student* for a_2:`occupation`, and *Italy* for a_3:`location`. Each attribute a_i is associated with a demographics hierarchy $\mathcal{D}_i$ whose nodes hierarchically partition the set of values for that attribute. Correspondingly, each demographics hierarchy node contains all users from $\mathcal{U}$ whose attribute values are covered by that node. The top left part of Figure 6.1 shows an example demographics hierarchy associated with attribute `location`. The top node covers users from all available geographic locations (which corresponds to $\mathcal{U}$), and the descendant nodes partition those users into non-intersecting subsets according to their geographic locations.

Definition 12 (Demographics Criteria) *$\boldsymbol{d} = \{a_1 = d_1, \cdots, a_k = d_k\}$ is a set of predicates over demographics attributes a_i, where each predicate requires attribute's values to be contained in a node from a demographics hierarchy.*

For example, $\boldsymbol{d} = \{\texttt{age} : \mathit{Young}, \texttt{location} : \mathit{Italy}, \texttt{occupation} : \mathit{Student}\}$ refers to a combination of predicates on user attributes `age`, `location` and `occupation`. To simplify our notation, we will use $\boldsymbol{d} = \{\mathit{Young}, \mathit{Italy}, \mathit{Student}\}$ to refer to the same set of predicates. Values in demographics criteria correspond to hierarchy nodes and are therefore a fixed set, which can be enumerated. We note that user attributes with continuous values (for example, `age`) can be transformed to categorical values in order to induce a hierarchy.

Definition 13 (Demographics Generality) *Demographics criteria $\boldsymbol{d}$ is more (less) general than $\boldsymbol{d}'$ if: $\forall\,(d_i, d_i') : \quad d_i' \in d_i \quad (d_i \in d_i')$. We denote these relationships as $\boldsymbol{d} \prec \boldsymbol{d}'$ $(\boldsymbol{d} \succ \boldsymbol{d}')$.*

All demographics criteria and their generality relationships form a *demographics lattice L*, with a size equal to the product of the hierarchies' sizes. We show an example of such a lattice in the right part of Figure 6.1, where a nine-element demographics lattice is formed by all node

combinations of two hierarchies, each containing three nodes (shown left). The links that go from one lattice node to another indicate generality relations. For example, such criteria as $\{Italy, Student\}$ and $\{France, Student\}$ are less general than $\{Europe, Student\}$, which is itself less general than $\{Europe, Academic\}$.

Definition 14 (Demographic Group) *$\mathcal{U}^{\boldsymbol{d}}$, defined by demographics criteria $\boldsymbol{d}$, is a set of users $u \in \mathcal{U}$, who satisfy predicates in $\boldsymbol{d}$.*

An example of demographic group is European students defined by the demographics criteria $\{Europe, Student\}$, shown in Figure 6.1, right. In this work, we only consider groups of users, that can be defined using demographics criteria and use demographics criteria to denote demographic groups in all our equations.

Definition 15 (Group Sentiment) *Given a demographics criteria $\boldsymbol{d}$, a topic t and a time period p, we define the group sentiment of $\mathcal{U}^{\boldsymbol{d}}$ as an aggregation of sentiments s_x over records $x = (u_x, t_x, s_x, p_x)$ where $u_x \in \mathcal{U}^{\boldsymbol{d}}$, $t_x = t$, s_x is the sentiment of u_x for t, and $p_x \in p$: $s(\boldsymbol{d}, p) = \frac{1}{|x|} \sum s_x \mid \{u_x \in \mathcal{U}^{\boldsymbol{d}}, p_x \in p\}$.*

In the rest of this chapter, we assume that sentiments are aggregated and analyzed with respect to the same topic t, and therefore we omit t where applicable.

The main scope of demographics analysis is to analyze time-behavior of sentiment. Therefore, we need to consider a time series of sentiments, aggregated using fixed intervals to allow meaningful comparisons of individual points within, as well as across time series.

Definition 16 (Sentiment Time Series) *s is a sequence of values s_i computed by aggregating sentiments on a time interval p, using fixed sub-intervals p_i of the same length: $s_i = s(\boldsymbol{d}, p_i)$.*

Our further analysis of sentiment dynamics is centered on sentiment time series for a given group, topic and time period. Before going into details about the right time granularity and a sentiment correlation function $\rho()$, we consider the relations between demographic groups, and discuss how they affect the formulation of our problem.

Definition 17 (Maximal Demographic Group) *Given a sentiment time series similarity function $\rho()$, a threshold θ and a time period p, we call a demographic group $\mathcal{U}^{\boldsymbol{d}}$ maximal, if and only if: $\nexists\, \boldsymbol{d}' \prec \boldsymbol{d}$, $s.t.\ \rho(\boldsymbol{d}, \boldsymbol{d}', p) > \theta$.*

Intuitively, the above definition says that a demographic group is maximal if there is no other, more general group, that for the time period of interest shares the same sentiment behavior with the given demographic group. We define maximal demographic groups with respect to their

sentiment time series similarity, by analogy to the definition of maximal itemsets in frequent itemset mining.

Demographics relations may also be of a partial-overlap type, e.g., between {*Europe, Students*} and {*Italy, Academic*} (where *Europe* is a superset of *Italy*). However, we can argue that while for negative correlations all relationships between groups can be interesting, for positive correlations, generality and partial-overlap relations represent trivial cases of sentiment dependency. Sentiment correlations in this case can be caused by aggregating the same sentiments from the overlap for both groups. Therefore, we need to consider a disjoint type of relation.

Definition 18 (Demographics Disjointness) *between two demographic criteria* $\boldsymbol{d}$ *and* $\boldsymbol{d}'$ *is strictly opposite to demographics generality:* $\exists(d_i, d_i') : d_i \cap d_i' = \emptyset$. *We denote these relationships as* $\boldsymbol{d} \not\frown \boldsymbol{d}'$.

In other words, disjointness on any of the attributes makes the entire criteria also disjoint. Based on the above observations, for the present work we limit the scope of possible relations to those between non-overlapping groups and fully-overlapping groups only.

Problems

In this part of our framework we are interested in finding strong and significant positive and negative correlations among the sentiments time series of demographic groups. For the sake of simplicity, we will only work with positive thresholds, specifying the sign of the correlation if needed.

Problem 4 (Correlated Sentiment) *Given a period of time* p *and correlation* ρ_{min}*, find pairs of maximal disjoint demographic groups* $\{\boldsymbol{d}, \boldsymbol{d}'\}$ *and the longest time interval* $p' \in p$ *where their sentiments correlate:* $|\rho(\boldsymbol{d}, \boldsymbol{d}', p')| > \rho_{min}$.

Note that we can further restrict the answer set to only contain demographic groups whose correlation is statistically significant. That is, we require that $|\rho(\boldsymbol{d}, \boldsymbol{d}', p')| > \rho_{min}$ at a significance level r_{min}. We discuss this issue in more detail in the following sections.

A proper solution to the above-formulated problem will allow identifying sentiment behavior at a much finer level of detail than currently possible, finding cases that are counter-intuitive and can only be observed by processing huge amounts of data. We further discuss some interesting examples of such findings in Section 6.5, where we report the results of applying the proposed approach to the analysis of movie opinions.

6.3 Sentiment Correlation

Similarity of sentiments between demographic groups can be measured using a correlation coefficient of their sentiment time series. For instance, we can use Pearson correlation coefficient, which subtracts global sentiment averages and normalizes the resulting deviations, thus measuring the *linear dependency* among tested variables. This measure analyzes time series of demographic groups without considering their biases, though as we show later different averages and time intervals can be used, resulting in different semantics of identified correlations.

Definition 19 (Sentiment Correlation) *Correlation ρ of two time series s and s' of length n (with averages $\overline{s}$ and $\overline{s}'$) is defined as the normalized inner product $(s \circ s')_1^n$ of local deviations from averages:*

$$\rho = \frac{(s \circ s')_1^n}{n\sigma_s\sigma_{s'}} = \frac{\sum_{i=1}^{n}(s_i - \overline{s}) \cdot (s'_i - \overline{s}')}{\sqrt{\sum_{i=1}^{n}(s_i - \overline{s})^2 \cdot \sum_{i=1}^{n}(s'_i - \overline{s}')^2}}$$

Indices $_1^n$ in the inner product denote that it is computed using data points from 1 to n. Correlation ρ takes values in the interval $[-1, 1]$, where a positive (resp., negative) sign indicates that sentiments are changing in the same (resp., opposite) way and the absolute value measures the strength of the correlation.

Sentiment average can be computed in different ways in order to reflect different dependencies between time series. In addition, the period of time during which correlation is computed may vary depending on on whether the whole period is considered at once or if it is processed into sub-intervals. In the next two sub-sections we discuss different sentiment average computation and interval processing, later used in our algorithms. We use the examples in Figure 6.2.

Sentiment Averages

Global Average: This correlation detects co-variation of sentiments regardless of their sentiment bias. In Figure 6.2, a strong positive correlation between *a* and *b* is detected even though time series *b* is entirely positive.

Zero Average: When an average is substituted with a zero value, the correlation formula detects *polarity correlation* (e.g., between *a* and *c* in Figure 6.2). Polarity correlation indicates a much stricter dependency between sentiments: not only their local deviations, but also their signs (polarity) should be synchronized. As an additional benefit, it is easier to compute, as it does not need to compute average values of sentiments.

Local Average: If we compute correlation with local average (shown as dashed grey lines in time series *a*), we are able to detect correlation between local deviations. For example, the time series *d* has the same deviations of sentiment as the series *a* during the first period, and inverse deviations during the second period. These periods can be detected by computing correlation using sliding windows with the running average.

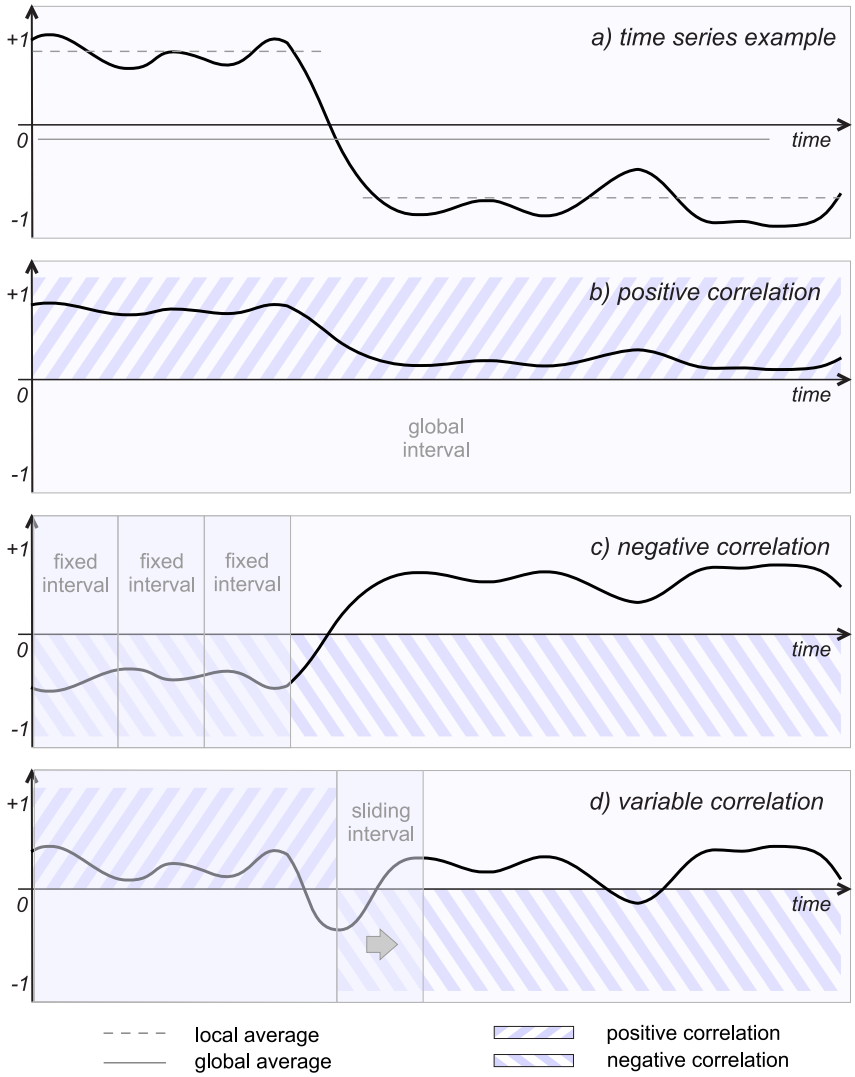


Figure 6.2: Examples of different correlation types.

Time Intervals

Global Interval: We compute a single correlation value for the entire input time interval. We observe, that for very long time intervals, global average sentiment is likely to be close to zero, so global average and zero average effectively become the same. The same is true for local average, which becomes closer to the global one with increased interval sizes.

Fixed Interval: The input time interval is divided into fixed-length sub-intervals.

Sliding Interval: The input time interval is divided into variable-length sub-intervals in a way that maximizes correlations. This goal can be achieved by optimizing sizes of correlation intervals, or by using a greedy algorithm that identifies intervals on-the-fly.

Correlation Significance

When computing the correlation value between two time series, we also need a measure that expresses how confident we are that this value accurately captures reality (e.g., we can be more confident that the correlation value between two particular time series is true when these time series are long than when they are comprised of just a few data points). This measure is the correlation significance. Given the correlation coefficient ρ, computed from n samples, we consider two *Null Hypotheses* about the correlation, which have the following test statistics:

Hypothesis H1 ($\rho = 0$): the value $z = \rho\sqrt{(n-2)/(1-\rho^2)}$
is distributed as the *t-distribution* with $(n-2)$ degrees of freedom.

Hypothesis H2 ($\rho < \rho_{min}$): the value $z = (Z(\rho) - Z(\rho_{min}))\sqrt{n-3}$
is distributed as the *standard normal*, where $Z(\rho) = artanh(\rho)$.

Definition 20 (Correlation Significance) *r is defined as the probability at which the considered null hypothesis is supported. For large n, the (one-tailed) significance of H1 and H2 is computed using the cumulative distribution function of the standard normal:*

$$r = 1 - \Phi(z) = 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{x^2}{2}} dx$$

While the first hypothesis is intended to verify if there exists any correlation between the two time series, the second hypothesis tests if they are correlated at least as high as ρ_{min}. For example, if $\rho = 0.9$, $\rho_{min} = 0.7$ and $n = 10$, we have that $r_{H1} \approx 0.0002$ and $r_{H2} \approx 0.06$. Relying on the significance threshold $r_{min} = 0.05$, H1 can be rejected as improbable, meaning that the two time series are indeed correlated. On the contrary, H2 cannot be rejected, signifying that ρ can be smaller than 0.7 and thus it is subject for pruning.

6.4 Method and Algorithms

In this section, we present our methods for storing sentiment time series for demographic groups and efficiently extracting correlations. We begin with the outline of our sentiment storage, followed by a description of our algorithms. Finally, we present smart pruning and compression techniques which take a full advantage of our storage and allow efficient problem solving.

6.4.1 Computing Correlations

Our algorithm for extracting significant correlations is based on the DTree storage and is listed in Algorithm 3. Due to lack of space, we present our algorithm for the sliding time interval and conventional correlation formula (see Section 6.4.1). The other approaches, using fixed or

global intervals, can be reduced from it by considering subsequent time intervals of constant sizes (fixed), or the entire time series (global).

The process of mining sentiment time series is performed in a top-down fashion, going from higher to lower granularities in a DTree and from root to leaf nodes in a demographics lattice. Our approach achieves hierarchical correlation management and pruning through remembering for every pair of groups which of the identified time intervals to `refine` or `exclude` at the next granularity level. Lines 13 and 16 of the algorithm only show the invocation of these functions, assuming that the corresponding time interval pruning takes an immediate effect on the next iteration.

In the second loop, we sequentially access the time series, while computing correlations between demographic groups in the third loop, where candidates are evaluated going from higher order nodes to their children. If a high correlation is detected for some candidate groups (line 12), then their positively correlated children are excluded, meaning that the corresponding lattice branches are not revisited in subsequent iterations of the loop. This pruning asserts the maximality of identified correlations and also reduces the candidate set.

The described algorithm employs the sliding interval approach, where correlation interval boundaries are determined in a greedy fashion: by comparing correlation coefficients between a forward (sliding) interval w_{next} of a fixed size, and an interval that runs from the previous boundary w_{prev}. We note, that while the sliding time interval w_{next} is updated for all demographics pairs as we scan the time series, the correlation time intervals w_{prev} are computed and maintained for each demographics pair individually, and their starting boundaries do not necessarily coincide (however, all their ending boundaries border with w_{next} while it slides). When global or fixed time interval approaches are used, it is possible to prune candidate demographic groups on-the-fly according to their estimated value of correlation, so that less and less computations are needed as we advance along the time series.

Finally, for all detected pairs of groups and correlation intervals, the algorithm can start the greedy generalization step, described in Algorithm 4. It iteratively supersedes groups with their maximal parents until they are disjoint and highly correlated.

Computing and storing correlation coefficients for all combinations of demographics nodes is only possible for small lattices, since it requires a quadratic space on the size of a lattice. But since we are interested in finding only high and significant correlations, it is possible to compute and store only such values, while still being able to answer queries with a good precision. In the following sections, we describe how correlation pruning and compression enable efficient implementation of our method.

In the following sections we describe an efficient way for computing correlations: first, by discarding insignificant results, and, second, by discarding correlations of children groups according to the maximality principle given in Definition 17. Furthermore, we propose effi-

Algorithm 3: Sliding algorithm for sentiment correlations.
Employs pruning using correlation estimates based on Lemmas 1-2.

Input : Time interval p, significance r_{min}, correlation ρ_{min}, lattice L, sliding interval size m
Output: demographic groups and correlation intervals

for $granularity = max \dots 1$ **do**
  for $p_i = \{p_1 \dots p_n\} \in p$ **do**
    $w_{next} = w_{next} + p_i - p_{i-m}$; //push next, pop last
    //w_{prev} are individual for each candidate
    //candidates are ordered by height($\boldsymbol{d}, \boldsymbol{d}'$) top-down
    for $(\boldsymbol{d}, \boldsymbol{d}') \in L \times L \mid \boldsymbol{d} \not\subset \boldsymbol{d}'$ **do**
      $w_{prev} = w_{prev} + p_{i-m}$; //update previous interval
      $\rho_{prev} =$ `correlation`$(\boldsymbol{d}, \boldsymbol{d}', w_{prev})$;
      $\rho_{next} =$ `correlation`$(\boldsymbol{d}, \boldsymbol{d}', w_{next})$;
      if $|\rho_{prev} - \rho_{next}| > \rho_{min}$ **then**
        //correlation interval is detected
        if $r(\rho_{prev} < \rho_{min}) < r_{min}$ **then**
          `refine` $(\boldsymbol{d}, \boldsymbol{d}', w_{prev}, granularity$ - 1);
          //exclude all correlated children groups
          for $(\boldsymbol{d}_1 \in \boldsymbol{d}, \boldsymbol{d}_2 \in \boldsymbol{d}', \rho > \rho_{min})$ **do**
          `exclude` $(\boldsymbol{d}_1, \boldsymbol{d}_2, w_{prev}, granularity)$;
        end
        $w_{prev} = \emptyset$;
      end
      //prune for the next granularity using Lemma 1
      $w_{BW} = p_{i-2..i}$; $\rho_{BW}(\boldsymbol{d}, \boldsymbol{d}', w_{BW}) =$ `Lemma1`(n);
      if $r(\rho_{BW} > \rho_{min}) < r_{min}$ **then**
        `exclude` $(\boldsymbol{d}, \boldsymbol{d}', w_{BW}, granularity$ - 1);
      end
    end
  end
end

Algorithm 4: Demographic group generalization algorithm.

Input : Correlation ρ, time interval p, demographic groups $\boldsymbol{d}, \boldsymbol{d}'$, maximality threshold θ
Output: Maximal demographic groups

//start from initial demographic groups complying to criteria
while $\boldsymbol{d} \not\subset \boldsymbol{d}'$ & `correlation`$(\boldsymbol{d}, \boldsymbol{d}', p) \geq \rho$ **do**
  $\boldsymbol{d} = \arg\max\{$`correlation`$(\boldsymbol{d}, parent(\boldsymbol{d}), p) > \theta\}$;
  $\boldsymbol{d}' = \arg\max\{$`correlation`$(\boldsymbol{d}', parent(\boldsymbol{d}'), p) > \theta\}$;
end
return the last $(\boldsymbol{d}, \boldsymbol{d}')$ complying to criteria;

cient methods of storing precomputed correlation values for fixed time intervals. We note that the proposed hierarchical pruning and correlation compression methods are applied on top of correlation values, and can be used in combination with various correlation algorithms. Some existing correlation methods [157, 96] can also be applied to our case, but are otherwise orthogonal to the pruning and compression methods discussed in this work.

6.4.2 Pruning Correlations

To find a pair of demographic groups with correlated sentiment, we have to evaluate all pairs of nodes in demographics lattice. However, we observe that correlation holds certain regularity properties on a demographics lattice and on time granularities, which are useful for pruning. We can apply pruning based on correlation estimates from the higher-granularity data (*vertical pruning*), and based on the observed part of the time series (*horizontal pruning*), as described below.

Vertical Pruning. Given the DTree, we would like to be able to estimate correlations for a smaller time granularity based on the averages computed for a higher time granularity. This is possible using the Spruill and Gastwirth correlation estimation method [120], which relies on the Bartlett and Wald regression estimator.

Lemma 1 *The estimate ρ_{BW} of correlation and its asymptotic standard deviation are computed using the following formula:*

$$\rho_{BW}(\boldsymbol{d},\boldsymbol{d}',p) = \frac{s'_{U3}-s'_{L3}}{s_{U3}-s_{L3}} \cdot \frac{\sigma(\boldsymbol{d},p)}{\sigma(\boldsymbol{d}',p)}, \quad \sigma(\rho_{BW}) = c\frac{1-\rho^2}{\sqrt{N}}$$

In the above equations, s_{U3} (s_{L3}) and s'_{U3} (s'_{L3}) are the averages of intermediate aggregates $s(\boldsymbol{d},p_i)$ and $s(\boldsymbol{d}',p_i)$ computed for $i \geq 2n/3$ ($i \leq n/3$), where n is the number of intermediate aggregates. The factor c is linearly depending on ρ and n and is estimated using the tabulation data given in [120]. We note that all standard deviations and intermediate aggregates used in this formula are directly accessible in the DTree at every granularity level.

Horizontal pruning. If both the correlation threshold ρ_{min} and the time interval p of size n are known, then for every subinterval $p_1 \ldots p_k$, $k < n$, with the corresponding inner product $(s \circ s')_1^k$, we can compute the upper bound of the correlation coefficient over p. We can then use this estimate to prune small correlations as more and more points of p are observed.

Lemma 2 *If δ_s and $\delta_{s'}$ are the maximum sentiment deviations and the inner product of sentiment deviations $(s \circ s')_1^k$ at point k is less than $(n\rho_{min}\sigma_s\sigma_{s'} - (n\text{-}k)\delta_s\delta_{s'})$, then $\rho(s,s') < \rho_{min}$.*

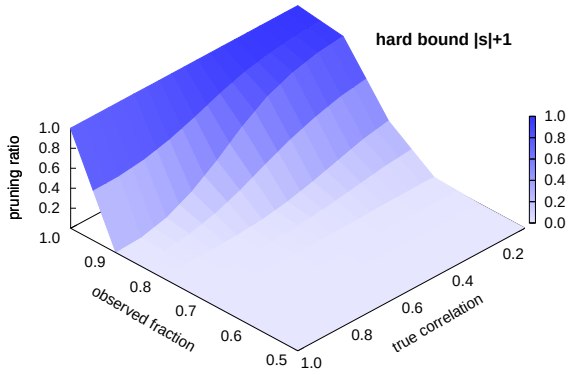


Figure 6.3: Performance of hard pruning.

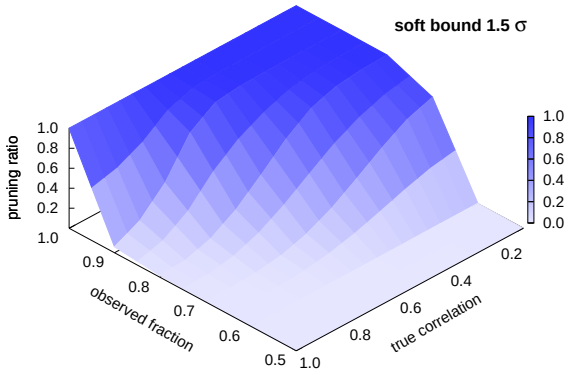


Figure 6.4: Performance of soft pruning.

Proof:

$$\rho(s,s') = \frac{(s \circ s')_1^k + (s \circ s')_{k+1}^n}{n\sigma_s\sigma_{s'}} < \frac{(s \circ s')_1^k + (n\text{-}k)\delta_s\delta_{s'}}{n\sigma_s\sigma_{s'}} < \frac{n\rho_{min}\sigma_s\sigma_{s'}}{n\sigma_s\sigma_{s'}} = \rho_{min}$$

We note that estimating maximum deviations is only possible for bounded time series. This is true in our case, where sentiments are distributed between [-1,1]. Thus, we can set $\delta_s = |\overline{s}| + 1$, which occurs when a sentiment is the most distant from the mean value (deviating in the opposite direction). This kind of threshold is essentially a hard (worst-case) estimate of sentiment deviations, which guarantees no false negatives during the pruning. Nevertheless, it is possible to achieve a more effective pruning when exploiting a soft deviation estimate, $\delta_s = \alpha \cdot \sigma$, which is based on the standard deviation σ and which error rate is controlled by α. Figures 6.3-6.4 demonstrate the performance differences between the two thresholds, when α is set to result in 5% of false negatives.

As in the case of vertical pruning, the standard deviations and mean values of time series, used in the above estimation, are stored at a higher granularity level in the DTree and thus directly available.

6.4.3 Compressing Correlations

Although the pruning techniques help to efficiently compute top-k correlations, the number of these correlations can sometimes grow very large. There exists a tradeoff between the precision and recall of storing top-k correlations, which is defined by size k. Improving both characteristics is only possible for larger top-k sizes. In contrast, performance and scalability requirements demand top-k size to be small. This problem can be addressed by compressing top-k correlations, as described below.

We propose two algorithms of top-k compression: a greedy algorithm of triangulation correlation compression, TCC, and clustering correlation compression, CCC, based on density clustering. Nevertheless, other existing methods of clustering and graph compression can be adapted to compress top-k correlations.

Triangulation correlation compression TCC

Given correlation coefficients between two demographics nodes and a third one, we can estimate upper and lower limits for the correlation between them. Based on the correspondence between correlation coefficients and angles of vectors, representing local deviations of time series to their mean, we can apply the triangular inequality, which gives us the following lemma:

Lemma 3 *If* $\rho(\boldsymbol{d},\boldsymbol{d}') = \rho_1$, $\rho(\boldsymbol{d},\boldsymbol{d}'') = \rho_2$ *and* $\rho(\boldsymbol{d}',\boldsymbol{d}'') = \rho$ *then*
$\rho_1\rho_2 - \sqrt{(1-\rho_1^2)(1-\rho_2^2)} \leq \rho \leq \rho_1\rho_2 + \sqrt{(1-\rho_1^2)(1-\rho_2^2)}$.

The detailed proof can be found in [72]. From the above inequality it follows that the transitivity of a positive and a negative correlation holds only if $\rho_1^2 + \rho_2^2 > 1$. This property requires absolute correlation values between two time series to be above 0.7 in order for the inequality to have any valuable prediction power. We note that this property is naturally achievable between nodes in a demographics lattice thanks to regularity and monotonicity of aggregated data. Therefore, Lemma 3 suits to our needs to compactly store correlations and recover missing values.

The simple greedy compression algorithm is listed in Algorithm 5. It removes elements from the top-k, which can be approximated using the triangulation principle. The compression process starts with a sorted list of correlations, which size is larger than k. Correlations are removed from the list one by one, being replaced with the next candidate in the list $(k+1)$ until the removal of any correlation introduces an error, larger than the one gained by adding a candidate. TCC algorithm can be further optimized by removing several correlations at once, until their approximations do not depend on each other. Such an optimization leads to a considerable performance benefit, since the approximation errors are not recomputed at every modification of a top-k list. However, the algorithm may become less optimal in this case. For the lack of space, we evaluate only the basic version of TCC, leaving possible extensions of this method for a future work.

Clustering correlation compression CCC

Correlation coefficient between time series of sentiment can be transformed to Euclidean Distance [157]. Relying on this distance metric, we can identify groups of time series, which are highly-correlated on the same fixed time interval. We propose to apply unsupervised clustering to find such groups and to compactly store only their average and pairwise correlations. Since

Algorithm 5: Triangulation correlation compression TCC.
Removes correlations which can be approximated using Lemma 3.

Input : demographics lattice L, number k
Output: Top-k correlations

for $(\boldsymbol{d},\boldsymbol{d'}) \in L \times L$ **do**
  add $\rho(\boldsymbol{d},\boldsymbol{d'})$ to $topk$;
end
sort $topk$ descending by ρ;
while *find* $(\boldsymbol{d},\boldsymbol{d'},\boldsymbol{d''}) \in topk$ *s.t.*
  $err = min|\rho(\boldsymbol{d},\boldsymbol{d'})$ - $Lemma3(\boldsymbol{d},\boldsymbol{d'},\boldsymbol{d''})|$ **do**
  if $err < |topk[k+1]|$ **then**
    remove $\rho(\boldsymbol{d},\boldsymbol{d'})$ from $topk$;
    add $topk[k+1]$ to $topk$;
    keep $\rho(\boldsymbol{d},\boldsymbol{d''})$ and $\rho(\boldsymbol{d'},\boldsymbol{d''})$ in the $topk$;
  else trim $topk$ to the size k; **break** ;
end

the space needed to allocate pairwise correlations is quadratic on the number of lattice nodes, replacing the correlations of individual nodes with those of their clusters can yield a significant compression ratio.

Because of the transitivity property of high correlations (according to Lemma 3), any set of highly correlated nodes is going to be densely packed in the Euclidean space, with a good cluster separation. Therefore, we find density-based algorithms of clustering more suitable, as their complexity in our case becomes asymptotically proportional to the number of elements in a cluster. Our clustering method uses the density-based algorithm DBSCAN [38], although any other distance-based algorithm can be used as well.

The compression process, described in Algorithm 6, starts with grouping lattice nodes into clusters based on their pairwise correlations. Clustering is performed using the absolute correlation values, and the sign of each node's correlation with respect to its cluster is stored and later recovered. Unlike in euclidean spaces, where a cluster has a mean value or a centroid, in the correlation space there are only distances between nodes available. Therefore, we replace individual correlations between nodes with average correlations between their clusters: for different clusters the average is computed from pairwise correlations between their nodes, and for nodes in the same cluster the average is computed across all pairwise correlations within that cluster. Finally, correlations between outlier nodes and clusters or between outlier nodes are added to the output list which is trimmed to fit the top-k size.

To achieve a good clustering, the density parameter should be set to a correct value, which is not known a-priori. As a minimum density parameter, DBSCAN uses a combination of neighbors range (which we substitute for minimal correlation) and their minimum number. We note that a lower minimum correlation corresponds to a broader neighbors range, unlike in original DBSCAN implementation. Figure 6.5 demonstrates an example of our parameters

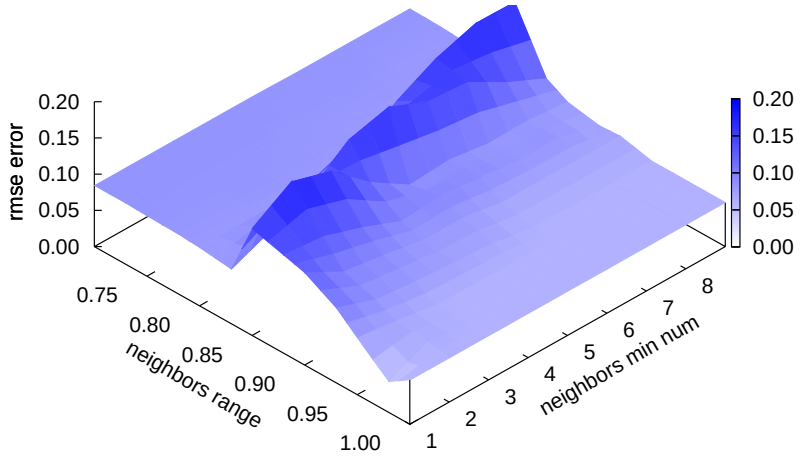


Figure 6.5: DBSCAN parameters space and optimization.

space and their corresponding compression errors for a fixed top-k size. We observe that for a broader range (Figure 6.5, left valley), DBSCAN tends to aggregate all nodes into one cluster.

Algorithm 6: Clustering correlation compression CCC.
Replaces correlations with cluster averages.

Input : demographics lattice L, number k
Output: Top-k correlations

$clusters = \texttt{DBSCAN}(L \times L)$;
for $i = 0;\ i < |clusters|;\ i{+}{+}$ **do**
  for $j = i;\ j < |clusters|;\ j{+}{+}$ **do**
    $\rho(i,j) = 0$; //add cluster-cluster correlations
    for $(\boldsymbol{d},\boldsymbol{d}') \in clusters[i] \times clusters[j]$ **do**
      $\rho(i,j)$ += $\rho(\boldsymbol{d},\boldsymbol{d}')$;
    end
    $\rho(i,j) = \rho(i,j)/|clusters[i] \times clusters[j]|$;
    add $\rho(i,j)$ to $topk$;
  end
  for $\boldsymbol{d} \notin clusters$ **do**
    $\rho(i,\boldsymbol{d}) = 0$; //add outlier-cluster correlations
    for $\boldsymbol{d}' \in clusters[i]$ **do**
      $\rho(i,\boldsymbol{d})$ += $\rho(\boldsymbol{d},\boldsymbol{d}')$;
    end
    $\rho(i,\boldsymbol{d}) = \rho(i,\boldsymbol{d})/|clusters[i]|$;
    add $\rho(i,\boldsymbol{d})$ to $topk$;
  end
end
for $(\boldsymbol{d},\boldsymbol{d}') \in L \times L,\ \boldsymbol{d},\boldsymbol{d}' \notin clusters$ **do**
  add $\rho(\boldsymbol{d},\boldsymbol{d}')$ to $topk$; //add outlier-outlier correlations
end
while $|topk| > k$ **do**
  remove the lowest;
end

The error in this case remains constant, since we approximate all correlations with a single value. When we increase the density parameter, the clustering error tends to grow due to outliers and since there are still not many clusters. Finally, for the optimum parameters, all the highly correlated nodes are clustered together and all the smaller correlations are represented by cluster distances, significantly reducing the compression error (Figure 6.5, right valley). Since it is not known which of the valleys contains the global optimum, we propose to broadly scan the space of possible DBSCAN parameters and then refine the optimum value using the gradient descent method. Nevertheless, DBSCAN is a one-pass method that relies on a precomputed distances index, and multiple clusterings used for optimization do not result in a significant performance degradation.

We note that top-k list can hold correlations not only between nodes, but also between clusters and between nodes and clusters. Depending on their presence in the top-k list and attribution to clusters, we retrieve correlation values in the way, described in Algorithm 7. It is also possible to apply a triangulation compression for clustering distances, creating a hybrid method that takes advantage of both TCC and CCC.

Algorithm 7: Top-k correlation retrieving method.

Input: demographics pair $(\boldsymbol{d}, \boldsymbol{d}') \in L \times L$
//Determine a cluster id for each node (if clustered).
if *clustered* **then** $\boldsymbol{d} = cluster(\boldsymbol{d})$; $\boldsymbol{d}' = cluster(\boldsymbol{d}')$;
//If the value for a pair of ids is present, return it.
if $topk(\boldsymbol{d}, \boldsymbol{d}') \neq null$ **then return** $topk(\boldsymbol{d}, \boldsymbol{d}')$;
//If the value is not present, estimate using Lemma 3.
$\rho_{low} = -1$; $\rho_{high} = +1$;
for *all* $(\boldsymbol{d}, \boldsymbol{d}', \boldsymbol{d}'')$ **do**
  $(\rho'_{low}, \rho'_{high})$ = `Lemma3`$(\boldsymbol{d}, \boldsymbol{d}', \boldsymbol{d}'')$;
  if $\rho_{low} < \rho'_{low}$ **then** $\rho_{low} = \rho'_{low}$;
  if $\rho_{high} > \rho'_{high}$ **then** $\rho_{high} = \rho'_{high}$;
end
return $(\rho_{low} + \rho_{high})/2$;

6.5 Experimental Evaluation

We ran experiments using both synthetic and real data. We first experiment with the synthetic data to evaluate the efficiency and performance of our algorithms, following it with the qualitative evaluation of correlations detected on a real dataset. We implemented our algorithms in Java, and ran the experiments using Java JRE 1.7.0 on a Windows machine with dual core 2.53 GHz CPU and 1.5 Gb of main memory.

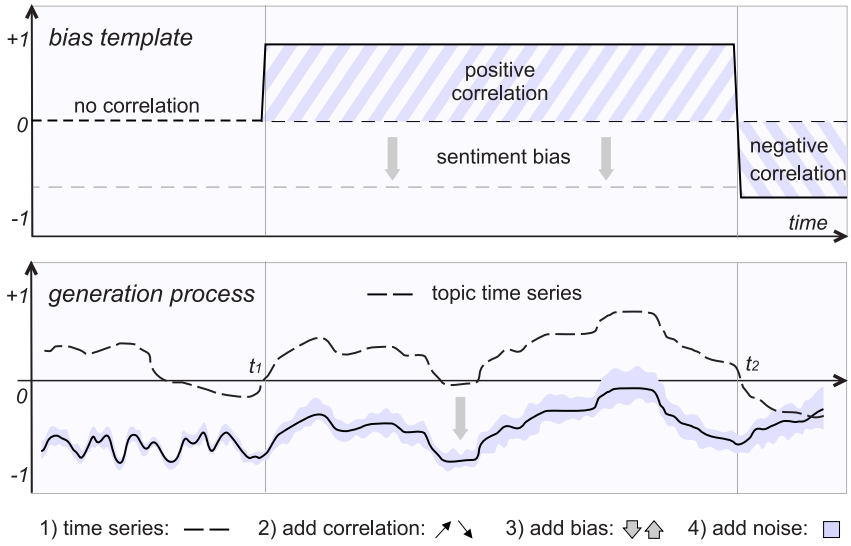


Figure 6.6: Generating biased sentiment time series.

6.5.1 Datasets

Synthetic Dataset

In order to accurately measure the precision of our system in identifying sentiment correlations, we conduct a series of experiments on synthetic data, containing time series of sentiments with artificially added positive and negative correlations, level biases and sentiment noise as demonstrated in Figure 6.6. We describe the layout of our dataset below.

Hierarchies. As a preliminary step, we generated a set of demographics hierarchies, for such attributes as age, gender, occupation, and location, containing 8, 3, 4 and 65 nodes respectively. Each node in every hierarchy was randomly assigned with a weight and a bias probability. Weights of nodes are distributed according to a *Zipf's* distribution, and normalized to add up to 1 at every level of a hierarchy. Nodes from different hierarchies, when combined, form a lattice of 6240 demographic groups as shown in Figure 6.1. The sentiment volume of each demographic group was taken as a multiplication of weights of attribute nodes, and its bias as a weighted sum of their individual biases. That way, we achieve natural regularity in the demographics lattice, providing a natural distribution of sentiments, which is necessary for a proper evaluation of extracting maximal groups and pruning.

Topic Sentiment. The dataset itself contains time series of sentiments generated independently for multiple topics over a time span of 8 years. Each topic is represented by a unique *topic time series*, produced using a *random walk* method, which aggregates *uniformly* distributed sentiments, whose timestamps follow *Poisson* distribution. We vary the parameter of rate for timestamps to produce faster or slower changing time series for each topic. Since sentiments for the topic time series are sampled uniformly, its mean value is close to zero in a long run,

meaning that it crosses the zero line a few times (for example, at points t_1 and t_2 in Figure 6.6). The original topic time series is stored for a randomly chosen demographic node, and we generate time series for other lattice nodes using individual *bias templates* for each of them. Bias templates contain sentiment bias levels and intervals of correlation with the topic time series (positive, negative or zero, randomly changed at "zero sentiment" points). The goal of our evaluation is then to correctly extract these correlation intervals between topic and biased time series, generated as described below.

Biased Sentiment. We produce a biased time series in a correspondence with the template, by copying (inverting) the topic time series in the case of positive (negative) correlation, or outputting randomized data otherwise (as seen in Figure 6.6, bottom). Following that, we add a certain positive or negative bias to the whole time series (shifting all the values) and the uniformly-distributed noise (for each value). Finally, we scale the time series to make sure that sentiments lay within the boundaries of [-1,1]. After generating a biased time series for the node, we insert raw sentiment data into the index in a proportion corresponding to node's volume. Moreover, we proportionally distribute these sentiments for all node's children (Figure 6.1), ensuring the regularity of sentiments in a lattice.

MovieLens Dataset

MovieLens dataset[1] consists of 1 million ratings left by 6 thousand users on 4 thousand movies. It also comes with rich demographics attributes: age, gender, occupation, and location, which we directly imported to our application. These attributes result in a lattice of over 30 thousand nodes, making almost *half a billion* possible pairwise combinations. We extracted the geographical location from postal codes, however the number of ratings for many nodes in this hierarchy was exceedingly small. Since we aim at extracting only significant results, we disabled the use of the location attribute in this experiment. We used five-star MovieLens ratings as sentiments, by mapping them to [-1,1] continuous sentiment scale, where one star corresponds to a highly negative (-1) sentiment, and five stars correspond to a highly positive (+1) sentiment, and other ratings are distributed evenly.

Since comments for movies usually appear during a period of their showtime and then fade out, we propose using genres as topics, thus providing a stream of sentiments with rather constant rate, where new movies serve a role of events, leading to sentiment changes. The dataset has 18 genres, and most of movies belong to several genres at once, with their ratings contributing equally to all of them. This results in a certain regularity of sentiments across topics and demographic groups and challenges the detection of interesting correlations.

[1] http://www.grouplens.org/node/73

6.5.2 Methodology

Efficiency evaluation is conducted on a synthetic dataset constructed as described in Section 6.5.1. It contains 10 topics with 400 biased time series for each of the topics, excluding children copies, while the much larger fraction of time series are generated randomly. We vary the level of noise added to time series in this dataset from 0.0 to 0.4 in absolute values, resulting in the same signal-to-noise ratios as sentiments are distributed on [-1,1]. We apply the scaling of time series after the noise was added.

We measure the average accuracy, precision and recall over all topics and all bias templates by measuring the correctness of correlation values over the extracted time intervals. For each of the time series, the extracted correlations are mapped to the binary scale $[\text{-}1, 0, 1]$ according to a 0.5 threshold, and compared to binary values stored in the corresponding template.

Precision is computed as the percentage of the length of extracted high-correlation intervals, which are found in the template as such (this is relevant for either +1 or -1 correlations). Recall is computed as the percentage of the length of high correlation intervals from the template, which were extracted as high correlations. Accuracy is computed as the precision of extracting all kinds of intervals from the template, including zero-correlation intervals. Finally, the root mean squared error (RMSE) is computed by measuring the actual differences between extracted correlation values and those stored in templates. It is computed as a square root of the average of squared errors, where the average is computed by weighting errors according to their time interval lengths.

6.5.3 Accuracy

We conduct the evaluation of accuracy to demonstrate the properties of the proposed correlation extraction methods, and their usefulness and efficiency when applied on noisy data. The observed behavior is not specific to our implementation alone. Rather, it marks the best possible performance for computing correlations using various fixed or sliding interval methods at particular aggregation levels (time granularity).

We evaluate the accuracy of correlation methods against noise for time granularities of 1 day and 10 days to demonstrate the effect of aggregation, and use additional measurements, such as precision and recall, to break down the observed performance for a more detailed analysis.

Baseline Correlation Methods

The results achieved by baseline correlation methods are depicted in Figures 6.7 and 6.8, respectively. We observe that the best accuracy[2] is achieved in the case of local average methods,

[2]The best achieved accuracy is not 100%, because some correlation intervals are smaller than the minimal correlation window.

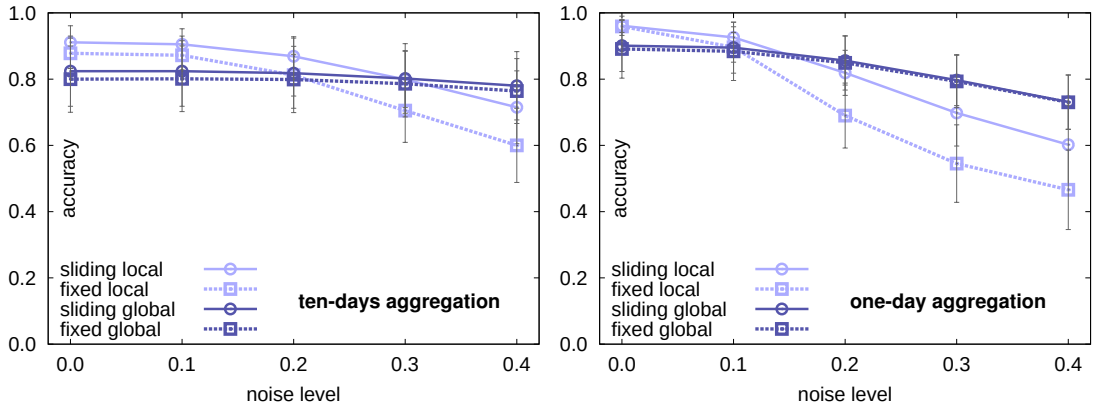


Figure 6.7: Accuracy of baseline correlations vs aggregation.

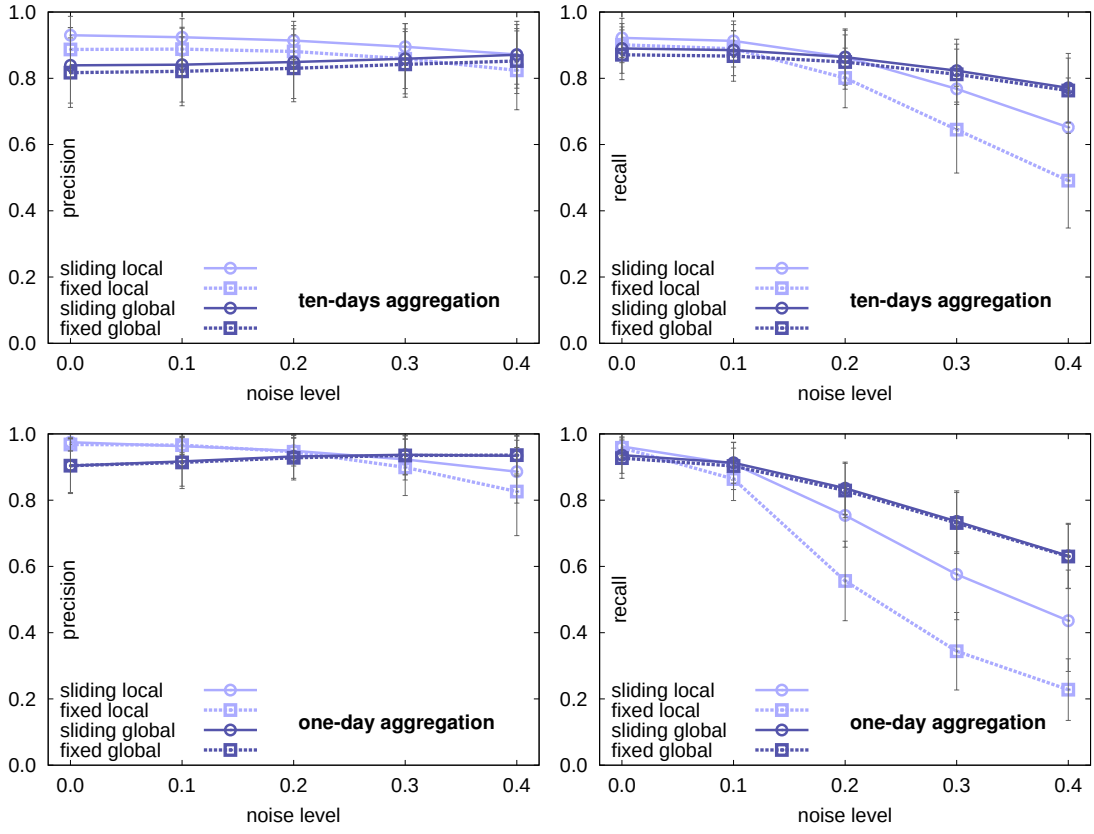


Figure 6.8: Precision and recall of baseline correlations.

albeit, only at a proper aggregation granularity (in our case, ten days). Using same methods at granularities that are affected by noise results in a substantial performance drop, as can be seen in Figure 6.7, right. Very high levels of noise, usually present at low granularities, can lower or even reverse the actual correlation between the time series, affecting precision and recall. On the other hand, increased granularity may reduce precision because of the discretization errors when identifying correlation interval boundaries. This can be observed by comparing zero-noise accuracy values for one- and ten-days aggregations: the latter shows slightly lower accuracy solely by the coarseness of time intervals. Using global average (computed for the whole period and same for all windows) has proven to be more noise-resistant although it is not always as precise as local average, and cannot be applied in ad-hoc scenario, requiring pre-specified global time interval. From the discussion above it is evident that correlations must be analyzed using sliding or fixed windows and at varying aggregation granularities.

Top-K Correlation Method

We evaluate the individual performance of *TCC* and *CCC*, by measuring their average compression error and its variance while varying top-k sizes from 1 to 10 times of their initial length. To construct the top-k list, we computed correlations over disjoint pairs of demographic nodes for fixed time intervals (with local averages), taken from 17 different topics in MovieLens. Then, we filtered correlations according to the significance and minimum correlation criteria, obtaining the lists of (approximately) 12K high ($\rho > 0.50$) and 2K very high ($\rho > 0.75$) top-k correlations for an initial set of 140K disjoint pairs.

Compression error is computed as the root mean squared error (RMSE) between actual correlations and those retrieved according to Algorithm 7. We note that an optimal compression (with the smallest error) for CCC clustering method was sometimes achieved with a size, smaller than that required by the compression ratio parameter. In such cases the remaining space was filled with the highest non-clustered correlations.

In Figure 6.9 we present the results of our evaluation. We observe that TCC triangulation compression shows better performance when it is able to fit all the high correlations necessary for describing the rest of correlations, what happens in the case of large initial top-k list ($\rho > 0.50$). In the case when all correlations are high ($\rho > 0.75$) and there is a high compression ratio, there is a large portion of correlations which do not fit into the compressed top-k list and neither can be triangulated from the correlations present in the list. The error in this case is the highest. On the other hand, CCC clustering compression benefits from compressing higher correlations as soon as there is enough space in top-k to store an optimal number of clusters. In this case most of the high correlations appear within clusters and the amount of correlations which are not approximated by cluster-cluster distances becomes relatively small. Nevertheless, CCC can become inefficient due to the clustering information overhead if there are many distanced

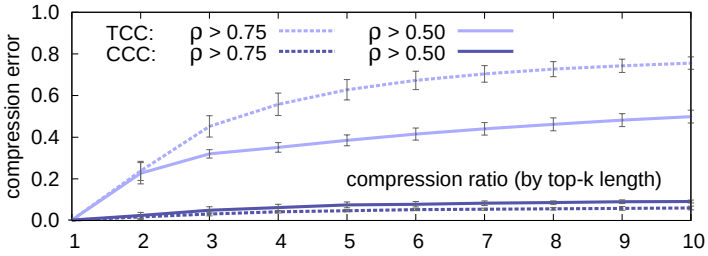


Figure 6.9: Compression error for top-k correlations.

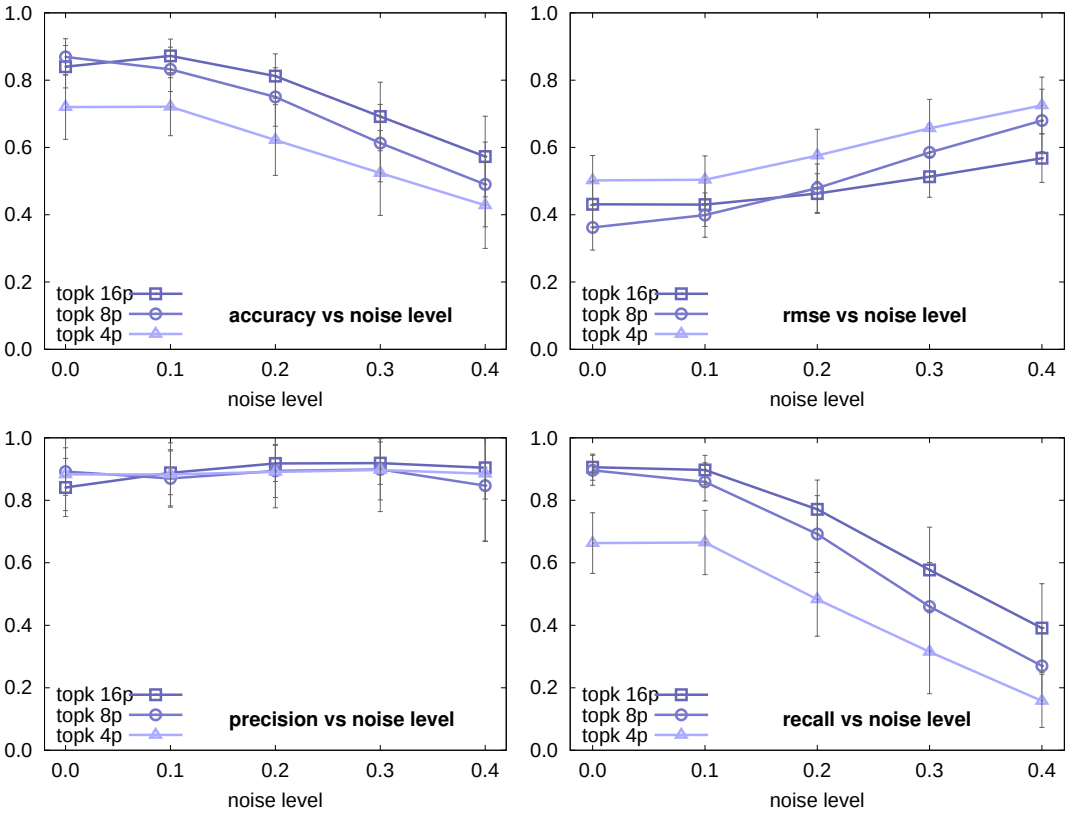


Figure 6.10: Accuracy of top-k correlations, 10 days aggregation.

small clusters of 3 items. Since TCC method is able to approximate the third correlation using the remaining two, it can be a good companion in such cases. We recommend to use hybrid CCC+TCC method for compressing correlations as the most universally applicable, especially in the case of moderate compression ratios.

We now look at the efficiency of correlation extraction of our top-k method using the same synthetic dataset, we used to evaluate the baseline methods. We calculated lists of high correlations ($\rho_{min} = 0.5$) for each of the fixed time intervals, containing 20-50K top-k values out of 15M (disjoint) and 40M (total) group pairs, and compressed them using the hybrid CCC+TCC method with clustering parameters optimized individually for the specific top-k size. We varied top-k sizes from 4 to 16 four-kilobyte disk pages (each page can hold up to 800 correlations or cluster distances). Figure 6.10 demonstrates that with the sufficient list of top-k correlations, computed for fixed-windows, it is possible to match the accuracy of conventional methods. A more detailed inspection shows that the drop in accuracy for smaller top-k sizes is caused mainly by a decreasing recall, due to the inability of top-k to fit all the high correlations, which our synthetic dataset is mainly composed of.

6.5.4 Performance

In Figures 6.11-6.12 we compare the time needed to extract correlation intervals using the same setup as in our accuracy evaluation.

In Figure 6.11, we report average times for the proposed methods using sliding and fixed time intervals (left), and the top-k technique (right). The time needed to compute correlations using sliding time intervals is approximately one third larger than the time taken by a fixed-interval method, since the prior needs to incrementally compute and compare correlations for two intervals: one is a fixed-length interval, sliding in front of the cursor, and another one is a dynamically expanding interval behind the cursor. The time of baseline methods remains fairly large since they compute correlations for a set proportional to $|L \times L|$.

In Figure 6.12, we demonstrate the effect of hierarchical pruning for fixed-interval correlations and compare it to the grid-hashing correlation pruning used by StatStream [157]. We note that StatStream cannot be applied for sliding-interval correlations when sliding intervals

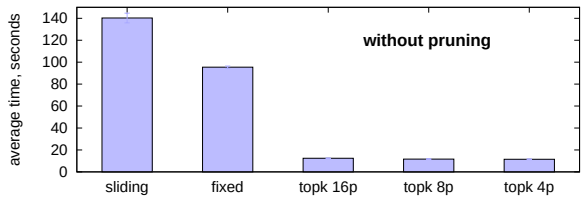


Figure 6.11: Performance of baseline methods.

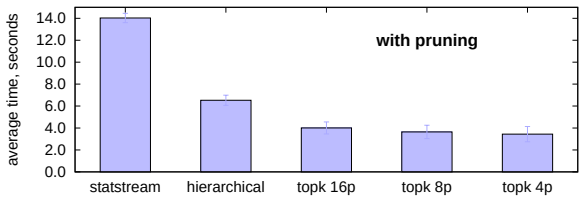


Figure 6.12: Performance of pruned methods.

among time series are of different lengths, as in our method. Moreover, the kind of time series approximation used in StatStream can not be used for bounded sentiments, where it results in numerous false positives at shorter interval lengths.

Both methods lead to significantly improved execution times in comparison to baseline methods, with hierarchical exhibiting better relative performance. The advantage of our method on highly correlated data is more pronounced with lower correlation thresholds, when maximality constraints allow earlier pruning. On the other hand, StatStream's performance benefits from better selectivity of higher thresholds and when time series are sparsely correlated, making it a good complementary approach.

Overall, we observe that the best performance is achieved by top-k correlations, which we report in both charts (with and without hierarchical pruning) for varying top-k sizes (16, 8 and 4 disk pages). It is evident, that top-k method is much faster even in the case of larger top-k sizes, where it matches the accuracy of direct methods. Furthermore, the top-k method demonstrates sub-linear scalability, sustaining almost the same performance even with exponentially increasing top-k sizes.

6.5.5 Usefulness

To demonstrate the usefulness of our approach, we automatically identified the highest (i.e., exceeding 0.9) maximal significant correlations between disjoint demographic groups in the MovieLens dataset. We note that a naive solution to this problem will require computing and comparing almost half a billion of time series. In Figures 6.13-6.14 we represent positive and negative correlations organized using a graph-like structure. Correlations are visualized as edges between demographic groups, labeled according to topics (genres). For brevity, we visualize only a small fraction (up to 10) of high correlations identified for each topic. We note that due to this filtering, some of the correlation edges are not present in the graph. This does not necessarily mean that such correlations are below the specified threshold. Finally, we do not report correlations between some highly-overlapping (but still disjoint) demographic groups, such as between {*Under 18*} and {*K-12 student*} or {*56+*} and {*retired*}.

We highlight nodes with the most unusual and interesting correlations, which took our particular attention, and provide their additional details in Table 6.1. For instance, in the top right corner of Figure 6.13 we observe a cluster of correlations on topic *animation* between the three very different groups: {*F, 56+*}, {*M,customer service*} and {*M,grad student*}. A more careful examination of this figure reveals that {*F, retired*} think of *thrillers* as {*M, 18-24, programmer*} - a short of a surprise, knowing that they have the same attitude to *romance* as {*M, K-12 student*}. Finally, {*academic*} and {*writer*} people have shown the most vibrant and unusual behavior in MovieLens dataset, producing many of the unsuspected anti-correlations we observed in Figure 6.14. Although some of these cases may look strange, we argue that the validity

of identified correlations can be in part confirmed by evaluating the trivial correlations, represented by white nodes in our figures (such as between non-intersecting groups $\{56+, retired\}$ and $\{50\text{-}55, other\}$). We leave the further exploration of our findings to a careful reader.

Such results can be used to drive the work in a number of directions: in sociology, researchers may investigate why these unexpected correlations exist, and examine more carefully and in greater detail the interests of users; in marketing, knowledge of group sentiments, how they change over time and how they are related to other groups could help expand existing markets, by influencing similar target groups, gaining a better understanding of the fan base, and monitoring the reaction of opposing groups; in collaborative filtering, new systems may expand their range of recommendations on particular topics with results from correlated groups, resulting in an enhanced user experience.

6.6 Conclusions

In this part of our work, we approach the novel problems of characterizing sentiment evolution in a demographic group and identifying correlated groups, which address the large-scale sentiment aggregation. We design efficient algorithms for sentiment aggregation based on a careful indexing of time and demographics into hierarchies and demonstrate that our problems can be solved effectively on a large scale using clever pruning, top-k and compression methods.

Our approach allows observing sentiment behavior at a much finer level of detail than currently possible, helping to identify cases that are counter-intuitive and can only be observed by processing large amounts of data. Moreover, it enables an unprecedented scale-up of traditional social studies and raises new data analysis opportunities, useful for sociology and marketing researchers.

We outline some interesting problems and extensions of the presented framework, which we plan to work on. We consider only a disjoint type of relation, although it is possible to expand the notion of relations between groups to any arbitrary path in a demographics lattice, and use it as a filtering argument to our problems. Also we are investigating the case where disjoint groups appear to be the same sets of users due to a strict dependency among attributes. Filtering high correlations between such groups is possible when their sets of users are known and can be done as a preprocessing step. Alternatively, we can compare the volume of sentiments between these groups, which becomes possible since our DTree storage preserves this information.

Figure 6.13: Positive sentiment correlations in MovieLens.

Figure 6.14: Negative sentiment correlations in MovieLens.

Topic	Group 1	Group 2	Begin	End	Correlation	Sign. H1	Sign. H2
Animation	{M,college/grad student}	{F,56+}	30.07.00	17.11.00	0.96	8.68E-07	9.02E-02
Animation	{M,K-12 student}	{M,25-34,executive/managerial}	07.11.00	07.03.01	0.94	2.67E-05	2.13E-01
Animation	{M,writer}	{F,homemaker}	28.10.00	13.10.01	-0.96	1.95E-08	0
Children's	{M,executive/managerial}	{F,artist}	30.07.00	07.11.00	0.94	2.03E-05	2.41E-01
Children's	{25-34,writer}	{45-49,executive/managerial}	08.10.00	15.02.01	-0.95	1.64E-05	1.28E-01
Children's	{M,Under 18,unemployed}	{F,18-24}	27.11.00	16.05.01	-0.94	2.90E-05	0
Children's	{25-34,self-employed}	{56+,academic/educator}	21.04.00	29.08.00	-0.93	4.28E-05	2.78E-01
Comedy	{F,25-34,sales/marketing}	{M,50-55,programmer}	17.11.00	13.09.01	0.97	4.11E-08	6.36E-04
Comedy	{customer service}	{45-49,artist}	18.10.00	26.01.01	0.96	3.25E-06	1.05E-01
Comedy	{F,35-44,self-employed}	{F,35-44,college/grad student}	07.12.00	16.04.01	-0.95	1.33E-05	1.28E-01
Adventure	{18-24,K-12 student}	{50-55,artist}	06.01.01	23.09.01	-0.95	1.30E-05	0
Fantasy	{45-49}	{M,Under 18}	17.11.00	03.09.01	0.95	1.44E-05	0
Fantasy	{M,technician/engineer}	{F,executive/managerial}	18.10.00	05.02.01	0.92	6.50E-05	3.71E-01
Fantasy	{F,18-24,college/grad student}	{F,25-34,other}	08.09.00	17.11.02	-0.92	6.80E-05	1.53E-01
Romance	{F,35-44,doctor/health care}	{M,56+,academic/educator}	17.12.00	02.11.01	0.98	4.23E-04	4.40E-06
Romance	{F,retired}	{M,K-12 student}	07.11.00	15.02.01	0.96	3.27E-06	1.05E-01
Romance	{M,artist}	{56+,academic/educator}	08.09.00	27.12.00	0.96	5.40E-06	9.02E-02
Romance	{F,35-44,technician/engineer}	{Under 18,other}	27.11.02	07.03.03	-0.96	3.69E-06	1.05E-01
Romance	{F,25-34,lawyer}	{F,50-55}	09.08.00	17.11.00	-0.96	4.25E-06	0
Drama	{college/grad student}	{academic/educator}	18.09.00	27.12.00	0.96	4.34E-06	1.05E-01
Drama	{M,45-49,sales/marketing}	{M,35-44,programmer}	30.07.00	17.03.01	-0.95	1.47E-04	0
Crime	{F,35-44,writer}	{M,35-44,doctor/health care}	05.02.01	29.08.02	-0.92	4.17E-03	1.95E-01
Action	{M,45-49,writer}	{F,56+,academic/educator}	30.07.00	26.01.01	0.96	3.55E-06	3.33E-02
Action	{F,programmer}	{M,customer service}	07.12.00	17.03.01	0.96	4.80E-06	1.05E-01
Thriller	{F,retired}	{M,18-24,programmer}	28.10.00	25.02.01	0.96	8.00E-06	7.76E-02
Thriller	{Under 18,K-12 student}	{F,56+}	17.11.00	25.02.01	0.95	9.07E-06	1.71E-01
Thriller	{M,25-34,technician/engineer}	{M,35-44,academic/educator}	17.11.00	27.03.01	-0.95	1.02E-05	1.28E-01
Thriller	{M,18-24,scientist}	{M,45-49,executive/managerial}	27.11.00	05.07.01	-0.95	3.75E-06	5.85E-02
Horror	{M,45-49,technician/engineer}	{M,executive/managerial}	17.11.00	07.03.01	0.96	3.48E-06	9.02E-02
Horror	{M,18-24,sales/marketing}	{F,25-34,executive/managerial}	21.04.00	07.03.03	0.96	3.28E-04	8.59E-07
Sci-Fi	{M,50-55,sales/marketing}	{M,K-12 student}	05.02.01	21.04.02	0.96	5.79E-06	1.21E-03
Sci-Fi	{M,45-49,programmer}	{M,academic/educator}	21.04.00	07.03.03	-0.98	3.07E-04	2.24E-01
Sci-Fi	{M,56+,programmer}	{F,Under 18}	16.01.01	11.01.02	-0.97	1.84E-04	1.94E-02
Musical	{M,college/grad student}	{F,45-49}	21.04.00	30.07.00	0.94	3.52E-05	0
Musical	{18-24,technician/engineer}	{18-24,lawyer}	07.11.00	03.10.01	-0.97	2.85E-06	0
Musical	{M,45-49,executive/managerial}	{M,45-49,academic/educator}	27.11.02	07.03.03	-0.95	1.53E-05	0

Table 6.1: Positive and negative sentiment correlations identified in MovieLens dataset.

Chapter 7

Dynamics Analysis

The analysis of user opinions expressed on the Web is becoming increasingly relevant to a variety of applications, ranging from monitoring of the blogosphere to product surveys. Following the large-scale aggregation of diverse sentiments with the analysis of contradictions, it is important to understand the underlying mechanisms which drive the evolution of sentiments in one way or another, for being able to predict these changes in the future.

In this chapter, we formulate a problem of identifying news events that caused dramatic changes of sentiments. We propose a novel framework for a complex news event modeling, which is capable of detecting time and longitude of events by observing a time series of event dynamics, such as news articles publications, and then correlating these data with a time series of any sentiment-based interestingness function.

The operation of the proposed framework is summarized as follows. First, we compute a sentiment interestingness time series, taking as an input raw sentiment data and interestingness function (e.g. based on the existing model of contradictions or sentiment volume). Second, we apply a deconvolution and probabilistic modeling to recover the time and longitude of the relevant news events. Third, we coherently analyze computed sentiment and news time series and automatically determine the time lag and the probability of their correlation/causality. Finally, we assign the corresponding news articles and evaluate them for a time interval of interest (identified using sentiment time series) to extract the essence of what happened.

7.1 Introduction

The problem of monitoring the evolution of sentiment on some topic, has been studied in the context of different research areas, from social studies to reputation management [130]. However, there is still a lack of understanding of what causes the community's sentiment to change. Some people change their attitude towards a topic because of their internal motifs, some others do so being influenced by their neighbors, but most likely people change their opinion when a new evidence comes into their consideration.

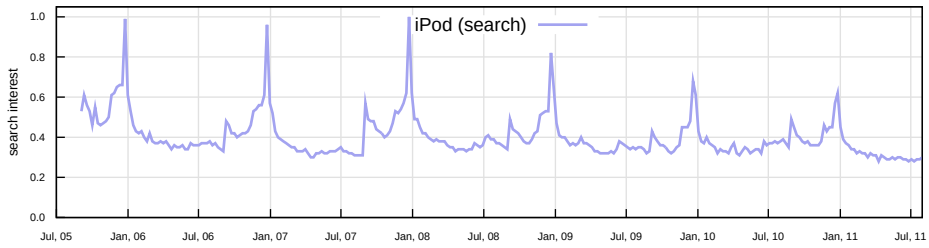


Figure 7.1: The search interest for the topic "iPod", outbursting during Christmas sales.

By aggregating sentiments, expressed in multiple texts, and assessing the result with statistical measurements, we can capture certain changes or shifts in global sentiment, which cannot be attributed to random variation. Understanding the reasons for these sentiment shifts is important to many applications and the corresponding problems have gained a lot of attention in scientific community.

For instance, we can model the global sentiment as if it was produced by a mixture of diverse (conflicting) opinions. Extracting such opinions is one of the desired targets of product reviews mining. Recently proposed methods can aggregate opinions expressed in customer reviews and extract their representative summary on a feature-by-feature basis. While representative opinions are likely to describe the meaning of the global sentiment at a *particular time*, their extraction requires complex text processing, making it practically impossible to monitor these opinions *over time* on a large scale.

An alternative way of understanding sentiment shifts can be in analyzing a (much smaller) correlated collection of documents for a possible explanation. Therefore, we propose studying the correlation of global shifts in sentiment to news from different news sources. Our objective is to understand and model relationships between sentiment changes and news events.

However, most of news events are announced as atomic pieces of information and their impact is not readily intelligible from text. To determine the importance of news to people, it is crucial to consider the publication dynamics of the whole social media, rather than only from news agencies or news media [126, 140]. Analyzing the aggregated publication volume on a specific topic over time can yield understanding event's importance and dynamics. However, social media can contribute to this volume all by itself (without any external stimuli) and also maintain a trending volume growth over long time periods. These effects distract the observed events dynamics and may even make them undetectable.

Our method addresses these problems by representing publication dynamics as the result of interplay between the original news importance and the social media's response. More specifically, our modeling is based of the idea that global news media can be described by a special "response" function, which determines the resulting dynamics of news publication or user interest for as event (like the exponential dynamics observed in Figure 7.1). This opens a possibility of recovering the original event sequence, its varying importance and time dimension.

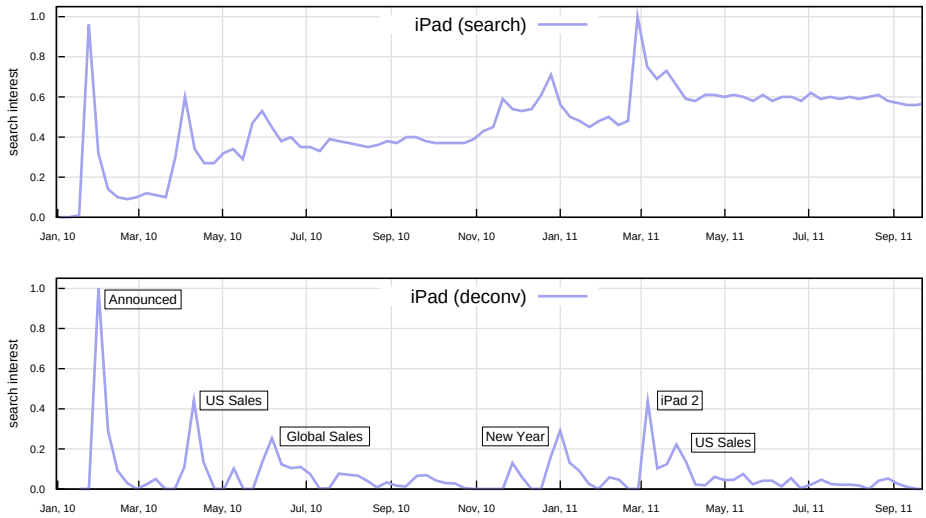


Figure 7.2: The effect of trend subtraction on Search Interest time series from Google.

Response function can be seen as a model of the reaction of mass media to an event, that is, it models a likelihood of the "delayed" publication. Much like in a phone conversation, where a delay in circuits creates the "echo" effect, news media tends to re-publish, cite, and discuss the previous articles, creating the unwanted noise. In this case, the peak intensity of publications does not always coincide with the beginning or a peak of the importance of the event. To tackle these problems and recreate the original event sequence we use a deconvolution, which is a widely used technique for improving audio or image quality.

In Figures 7.2-7.3 we demonstrate the output of our system for *search interest* and *news frequency* time series extracted from Google for the topic "iPad". In Figure 7.2, the top time series appears composed of a growing search interest (a trend) and a series of bursts on top of it, corresponding to several events. The top time series in Figure 7.3 demonstrates the volume of news publications with many bursts appearing on top of each other, and a substantial background volume, making it very hard to detect news events. Below, we plot the same time series processed using our methods. The output time series demonstrate a more vivid event separation, making them easily detectable, and much clearer event dynamics. Moreover, the bottom left time series appears without the trend. We note, that achieving the same effect is not possible by subtracting a trend computed using linear regression.

7.1.1 Motivating Scenarios and Examples

We would like to make the reader more familiar with our goals by introducing few use cases where applying our methods can help extracting and analyzing the dynamics of changing sentiments in a more reliable and principled way.

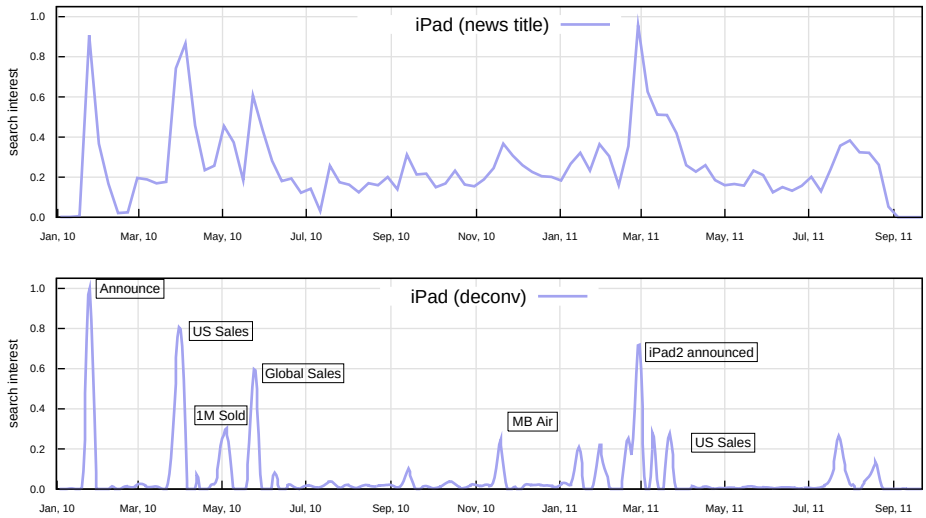


Figure 7.3: The effect of deconvolution on News Frequency time series from Google.

Scenario 1: Imagine that at some moment lots of people are tweeting about measles, because there is an outbreak. But if the government did not manage to get rid of measles quickly, and there happened an epidemic, people learn this from news and begin to write more negatively about it.

We can then identify the amount of social media content that refers to the news articles and study the corresponding opinion changes connected to these articles or, alternatively, we can filter out the postings that are reacting to the news story and this might provide more relevant information on disease outbreaks and their spread.

Scenario 2: Since the early announcement, Samsung's "Galaxy Tab" was regarded in social media as very fine competitor to Apple's "iPad", receiving mostly positive sentiments. However, the attitude of people has dramatically changed to negative at the moment when Samsung published its price quote for the device.

By observing the dynamics of the social media reaction to these and other impacting news, it becomes possible to predict changes in public opinion more rapidly - as soon as we are able to recognize the establishing trend.

7.1.2 Contributions

To the best of our knowledge, this is the first work that applies a thorough modeling of news distribution in various media and news interaction with sentiments. Our main contributions are summarized as follows. First, we develop a model of news event dynamics based on convolution, which allows capturing several important characteristic features of events and distinguish their types. Second, we assess several sentiment feature measures on their correlation with news

$S(t)$	Raw sentiment time series
$s(t)$	Sentiment feature time series
$e(t)$	Event importance time series
$n(t)$	News frequency time series
$x(t)$	Aggregated (total) news volume
$rf(t)$	Response function of an individual
$mrf(t)$	Response function of entire media

Table 7.1: Notations used in this section.

events of various types. Third, we propose a method of determining news events, which caused certain changes in sentiment. To achieve this goal, we build a classifier, predicting changes in sentiment based on event's features, and on the past history of correlation.

Since our framework relies on deconvolution, it can accommodate various response functions, suitable for different cases. We note that our method does not require describing the news publication dynamics by a differential equation. Instead, it can automatically learn the dynamics and its parameters from the data. Additionally, we propose a method of automatic event annotation from news articles based on contrasting the local and global popularity of keywords. To eliminate noise and make the above analysis more robust, we propose mapping news articles to events using a probabilistic model with automatically identified parameters.

The remainder of this chapter is structured as follows. In Section 7.2 we formally define the problem, and in Section 7.3 we discuss the related work. We present our approach for detecting sentiment shifts and extracting the corresponding news events in Section 7.4, and describe the experimental evaluation in Section 7.5. Finally, we conclude in Section 7.6.

7.2 Problem Definition

We summarize the most important notations used in our paper in Table 7.1 and discuss them in Section 7.2. Following that, we introduce our problems in Section 7.2.

Definitions

We are given a time series of numeric values, $s(t)$, which is derived from raw sentiments $S(t)$ for a particular topic T and represents some sentiment feature or interestingness measure. The example look of these series is demonstrated in Figure 7.4, a-b, where $s(t)$ represents the contradiction level of raw sentiments $S(t)$. Along with the sentiment time series, we are given the news frequency time series represented by $n(t)$ (Figure 7.4.c), and the corresponding correlation function $\rho(s,n)$, which takes both time series as input parameters and computes a real-valued correlation coefficient. We note that $\rho(s,n)$ can be a special function adapted for particular

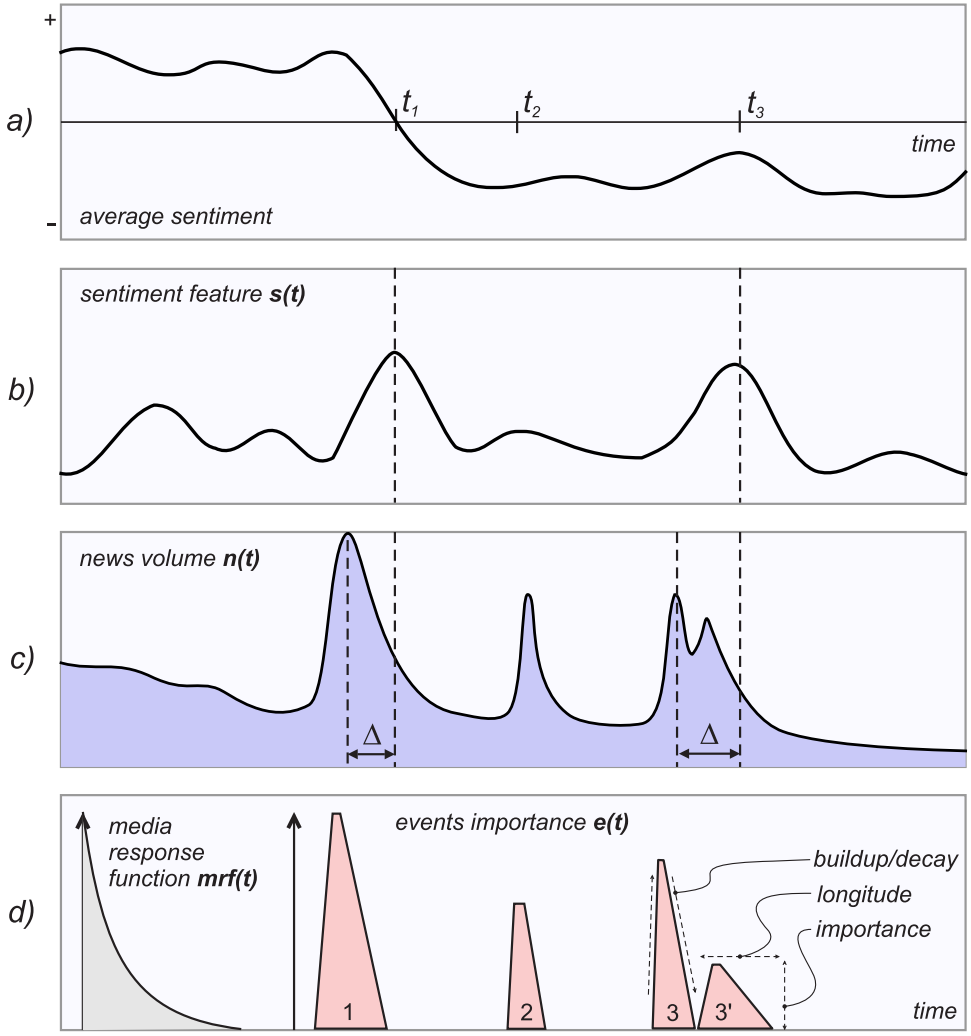


Figure 7.4: An example sentiment time series, demonstrating the correlation of contradiction value to a news time series.

measures used to compute $s(t)$ and $n(t)$. For instance, $n(t)$ can be rather bursty time series, where notions of a mean value and deviations from the mean, make no apparent physical sense. Thus, it can be more effective in this case to consider a correlation measure based on derivatives of time series or on their bursts alignment. Finally, we use the event importance time series $e(t)$, individual and media response functions $rf(t)$ and $mrf(t)$ (represented in Figure 7.4.d) to model events dynamics.

Problems

We are given a collection of sentiment interestingness time series $s(t)$ and news volume time series $n(t)$ for various topics. For a particular topic T, we want to detect the most interesting points in time according to $s(t)$, and extract and analyze the news events which have caused them, so that it will be possible to predict the future sentiment changes upon observing the relevant news events. This general problem can be decomposed into a set of sub-problems with a scope on the news-sentiment interaction and news-events interaction respectfully.

News and sentiments interaction can be coarsely modeled by correlation. However, the problem of finding pairs of correlated time series is inherently quadratic by itself, as it is not known if correlation occurs between time series for the same topic, or there may exist several topics which can influence on the sentiment.

Problem 5 *Given $s(t)$ for a topic T and a correlation measure $\rho(s,n)$, determine a list of $n(t)$ for various topics, which correlate with $s(t)$. For each correlation pair, determine a time lag between the two series, or a list of time lags, ranked according to the correlation coefficient, if there are several of them probable.*

After determining a substantial amount of news events, which caused sentiment changes, it may be possible to predict shifts in sentiment for related topics by recognizing features of the event and using them as input to a classifier model. Related topics can be determined by referring to the list of sentiment feature time series, which the news time series for this event correlates with.

Problem 6 *Given news event **e** at a time t and its features, predict a shift in sentiment, topic(s) and delays, for which the shift may occur.*

Predicting possible sentiment shifts can be done with the help of a classifier model trained on a dataset of news events. For training, this classifier can use either supervised data in the form of confirmed causality cases, or rely on automatically extracted news events and sentiment shifts (by their correlation).

7.3 Background and Related Work

To the best of our knowledge, there is currently no other system, or method, capable of identifying the causality between sentiment-derived time series and events, automatically annotating the most interesting sentiment shifts (or bursts) based on the relevant news events, and predict possible sentiment shifts based on the properties of event dynamics.

There exist approaches that tackle specific aspects of this problem, yet, they cannot be combined to solve the problem that our approach solves. For example, the most common aggregated sentiment measures do not provide a kind of time series which can readily correlate with news frequency. In addition, only few of the existing news tracking methods are able to differentiate between event types, or extract their additional semantics. Moreover, neither of them provides a correct time framing and importance level of the event, or other characteristics useful for event modeling or sentiment prediction purposes. Not only the current methods are incapable of identifying these important characteristics of an event, but also they are designed for specific kinds of news propagation media - blogosphere, microblogging, youtube. Therefore, these approaches cannot be considered as direct competitors.

In our case, opinion tracking is the most interesting from the perspective of contradiction analysis, which is a recently emerged problem being studied in different domains analyzing textual data. The particular opinion tracking methods which can be adopted to our problem are *sentiment volume* [126], *clustering accuracy* [140] and *contradiction level* [134]. The main focus of this part of our framework is then the analysis of news dynamics. Below, we describe some existing approaches and models, relevant to this particular part of our problem. We provide a detailed evaluation of their properties and design principles, and use it to establish a strong theoretical and empirical background of our method.

7.3.1 Models of News Dynamics

Lehmann et al. [73] study collective attention in Twitter and its propagation through user network. They measure the aggregated volume before, during and after the event's peak by subtracting the baseline level of attention (computed using a sliding window). Based on the relationship between these three values, they define four classes of news events according to their expectedness and impact. We represent these classes in Figure 7.5. In this figure, darker bars represent the observable volume of news over time, and lighter bars represent a reference shape for comparison. Accordingly, events can be: a) *expected impacting*, where there is a growth of volume before an event (anticipation) and a decay afterwards (response); b) *expected non-impacting*, where event's outcome is of a lesser concern than an event itself; c) *unexpected impacting*, featuring an instant appearance and a lengthy response; d) *unexpected non-impacting* or *transient*, where neither an event nor its outcome are important to the media;

Alternatively, Crane and Sornette [29] consider events as having internal (*endogenous*) or external (*exogenous*) origin and being of either *critical* or *sub-critical* importance to social media. Accordingly, they categorize dynamics of events into four classes, summarized in Figure 7.6. We observe that *exogenous critical* and *exogenous sub-critical* event types in their classification coincide with *unexpected impacting* and *unexpected non-impacting* event types from [73]. In addition, the authors also introduce a concept of *endogenous* events, which is broadly similar to memes in social media.

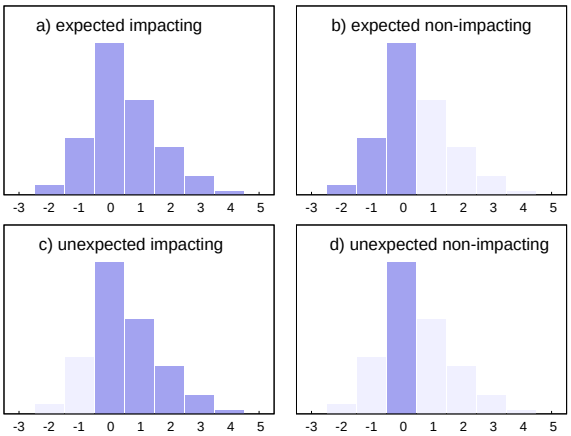


Figure 7.5: Classes of event importance from [73].

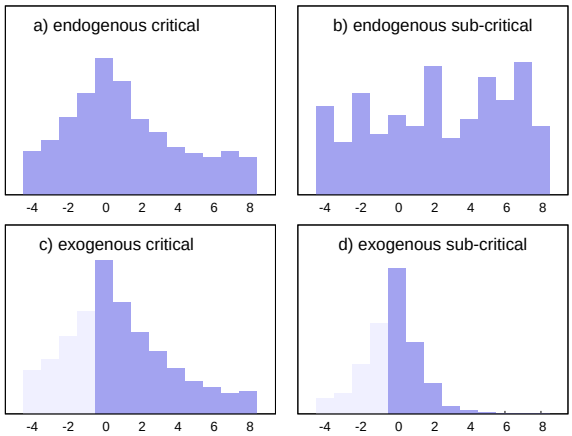


Figure 7.6: Classes of event dynamics from [29].

While these classes capture events semantics on a very good level, a more detailed analysis of news dynamics requires reliable extraction of peak shapes and proper modeling of social media. The first requirement is important since publication volume often contains noise and (evolving) background level, which masks individual peaks. The second requirement is necessary to distinct between the volume generated by endogenous factors (imitation) and that generated by exogenous factors (news importance). Therefore, we investigate the most recent models of news dynamics and their ability to correctly describe and predict data.

Meme Model

Memes in social media are the outbursts of publications on some topic, which can be assigned to “endogenous critical” type of the above classification. Since they have no particular external driving force or impact on sentiment, we want to distinguish them from “expected impacting” events. The topic of news dynamics in social media (and memes in particular) was studied by Leskovec et al. [75], who propose a model of meme and news dynamics based on the three assumptions for the interaction of news sources: *imitation*, *recency preference* and *concurrency*. Imitation (endogenous) hypothesis assumes that news sources are more likely to publish on events which have already seen larger volume of publications. Recency hypothesis marks the tendency to publish more on recent events. Finally, concurrency hypothesis states that news

sources have a limited capacity and choose only one event at a time to report on. The authors express imitation and recency preference through the functions of a combined news volume, $f(x)$, and a time passed since the beginning, $\gamma(t)$:

$$n(t) = \frac{dx}{dt} = c \cdot f(x)\gamma(t) \tag{7.1}$$

In the above equation, dx/dt represents the amount of publications at a time t, and x represents the total amount integrated since the beginning. We note that whereas equation (7.1) is based on the above assumptions, it is different to the equation studied in [75], which the authors erroneously formulated due to a misconception between x and dx/dt.

Leskovec et al. demonstrate that in the simulated environment consisting of concurrent news sources, both time and volume components are necessary to generate the oscillating nature of news volume. However, we did not find any evidence supporting the "global" positioning of the above equation with respect to time series modeling. Our evaluation of the top 100 time series published by Leskovec et al. reveal that global assumptions on time and volume formulated as the basis of their simulated experiment do not hold in real data. First, the global volume (accumulated since the beginning of the time series) is not useful to predict any but the very first peak. Second, the global time (either from the beginning of the time series or from the first peak) is both arbitrary (which peak is the first? when a time series begins?) and can not capture subsequent peaks. This said, we assume that the authors implied the global nature of their model with respect to parameters, while the time and the volume were local for every seeded event. However, we observed the irregularity of model's parameters for different peaks of the same time series. Normalizing the predicted volume over the top 100 time series (the concurrency factor) did not yield any improvement either for global or for local parameter scenarios. This can either indicate that the time series evolve independently of each other, or that top 100 time series is not sufficient to cover the whole publishing activity. Nevertheless, we believe that equation (7.1) intuitively captures some of the properties of news dynamics and therefore worth considering.

Stochastic Multiplicative Model

Asur et al. [6] propose a model for news dynamics, described by stochastic multiplicative process, driven by independent random variables (noise) and a time-decaying variable (recency). Based on this model, they predict a linear initial growth of publications volume, and a log-normal distribution of this volume over different time series. Both hypotheses are supported by the empirical evaluation on Twitter [6] and Digg [148] data. However, this model is not useful for prediction purposes, since it relies on a mixture of random variables (located at subsequent time intervals). Even if it is possible to infer values of these variables for past time intervals, the hypothesis that they are independently distributed forbids estimation of subsequent variables.

A closer look at this model reveals that it is formulated in a recursive manner, where the total volume accumulated by a time tick $t+dt$ is expressed through the volume at time t, i.i.d. random variable $\xi(t)$ and time-decaying component $\gamma(t)$:

$$x + \frac{dx}{dt} = [1 + \gamma(t)\xi(t)]x \tag{7.2}$$

To analyze the proposed dynamics, the above equation can be transformed into a more convenient form (remember though, that dx/dt, x and t remain discrete):

$$n(t) = \frac{dx}{dt} = \gamma(t)\xi(t)x \tag{7.3}$$

Now it can be clearly seen that the volume of news published at time t depends on the previously accumulated volume x, discounted by $\gamma(t)$, and on random variable $\xi(t)$. Although the dependency of (7.3) on random variables forbids its derivation in the analytical form, we observe that it is very similar to (7.1), and can also be analyzed as the product of exponent and time factors, especially in the case when $\xi(t)$ are not i.i.d.

Another important observation in favor of continuity of $\xi(t)$ is that they may represent the exogenous factor, the news importance $e(t)$, which pushes volume up and counteracts the decreasing trend of $\gamma(t)$. Comparing (7.1) and (7.3) under this perspective, it becomes evident that they are essentially the specific cases of a more general model, multiplying endogenous and exogenous factors. Our model differs from these two by considering a convolution between endogenous and exogenous factors.

Hawkes Poisson Process Model

A study of social system's dynamics by Crane and Sornette [29] comes the closest to our work with regard to a modeling based on users response. The authors study dynamics of book sales [119] and social content [29] based on a widespread model of hyperbolic (long-memory) user response function $rf(t) \sim 1/t^{1+\theta}$, $0 < \theta < 1$. Taking the ensemble average of a Hawkes Poisson Process driven by this response function and a spontaneous rate $e(t)$, they express $n(t)$ being conditional on itself and on an average branching ratio μ:

$$n(t) = e(t) + \mu \int_{-\infty}^{t} rf(t-\tau)n(\tau)d\tau \tag{7.4}$$

Following that, the authors derive the resulting response function of social media to exogenous and endogenous events by considering the output of (7.4) in the case when $e(t) = \delta(t)$ (Dirac function):

$$mrf(t) \sim 1/t^{1-\theta} \quad \text{(exogenous critical)} \tag{7.5}$$

$$mrf(t) \sim 1/t^{1+\theta} \quad \text{(exogenous sub-critical)} \tag{7.6}$$

$$mrf(t) \sim 1/t^{1-2\theta} \quad \text{(endogenous critical)} \tag{7.7}$$

Correspondingly, the equation (7.4) takes the form of the convolution between event importance $e(t)$ and a media response kernel $mrf(t)$:

$$n(t) = \int_{-\infty}^{t} mrf(t-\tau)e(\tau)d\tau \tag{7.8}$$

Summarizing the above studies, we see that the imitation factor does not play as important role in news dynamics as in meme dynamics. Moreover, [6] and [73] observe that propagation of news through user network is not epidemic, i.e. it rather depends on news importance than on numbers of followers of users who spread the news. Therefore, we consider publication likelihood being dependent on a recent volume more than on a past volume, deviating from the purely endogenous model. In addition, we consider that news events have continuity and varying importance, which also affect the publishing dynamics (exogenous assumption). These assumptions require a more complex modeling of news dynamics, but result in more accurate models, which we discuss in the following sections.

7.4 Method

The operation of the proposed framework is summarized as follows. First, we compute a sentiment interestingness time series, taking as an input raw sentiment data and interestingness function (e.g. based on the existing model of contradictions or sentiment volume). Second, we apply a deconvolution and probabilistic modeling to recover the time and longitude of the relevant news events. Third, we coherently analyze computed sentiment and news time series and automatically determine the time lag and the probability of their correlation/causality. Finally, we assign the corresponding news articles and evaluate them for a time interval of interest (identified using sentiment time series) to extract the essence of what happened.

Our problem requires collecting and processing two different kinds of data, such as sentiments and news, which come from different sources and also at a very different rate. Nevertheless, the output time series should be aggregated using the same rate for the purposes of correlation. Therefore, we represent our method as a composition of processes, each using dedicated data-specific methods, yet having common input and output specifications, thus allowing to interchange the algorithms in a framework. In Figure 7.7 we aggregately represent the func-

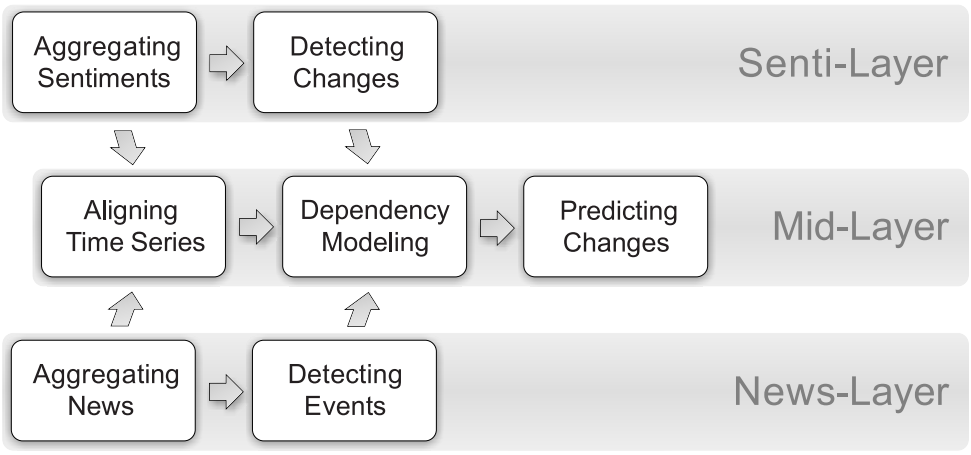


Figure 7.7: A diagram showing the key components of the problem, their organization and interaction. Arrows show the logical dependency between components.

tionality of our method by a few major components, allocated into the three main layers: *News Layer*, *Sentiment Layer*, and *Middle Layer*.

The first layer, *Sentiments*, takes care of aggregating sentiments for a topic and detecting interesting changes, which can be contradictions, outbursts of sentiments' volume or other changes in sentiment happening over time. Our framework allows using arbitrary sentiment aggregation functions, depending on the demands of the particular analysis.

As for the *News* layer, we have to aggregate the volume of the news for a topic into a time series, which will further be analyzed to detect news events. To detect news event and extract its features we can perform a deconvolution of the corresponding time series or its relevant fragment. On the other side, annotating a particular event involves aggregating and analyzing the news articles, which have been assigned to this event by our model.

Finally, both layers provide time series data for the *Middle Layer*, which, given a proper measure of correlation, automatically aligns the time series according to their time lag, and provides means of navigating to the corresponding time intervals in both series, as well as their degree of causality through dependency modeling. It also contains models, which predict if a given event can cause shifts in sentiment, and use dependency modeling to tell for which topics and at which time this may happen.

For sentiment extraction and contradiction detection our problems remain the same: topic-induced noise and classifier-induced noise. For example, if all media call "Galaxy Tab" a "tablet", and the one being observed calls it "slate", it can neither be used for sentiment extraction, nor contribute to news popularity. It can be fought by making a topic (feature) identification more reliable. The classifier-induced noise should be addressed by improving the sentiment extraction.

7.4.1 Correlating News and Sentiments

We consider several measures of the aggregated sentiment features $s(t)$, which we call sentiment interestingness time series. These time series require different correlation methods $\rho(s,n)$.

In the case of continuous and smooth time series with the characteristic pendulum behavior (e.g., average sentiment), we can use the Pearson cross-correlation coefficient, which is defined as the normalized covariance of the two time series:

$$\rho(s,n,\delta) = \frac{Cov[n(t),s(t+\delta)]}{\sigma_s \sigma_n} \tag{7.9}$$

However, Pearson correlation is intended to determine a linear dependency between variables, which is hardly observable for bursty time series such as unexpected events or sentiment contradiction. Such time series do not have a definite average level, around which the movement is happening. Instead, their values are outbursting from the minimum level at some points in time. This behavior requires a special correlation technique, which takes as input only the bursty points within a specified time interval. Let's assume that sets of bursts for sentiment and news are denoted as S_t and N_t respectively. Following this, any kind of binary similarity measure can be applied, for example *cosine similarity* or *Jaccard coefficient*:

$$\rho(s,n,\delta) = \frac{|S_{t+\delta} \cap N_t|}{||S_{t+\delta}||||N_t||}, \quad \rho(s,n,\delta) = \frac{|S_{t+\delta} \cap N_t|}{|S_{t+\delta} \cup N_t|} \tag{7.10}$$

In the above equations, intersecting bursts are determined according to some proximity region ξ, and one of the time series is shifted in time by some constant delay δ. In addition to counting the number of overlapping bursts, we can apply burst weighting, for example based on their magnitude.

We consider that sentiment changes may be preceded by the news events with some delay. In order to align the two sequences, we have to determine the time lag between them, which is generally different for different domains (compare home appliances and cell phones). It can be determined by maximizing the cross-correlation coefficient:

$$\Delta = \arg\max \delta \left[\rho(s,n,\delta)\right] \tag{7.11}$$

where δ is the time lag constant (unknown). However, a direct optimization method has certain computational inefficiency, since it requires computing the correlation for every candidate parameter δ. Moreover, a news time sequence can be the result of a repetitive or regular process, leading to a non-monotonic cross-correlation. To reduce the negative effect of both problems, we need to develop an effective way of estimating the boundaries for the the time lag.

7.4.2 Detecting Impacting Events

As we already noted, not every kind of publications outbursts is caused by external news. For instance, "endogenous critical" and "expected impacting" events may produce a similar response in social media, yet only the latter one is relevant to our study. Moreover, not every kind of news dynamics has an impact on sentiment, so we want to distinguish them with a very fine level of detail. Said this, we introduce our model for social media and news dynamics. We start with the description of basic social media responses and our representation of events importance. Following that, we introduce a novel method to extract important properties of events, which is based on deconvolution, and propose methods to estimate parameters for this process.

Modeling News Dynamics

We model the observed news dynamics (frequency of publications) as a response of social media to external stimuli. We represent the output as a convolution of the two functions: news events importance sequence and a media response function:

$$n(t) = \int_{-\infty}^{+\infty} mrf(\tau) \cdot e(t-\tau)\, d\tau \tag{7.12}$$

where $mrf(t)$ is the media response function (in general, decaying), and $e(t)$ is the actual event sequence, which is unobserved. We demonstrate the look of both functions in Figure 7.4.d.

However, in order to recover the original event sequence, we need to perform a deconvolution of the news frequency time series - the task, for which we should know an exact shape of $mrf(t)$. Our assumption here is that news events become obsolete and cease being published very soon after their appearance. Another reason why this happens is because of the satura-

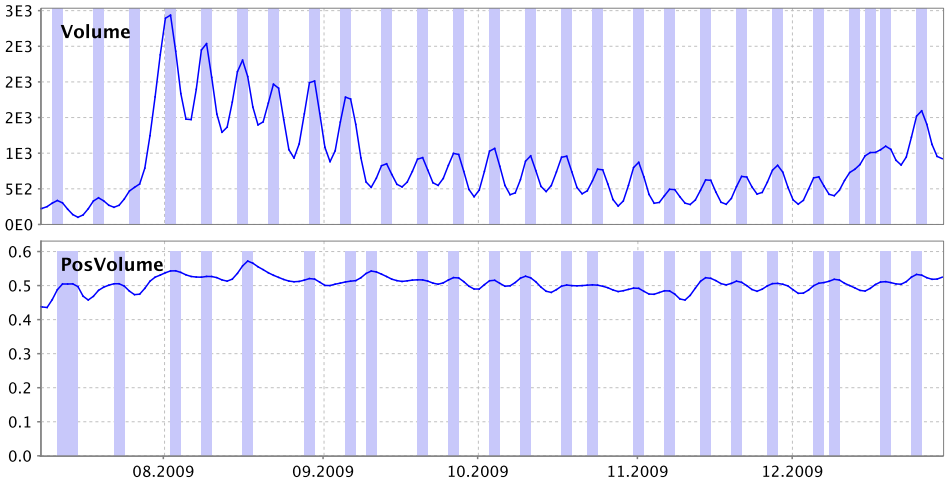


Figure 7.8: Correlated bursts of $n(t)$ and $s^{+}(t)$ for the movie "Hangover".

tion of the media: the likelihood (the temporal rate) of news publication is usually inversely dependent on the number of news, which have been published previously on the same event. Some external evidence which proves this assumption is shown in the Figure 7.2, where the time series of keyword search popularity demonstrates exponential decaying of popularity.

To model this behavior, we propose using a family of normalized decaying functions, which have the aggregated volume equal to 1.0 and which are defined on $t > 0$ using the Heaviside step function $h(t)$. For instance, these functions can be *linear*, *hyperbolic* or *exponential*, as demonstrated in Figure 7.9.

$$mrf(t) = \left(\frac{2}{\tau_0} - \frac{2t}{\tau_0^2}\right) h(t)h(\tau_0 - t) \tag{7.13}$$

$$mrf(t) = h(t)\frac{\alpha-1}{\tau_0}\left(\frac{t+\tau_0}{\tau_0}\right)^{-\alpha} \tag{7.14}$$

$$mrf(t) = \frac{1}{\tau_0}e^{-t/\tau_0}h(t) \tag{7.15}$$

Linear Response (7.13). In the case of a linear response, the probability of publishing on an event linearly decreases with time, and the media cease publishing on events after the finite cutoff time τ_0. Linear response is characterized by a constant rate of content generation, and results in nearly linear dynamics of news volume for spike event shapes. Since this kind of dynamics was observed by [6] for event buildups, we are interested in evaluating it on our data.

Exponential Response (7.15). For the exponential response, the decay is initially more rapid than the linear, but becomes less pronounced towards the end, with the probability reaching 0 in an infinite time. Here, τ_0 parameter is having the sense similar to half-life time of probability, and the rate of probability decline is proportional to its current value.

Hyperbolic Response (7.14). In the hyperbolic (power law) response case, the probability follows a more pronounced decay than the exponential, decreasing very rapidly when the time τ_0 is small, but having a long tail afterwards. Here, parameters $\alpha > 1$ and $\tau_0 > 0$ control the sharpness of a response. Hyperbolic response is very interesting since it can reach infinity in a constant time (unlike the exponential) and in the case of $\alpha \approx 1$ its rate of decline is proportional to the square of its current value. Curiously, this type of dependency is known to occur in well-connected systems, where the response is proportional to the number of interactions between individuals, opposed to exponential dynamics, where it is proportional to the number of individuals.

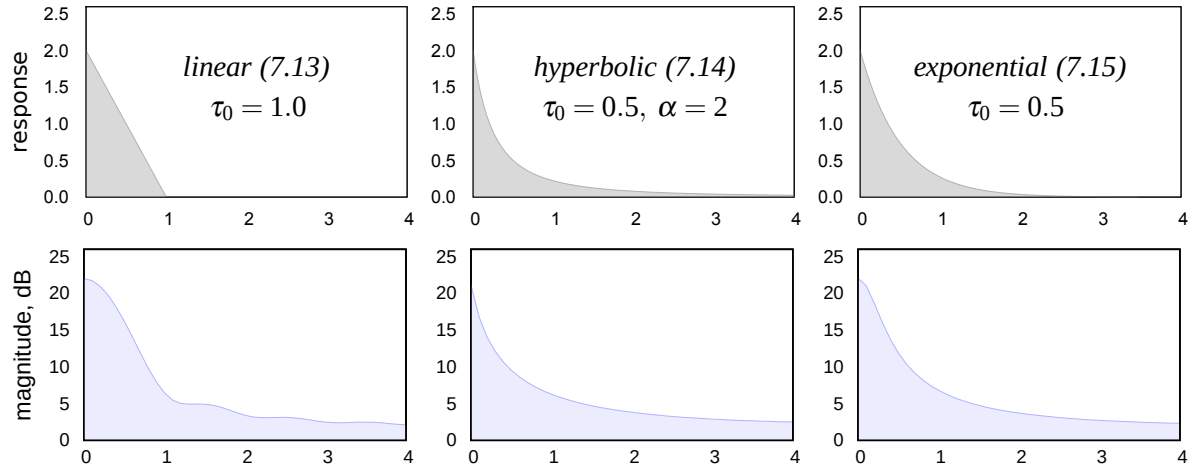


Figure 7.9: Media response functions and their frequency domain response.

Modeling Events Importance

We model events by using a family of functions of event importance $e(t)$, represented in Figure 7.4(d). The most basic shapes of the varying importance are rectangular (Event 2) and triangular (Events 1 and 3), which can also mix creating trapezoid-like shapes (Event 3').

Accordingly, we need to introduce a set of meaningful event parameters, that can succinctly describe these shapes. In this work, we consider *buildup and decay* rates, *longitude* of an event and its *maximum importance* level.

Rectangular model of $e(t)$ is suitable for long-duration events with roughly constant importance and coverage, like "Olympics". This model is represented using a step function with the constant height of n_0 and the longitude τ_e, originating at a time $t = 0$:

$$e(t) = \begin{cases} n_0, & t \leq \tau_e; \\ 0, & t > \tau_e. \end{cases} \tag{7.16}$$

Triangular model adds the parameters of buildup and decay, when $e(t)$ is of varying importance to mass media during its period. Accordingly, we can represent it using a piecewise linear function, which originates at a time $t = 0$, and reverses its direction at a time t_0, resulting in a triangular shape:

$$e(t) = \begin{cases} at, & 0 \leq t \leq t_0; \\ (a+b)t_0 - bt, & t_0 < t \leq (a+b)t_0/b; \\ 0, & t > (a+b)t_0/b. \end{cases} \tag{7.17}$$

We demonstrate the look of the resulting news functions in Figures 7.10 and 7.11. In this example, we used news events of the three different longitudes (τ_e = 0.5, 1 and 2 days) and

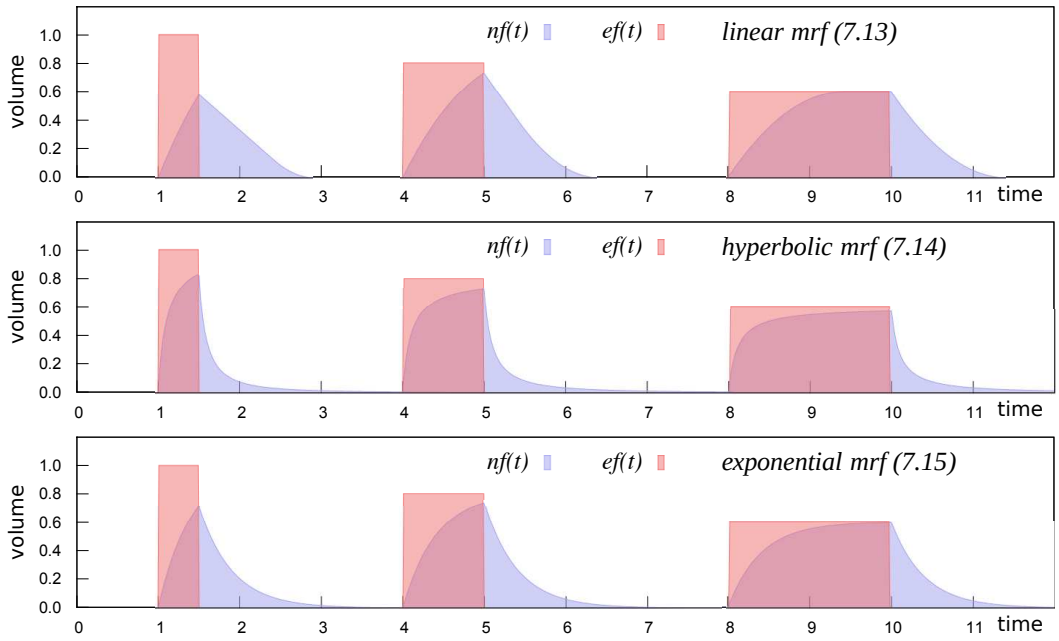


Figure 7.10: Rectangular event importance shapes and their convolution.

selected parameters of response functions so that $n(t)$ would reach the same amplitude after 1 day (as can be seen in the second event). We observe that a varying time length of events in this model results in varying sharpness of news frequency peaks. Correspondingly, short-time events have a shorter period of actuation and long-time events eventually result in the saturation of news media. In all cases, the period of relaxation of news media is mainly characterized by the shape of the corresponding media response function.

According to our model, the height of the event on the event sequence $e(t)$ indicates its importance, while the length describes its longitude. Events can be of a constant importance, like those shown in Figure 7.10, or of varying importance, shown in Figure 7.11. In both cases, the time longitude as well as the maximum importance can be different for different events even for the same topic.

News Deconvolution

Deconvolution is the process opposite to convolution (7.12). After performing this procedure we are able to recreate the original event importance sequence, as shown in Figure 7.4.d, right. In Figure 7.12 we demonstrate the effect of deconvolution on an example time series from [75], where all our three models are applied with the same parameter τ of 0.8 days. Accordingly, the linear response has the smallest power and the hyperbolic response has the largest power for comparable values of τ.

For instance, in the case of exponential response function (7.15), equation (7.12) has the ex-

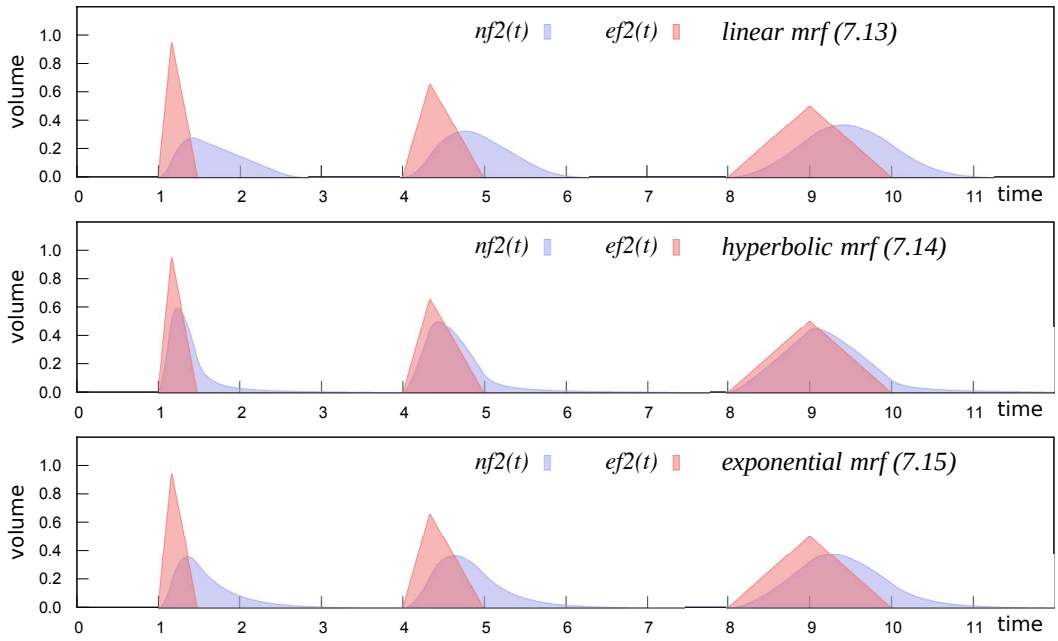


Figure 7.11: Triangular event importance shapes and their convolution.

act analytical solution, providing an equation for deconvolution expressed through the function of volume and its derivative [44]:

$$e(t) = n(t) + \tau_0 \frac{d}{dt} n(t) \tag{7.18}$$

However, this expression is not directly applicable on noisy data, since its second component multiplies the first derivative of the volume function, which is the most heavily affected by noise. This results in the output series $e(t)$ having almost τ_0 times higher noise than the input series $n(t)$. A possible solution to this problem lies in applying the regression of input values before determining their derivative. Another solution is to cut off all the high frequency components of the signal, which, as we show later, can be seamlessly integrated to our framework.

Moreover, the deconvolution can be expressed in the closed functional form only for a limited number of response functions, while our framework is designed to support any kind of finite integrable functions. Therefore, we take an approach to deconvolution, which relies on the frequency domain analysis of the signals, as explained below.

Convolution theorem states, that Fourier transformation of a time-domain convolution of two series is equal to a multiplication of their Fourier transformations in the frequency domain:

$$\mathcal{F}\{n(t)\} = \mathcal{F}\{e(t) * mrf(t)\} = \mathcal{F}\{e(t)\} \cdot \mathcal{F}\{mrf(t)\} \tag{7.19}$$

According to this equation, for every cyclic frequency ω, we can obtain Fourier coefficients

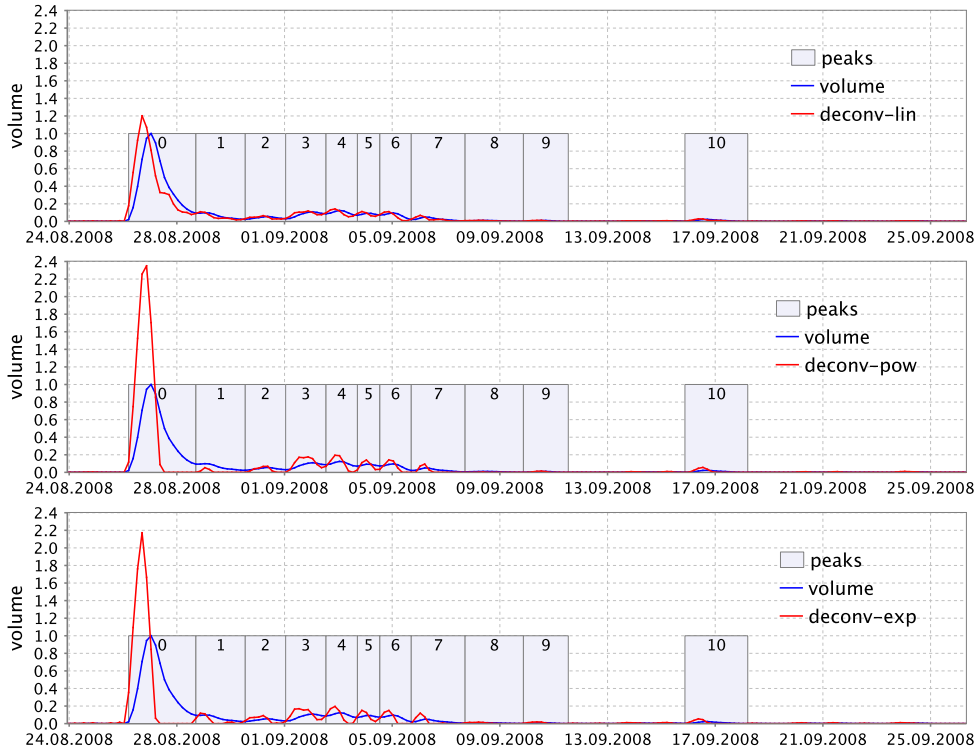


Figure 7.12: Example time series from [75] with $e(t)$ obtained by deconvolution.

$e(\omega)$ of the original series $e(t)$ by dividing Fourier coefficients of the observed series $n(t)$ by the corresponding coefficients of the response function $mrf(t)$. Then, we can obtain a time-domain representation of $e(t)$ by performing an inverse Fourier transformation:

$$e(t) = \mathcal{F}^{-1}\{e(\omega)\} = \mathcal{F}^{-1}\{n(\omega)/mrf(\omega)\} \tag{7.20}$$

Fourier coefficients are expressed as complex numbers where j is imaginary one, $j = \sqrt{-1}$. This component is responsible for a phase shift of signals, and is of a particular importance to our application, since the shifting phase allows to determine the correct timing of an event. Below, we obtain the analytical Fourier representations of the three example response functions considered in our framework.

For Linear Response (7.13) we have an integral from 0 till τ_0:

$$\begin{aligned} mrf(\omega) = \int_0^{\tau_0} (\tfrac{2}{\tau_0} - \tfrac{2t}{\tau_0^2}) e^{-j\omega t} dt &= \tfrac{2}{\tau_0} \int_0^{\tau_0} e^{-j\omega t} dt - \tfrac{2}{\tau_0^2} \int_0^{\tau_0} t e^{-j\omega t} dt = \\ = (\tfrac{2t}{j\omega\tau_0^2} - \tfrac{2}{\omega^2\tau_0^2} - \tfrac{2}{\omega\tau_0}) e^{-j\omega\tau_0}\big|_0^{\tau_0} &= \tfrac{2}{\omega^2\tau_0^2}(1 - e^{-j\omega\tau_0}) + \tfrac{2}{j\omega\tau_0} \end{aligned} \tag{7.21}$$

For Exponential Response (7.15), we integrate from 0 till infinity:

$$mrf(\omega) = \frac{1}{\tau_0}\int_0^\infty e^{-t/\tau_0}e^{-j\omega t}dt \quad = \quad \frac{-1}{1+j\omega\tau_0}e^{-(1/\tau_0+j\omega)t}|_0^\infty = \frac{1}{1+j\omega\tau_0} \tag{7.22}$$

For Hyperbolic Response (7.14), we substitute $t = z/j\omega - \tau_0$, obtaining the integral from $j\omega\tau_0$ to ∞, which can be expressed through incomplete gamma function:

$$\begin{aligned} mrf(\omega) = \frac{(\alpha-1)}{\tau_0}\int_0^\infty (\frac{t+\tau_0}{\tau_0})^{-\alpha}e^{-j\omega t}dt \quad &= \quad \frac{(\alpha-1)}{j\omega\tau_0}\int_{j\omega\tau_0}^\infty (\frac{z}{j\omega\tau_0})^{-\alpha}e^{-z+j\omega\tau_0}dz = \\ = (\alpha-1)(j\omega\tau_0)^{\alpha-1}e^{j\omega\tau_0}\int_{j\omega\tau_0}^\infty z^{-\alpha}e^{-z}dz \quad &= \quad \Gamma(1-\alpha, j\omega\tau_0)(\alpha-1)(j\omega\tau_0)^{\alpha-1}e^{j\omega\tau_0} \end{aligned} \tag{7.23}$$

Parameters Estimation

While $n(t)$ can be directly measured and its derivative $n'(t)$ can be estimated using numeric methods, we still have to find the appropriate time constant τ_0, which is different for different topics. To estimate this constant, we propose exploiting one of the following assumptions on the shape of news event function $e(t)$:
1) $e(t)$ is a spike function, in which case: $e(t) = \sum_{i=1}^k \delta(t-t_i)$;
2) $e(t)$ is a pulse function, in which case: $e(t) = \sum_{i=1}^k h(t-t_i')h(t_i''-t)$;
In the first case, t_i are the times of events, and $\delta(t)$ is Dirac's delta function. In the second case, t_i' and t_i'' indicate beginnings and endings of events, and $h(t)$ is the Heaviside function. We note that in both cases, the space in between events is empty. If t_i'' is the ending of the previous event and t_{i+1}' is the beginning of the next one, we have:

$$\int_{t_i''}^{t_{i+1}'} e(t)dt \approx 0 \tag{7.24}$$

Since $e(t)$ in the first case is most of the time equal to zero, we can write the following equation for estimating τ_0 using linear regression:

$$n(t) + \tau_0 \cdot n'(t) = 0;\ \tau_0 = -\frac{\mathrm{Cov}[n'(t), n(t)]}{\mathrm{Var}[n'(t)]}, \tag{7.25}$$

where Cov[] is the covariance and Var[] is the variance. Although this estimate can be biased to the higher side by $e(t)$, it is computationally efficient and stable with regard to noise. In the second case, the impact of $e(t)$ can no longer be discarded, and thus other methods are needed. For instance, we can apply an optimization method, which estimates τ_0 by minimizing the area under the curve or, equivalently, maximizes τ subject of constraints:

$$\tau_0 = \arg\min \tau \left[\int_{t_a}^{t_b} |n(t) + \tau\frac{d}{dt}n(t)|dt\right] = \arg\max \tau \left[|_{t_a}^{t_b} n(t) + \tau\frac{d}{dt}n(t) \geq 0\right] \tag{7.26}$$

However, the above methods are only applicable to exponential response. A more general method is to estimate the parameters of response functions directly from decay shapes. This can be done by regression parameter fitting, following the peak detection and the extraction of the descending slope. To estimate the parameters, we first normalize the descending slopes of time series and then analyze them using either linear, power-law or exponential regression. Next, we take the average of the extracted parameters across few larger peaks, weighting them according to the residual errors of fitting.

7.4.3 Annotating Events

We work with time series of social response to events coming from different sources, not necessarily the news agencies. If the news media coverage for some events is sufficient, we can detect events merely by looking at news publication dynamics. However, the attention of news media to some events or topics can be insufficient, but we still want to recognize them reliably if these events generate a significant social response, detected by our methods. Consider for example of an event being the price announcement for a very anticipated device. These events are particularly rarely covered by news media, yet they cause hot discussions in relevant communities and apparent sentiment shifts. In such cases, we want to navigate to the news time series, no matter how scarce they are, and extract event annotations. Thus, we need special methods which can take into account time framing of events.

Assigning News Articles to Events. After extracting news events time series, we should distinguish between subsequent and duplicate events, and be able to map each news article to the correspondent event. In the following, we propose to use a probabilistic framework which models the news sequence and allows mapping between events and news articles.

We assume the principle of locality and independence of news events, according to which the occurrence of each event is independent on all the previous events and is determined only by the average rate λ and a time t passed from the last event. This process is described by a Poisson probability:

$$P[event] = \lambda t \cdot e^{-\lambda t} \tag{7.27}$$

The value of λ can be estimated using the auto-correlation of news time series. Thus, we can use the above formula to merge the subsequent events according to their probability of occurrence right after the main (first) event. The same formula can also be applied to news articles publication probability, in order to map news articles to events. After obtaining the collection of relevant articles, we can employ linguistic or statistical methods to extract the text of the event, as described in the following section.

The subsequent news aggregation and their text analysis also appear to be challenging problems, requesting specific methods, which fall out of the scope of this work. Therefore, in the

rest of this section we only briefly consider some of their issues, important for our methods, pointing the interested reader to the relevant literature.

Let us remind the problem: given a time interval and a topic, identify what has happened in the news. First of all, we assume that most of the news events contain new information. This means that if there are news present in a time interval - they are likely to contain the cause of a sentiment shift, and if they are not - sentiment shift is either externally unconditioned (endogenous) or is a temporal fluctuation, and both of these cases are falling out of the main scope of our work.

Extracting Event Annotation. During the interval there can be more than a single news publication about the same event, and these publications may also have different phrasing or expressions. Nevertheless, we want to detect the essence of an event reported in the news. We propose to compare the statistics of the news articles of an interest (falling into a specific time interval) to the same statistics calculated over the entire collection of news (same topic, but for all intervals). This can be done using unsupervised clustering (by comparing two cluster centroids, then finding their difference), or by comparing arrays of TF-IDF scores, since new keywords should leave a distinct footprint in frequency. In this case, when in a time interval there are several news articles from different authors, we can aggregate them before analyzing, in order to remove individual linguistic differences. This opens a possibility of an unsupervised extraction of news events, as described below.

We model the event annotation $A[event]$ as a set of relevant terms (keywords) $A_e = \{T_j\}$. Accordingly, we want to extract those terms, which became more popular in the current collection of documents for a time interval p and topic T, $\mathcal{D}_p^T$, compared to the past collection of documents, $\mathcal{D}^T$:

$$A[event] = \{T_j \mid \textit{TF-IDF}(T_j, \mathcal{D}_p^T) - \textit{TF-IDF}(T_j, \mathcal{D}^T) > \rho\} \tag{7.28}$$

where ρ is the relevancy threshold, and *TF-IDF* is measured as the weighted term frequency:

$$\textit{TF-IDF}(T, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{D_i \in \mathcal{D}} \textit{TF-IDF}(T, D_i) \tag{7.29}$$

The above method relies on statistical analysis. However, in some cases there is only a single news article present. In this article, the author may use different expressions, resulting in increased probabilities of irrelevant words. If author uses specific terms, which are different from the general corpus (for instance, stock market company names, indices), they can be erroneously considered as keywords for an event. In this case, there is a necessity to apply more sophisticated natural language text analysis. For instance, this can be done by parsing article's sentences and extracting relevant phrases (according to a database of patterns), that contain the description of an event. Nevertheless, these methods fall out of the scope of our work.

7.5 Experimental Evaluation

The main goal of our experimental evaluation is to study properties of the real data and evaluate the proposed and existing models and their assumptions. We begin with the evaluation of our method's parameters estimation principle, since all our subsequent experiments are automated and rely on this phase. Following that, we want to analyze and compare the differences in dynamics of social media, as well as the applicability of the proposed response functions.

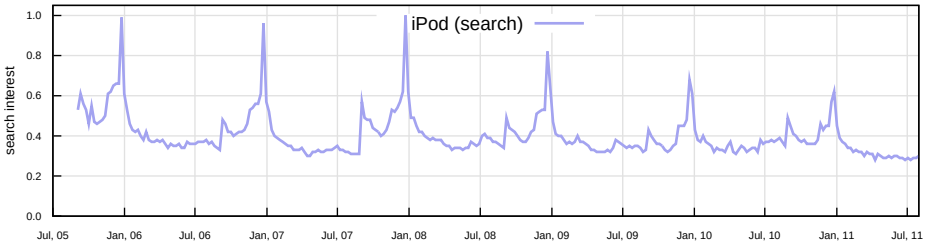


Figure 7.13: The search interest for the topic "iPod", featuring exponential decay.

Evaluation of Parameter Estimation

To demonstrate regression parameter fitting, we use a time series of search interest for the topic "iPod" featuring six events, coinciding with Christmas sales (Figure 7.13). The outbursts of search interest corresponding to these events have distinguishably exponential shapes, and we take them as input data for estimating parameters τ_0 (as shown in Figure 7.14). To estimate the parameters, the descending slopes of time series were normalized and analyzed using exponential regression. Reported on figures are the inverse τ_0 parameters measured in weeks and their corresponding squared normalized errors.

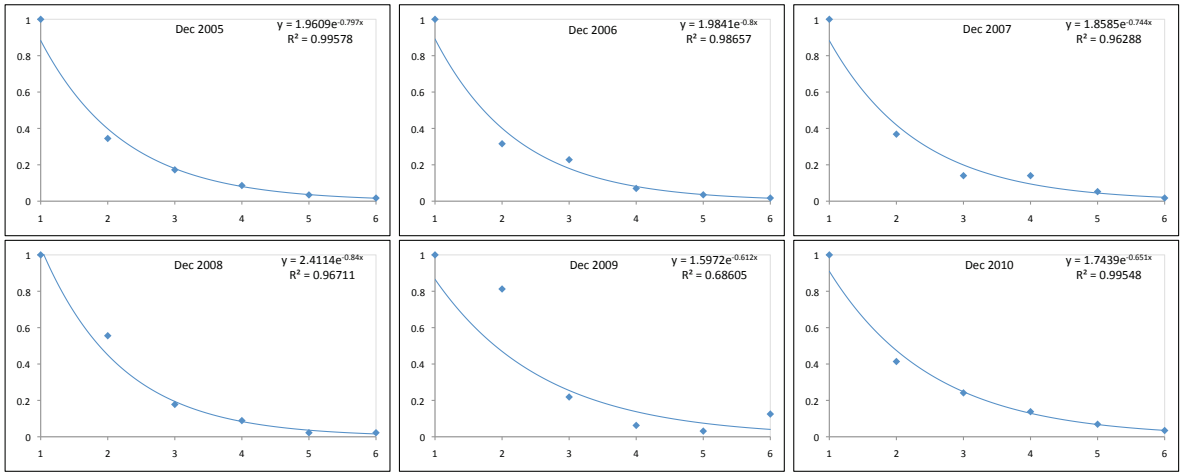


Figure 7.14: Decay parameters estimation using exponential regression.

Series	y_0	$1/\tau$	τ	R^2
Dec 2005	1.96	0.797	1.25	0.996
Dec 2006	1.98	0.800	1.25	0.987
Dec 2007	1.86	0.744	1.34	0.963
Dec 2008	2.41	0.840	1.19	0.967
Dec 2009	1.60	0.612	1.63	0.686
Dec 2010	1.74	0.651	1.54	0.995
Average	1.93	0.741	1.37	0.932
(weighted)	(1.94)	(0.747)	(1.35)	0.932
Variance	0.28	0.09	0.18	0.12
(weighted)	(0.06)	(0.01)	(0.02)	0.12

Table 7.2: Estimated decay parameters.

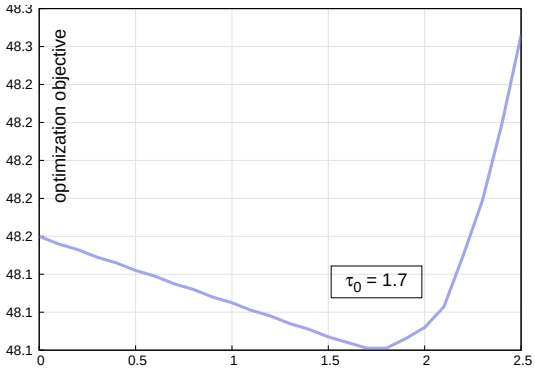


Figure 7.15: Optimized decay parameters.

The results of our evaluation are summarized in Table 7.2. The most interesting outcome is that the parameters of social response dynamics remain the same for the course of five years, despite the decreasing interest of users. Also we observe that our weighting technique results in a smaller variance of the parameters and a closer approximation of their real values in the presence of noise, by discounting the weight of outlier results, as can be seen in the case of "Dec 2009", which had the deviating value of τ and in the same time the worst fitting quality.

Next, we compare the results of our parameter fitting to those of the proposed optimization method, shown in Figure 7.15. We see that the optimal value of τ detected by this method is larger than the one determined by a direct regression, coming in a full agreement with our theory. Nevertheless, this method has a stable behavior, allowing this bias to be corrected.

Evaluation of Meme Model

Taking into account our findings in Section 7.4.2, we used dynamics equation (7.1) to predict values of every peak through applying it on the volume and time accumulated since peak's beginning. When considering the linear form of the imitation factor expressed as $f(x) = a + bx$, and $\delta(t) = t^{-\alpha}$ ($\alpha > 1$), we reach the following model:

$$\begin{aligned} \frac{dx}{a+bx} &= ct^{-\alpha}dt \\ \frac{1}{b}\, ln\,|a+bx| &= \frac{c}{1-\alpha}t^{1-\alpha}+c \\ f(x) = a+bx &\sim exp(\frac{bc}{1-\alpha}t^{1-\alpha}) \\ n(t) = \frac{dx}{dt} &\sim exp(\frac{bc}{1-\alpha}t^{1-\alpha})t^{-\alpha} \end{aligned} \tag{7.30}$$

In Figure 7.16 (top) we demonstrate an example prediction of this model in the case when $\alpha = 3.3$ and $bc = 3.3$. We observe that the first peak's buildup and decay can be matched using these parameters only approximately, while the shapes of all the subsequent peaks can not be matched at all.

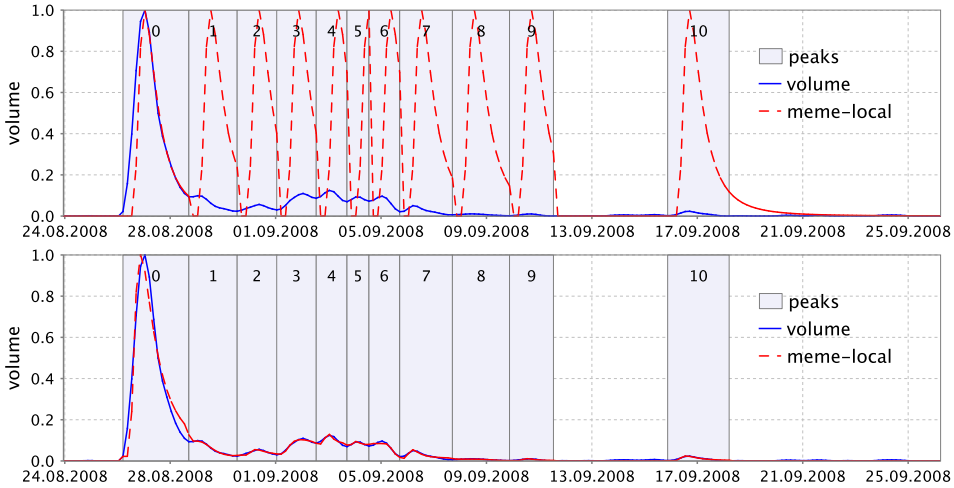


Figure 7.16: Volume, predicted by meme-local model from [75], using fixed (top) or individually fitted (bottom) parameters.

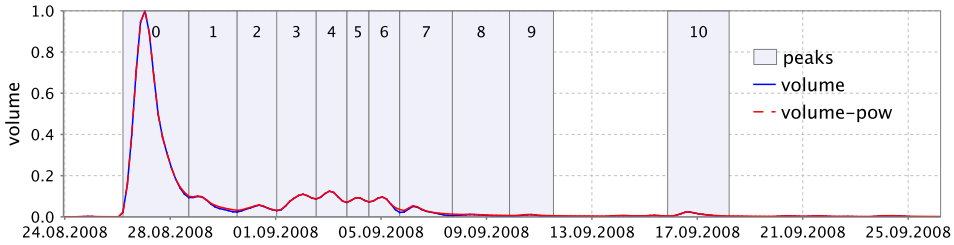


Figure 7.17: Example time series from [75] and its approximation from the event model using a hyperbolic deconvolution with automatically extracted parameters.

Fitting (7.1) to individual peaks using *least squares* regression yields much better accuracy for small peaks (Figure 7.16, bottom), but does not match sharpest peaks, where buildups and decays seem to require different model's parameters. In the above experiment, fitting employed four parameters, which were independent among peaks and appeared different for every peak. Two of these parameters are effectively stretching the model to accommodate real data: base level (a) indicates the volume accumulated before the peak, and scale (b) serves the purpose of fitting the peak's height, to counterweight the normalization effect caused by dividing by volume in (7.1).

In contrast, our deconvolution model uses a single set of parameters for the entire time series, which are automatically estimated by performing a peak slope regression (using a single or multiple peaks). Figure 7.17 demonstrates the output of a hyperbolic deconvolution model, which fits the time series consistently across all peaks. The output in this case is the time series, constructed by performing an inverse process of convolution over the deconvolution-estimated time series of event importance.

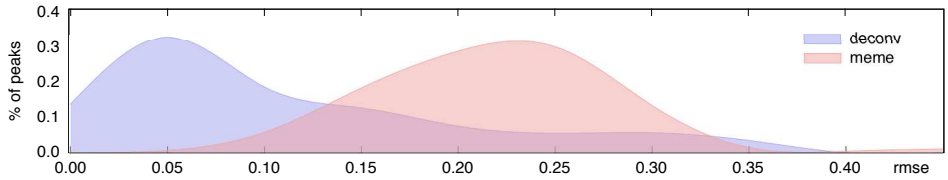


Figure 7.18: Error distribution for meme and deconvolution models (hyperbolic).

We computed residual errors of peak fitting for meme and deconvolution models using approximately 500 peaks automatically extracted from top 100 time series. Errors were measured as RMSE, and then normalized according to every peak's height, so that the results can be averaged and compared across peaks. First, we compare the observed error distributions between the meme and hyperbolic deconvolution models in Figure 7.18. A hyperbolic deconvolution model was chosen as the closest match to a meme model regarding the predicted shapes of peak slopes (refer to Equation 7.30). Moreover, this model was based on triangular $e(t)$ shapes, automatically extracted using a deconvolution with the parameters $\tau = 0.8$ days and $\alpha = 2.0$, fixed for all time series. Deconvolution model reached the average RMSE of 0.11 with the standard deviation of 0.09. Meme model demonstrated the average RMSE of 0.22 with the standard deviation of 0.09. Taking into account the artifacts of data aggregation, noise and deviation of peak starting times, the error level of 0.2 can still be considered as acceptable for meme model. Nevertheless, it fitted more than a half of evaluated peaks with larger errors than our model, which used a single and arbitrarily chosen set of parameters for *all* time series.

We note that choosing smaller decay times will result in smaller differences between the event importance and the volume. On the other hand, this will introduce more errors to the estimation of triangular $e(t)$ shapes (currently performed by a linear regression) and therefore result in more errors for the output approximation. Choosing larger decay times, can also result in approximation errors due to omission of smaller peaks. Therefore, it makes sense to perform deconvolution using the estimated parameters, especially since our model is very flexible with regard to parameter errors, as demonstrated by Figure 7.18.

Evaluation of Deconvolution Model

In our previous experiments we used a fixed set of parameters for the proposed deconvolution models, demonstrating that they are able to fit data quite well even without any prior adaptation. We can guarantee that the deconvolution with smaller decay parameters is also fail-safe, as long as event shapes are not approximated. However, only a more powerful deconvolution helps us to recognize the actual event importance, and represent it using *linear* dynamics. In this case, wrongly chosen parameters (to the higher side) may result in smaller events becoming outcast by a "shade" from their preceding larger neighbors. Therefore, we need to apply the deconvolution using the largest possible but still correct parameters.

To compare the accuracy of our models for different data, we automatically extracted the deconvolution parameters using peak slope regression over the identified peaks. Then, we averaged these parameters for every time series and applied deconvolution using a single parameter set. Following this, we approximated event importance using piecewise linear regression, obtaining the time series used in our analysis of events. Nevertheless, to evaluate the correctness of our modeling of the media response and event importance, we need to quantify the accuracy of fitting. A direct measure of fitness can be computed by averaging residual errors of piecewise linear regression of events importance. However, this measure only improves with the more powerful deconvolution, as more and more events obtain spike shapes. We therefore propose to evaluate accuracy by doing an inverse process of convolution over the estimated model, and comparing the output of this model with the original. Accordingly, errors in fitness are measured as RMSE, and then normalized for every peak's height, so that the results can be averaged and compared across peaks. Below, we compare our models on several datasets with different characteristics.

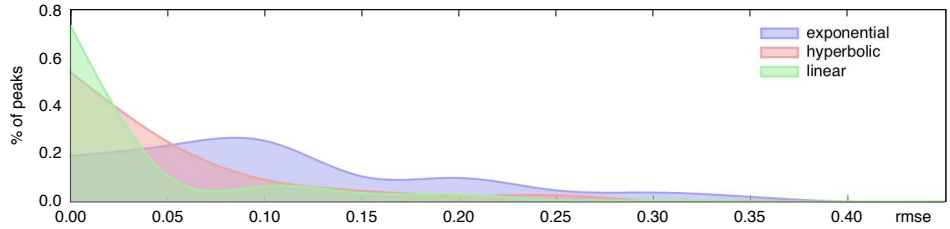


Figure 7.19: Error distribution of deconvolution models for Meme dataset.

Meme Dataset The performance of our methods on meme dataset is demonstrated in Figure 7.19. We observe that the linear and hyperbolic deconvolution models reached the best accuracies with the average errors of 0.03 and 0.05 respectively, and the standard deviations of 0.06 (thus, the the difference in performance is not significant). However, while the linear model had the average parameter τ of 0.5 (very small), the average parameters τ and α of the hyperbolic model were 0.36 and 2.8 respectively. A more careful evaluation of the estimated parameters reveals that the hyperbolic model has almost the same parameter τ of 0.36 ± 0.03, but quite different α of 2.8 ± 0.96. A similar pattern is observed for triangular model, where a single decay parameter τ has values in the range 0.5 ± 0.3. Exponential model demonstrated the average RMSE of 0.11 with the standard deviation of 0.09, almost the same to our previous experiment with fixed parameters, but in this case the estimated parameter τ was on average equal to 0.65, thus only approximating a usually much steeper hyperbolic response. From these observations we can conclude that meme data is more likely to have a hyperbolic response pattern. Overall, we can conclude that while the dominance of particular response functions is clearly visible for the meme domain, their parameters are significantly different.

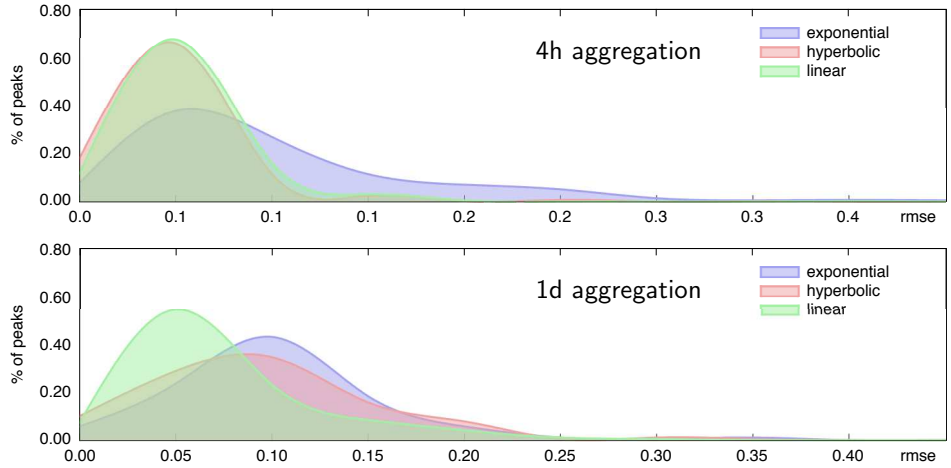


Figure 7.20: Error distribution of deconvolution models for Twitter dataset.

Twitter Dataset has very different properties when compared to the output of blog or news media. First, this platform has a distinct bias towards current events and temporal activity of users. So usually there are no global trends, since any activity fades out to zero after some time. Second, there are different types of dynamics present at the same time: *daily activity* and *overall activity*. Whereas the first one is largely driven by work schedules in different time zones, the second one demonstrates a more clear pattern of event interest. To demonstrate this difference, we analyzed 30 time series from Twitter using either $4h$ or $1d$ aggregations. The results of our evaluation are presented in Figure 7.20.

For the $4h$ aggregation we observe that the linear and hyperbolic deconvolution models have the same accuracies, and both have much better performance than the exponential model. Finding the differences between the hyperbolic and linear model in this case is like splitting hairs, since their only distinction can be seen in the tail dynamics, which is masked by other peaks. Nevertheless, for this kind of activity we would like to give the edge to the linear model, as it better describes the linear increase and decay effects that shifting time zones produce for events. Our experiments with the same data using $1d$ aggregation also demonstrate a good performance for the linear model, though the average parameter τ was 2.5 with the deviation of 0.9, indicating that a rather high fraction of peaks had small τ. Such results only prove that most events have a nearly linear importance around their peaks. Since in our dataset we usually have a small number of samples per peak (8-10), it is not possible to verify the hypothesis of the linear model. However, the two other models (hyperbolic and exponential) have the estimated decay parameters more powerful than the linear model ($\alpha = 2.4 \pm 0.5$, $\tau_h = 1.3 \pm 0.6$ and $\tau_e = 1.5 \pm 0.7$). Thus, their performance shows the real accuracy of volume approximation from the event importance. Finally, we observe that both models encountered large variations of the error, probably indicating the existence of *different* kinds of response dynamics for different topics or different response parameters for various events during the same time series.

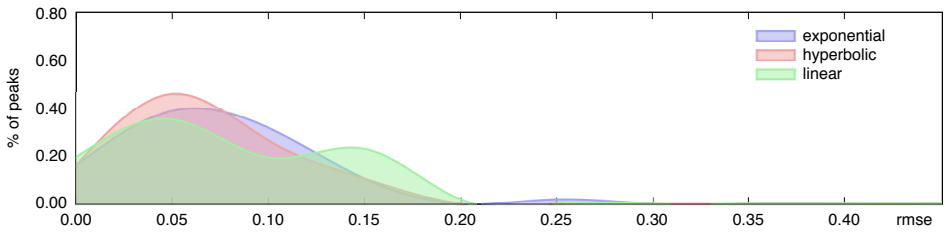


Figure 7.21: Error distribution of deconvolution models for Google dataset.

Google Dataset. To further verify the differences in dynamics for different media, we extracted time series of search interest from Google. The extracted dataset contained the same topics as in our Twitter evaluation, but with a much longer time span from Jan 2009 to Jan 2013 and with a higher aggregation granularity of 1 week. Using this level of aggregation, we eliminate daily and weekly variations in user activity, dealing with cleaner but more global trends of user interest. Therefore, we did not apply regression smoothing for this data, since values do not have small-scale temporal variation. Our goal is to observe if the reaction and global trends of web search have significantly different properties of dynamics, compared to short-lifecycle media, like Twitter or news agencies.

Reported in Figure 7.21 are error distributions for our models. While exponential and hyperbolic models fared particularly well for these data, the linear model was suitable only for a half of time series, as can be seen from the second bump in error distribution. Again, this confirms our previous observation that topics can in fact affect the dynamics of the media. To understand why, let's refer to the particular time series we analyzed, demonstrated in Figure 7.22. We see that an impacting death of Michael Jackson resulted in distinctly hyperbolic dynamics, with the response lasting much longer compared to exponential dynamics of interest for Harry Potter movie premieres. From our evaluation, we can conclude that while different media have preferences for particular response dynamics, it is more often determined by event types.

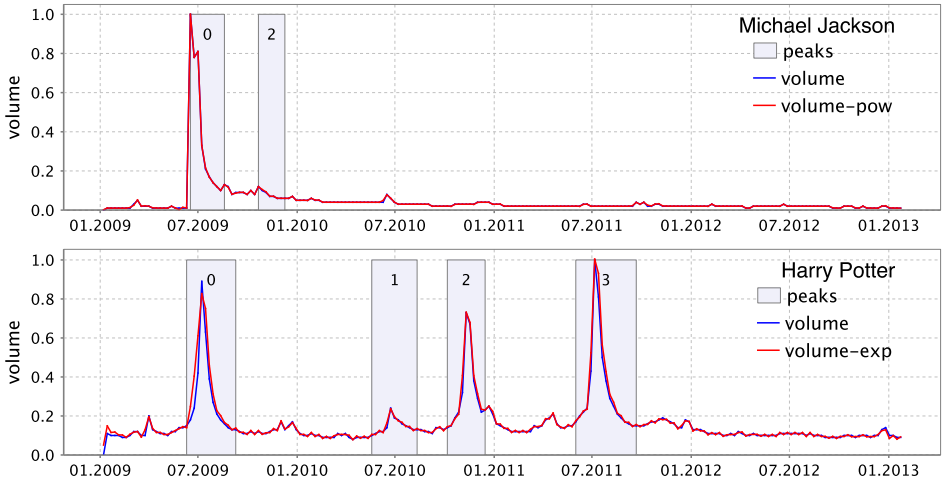


Figure 7.22: Search interest time series from Google featuring different dynamics.

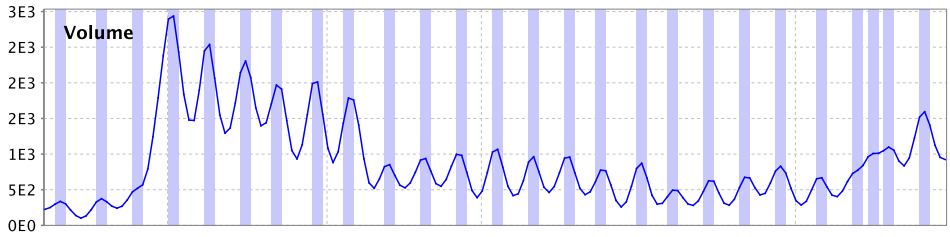


Figure 7.23: Bursts of $n(t)$ extracted using deconvolution, $\xi = 1d$.

Correlation Statistics. In the above sections we discussed the performance of deconvolution models and evaluated the quality of automatic parameter estimation. Now we begin our evaluation of sentiment shifts and events correlation, where events are extracted with the help of deconvolution, and sentiment shifts are determined using functions of aggregated sentiment proposed in Section 4.2 and Section 5.3. In particular, we use the measures of positive and negative sentiment volume $s^+(t)$ and $s^-(t)$ (normalized) and sentiment contradiction C. We determine bursts of sentiments and events according to a threshold based on the running average.

We note that while we can determine bursts of sentiments using simple methods, events require more robust methods. Let's consider an example of sentiment and event burst extraction demonstrated in Figure 7.23. One can see that two of the three events in the right part of the time series are located on the monotonic slope of $n(t)$, and thus are not detectable either with thresholding or with derivative tracking. Remarkably, deconvolution can recover these and other "hidden" events and result in more accurate correlation. Moreover, deconvolution helps identifying correct event timings. It generally shifts peaks backwards in time to the extent determined by a response function and event dynamics. This, in turn, helps to identify correct time lags between events and sentiment shifts.

The results of our evaluation are presented in Table 7.3. We extracted a set of 30 topic time series from Twitter, for the period of half a year from June 2009 till December 2009. A subset of 23 of these time series contained more than 2 real events, which were useful for measuring correlation. For each of these series, we measured the correlation between the tweets volume, and the three sentiment measures - positive and negative sentiments, and sentiment contradiction. For each of these, we present the values of Cosine similarity ρ_C, Jackard coefficient ρ_J and the time lag, resulting in the best correlation. Our first observation is that ρ_J is always smaller than ρ_C, since for highly-overlapping sets of bursts the harmonic mean of set sizes is smaller than the size of a joint set. Also we observe that positive sentiments demonstrate the highest overall correlation, followed by contradictions and negative sentiments. Taking a closer look at the data, we see that positive sentiments are usually preceding the unexpected events (having correlations with a negative lag), whereas negative sentiments are observed after or during these events (positive lag). For instance, such behavior is observed in our table for events surrounding topics "Iran Election", "Iraq" and "Fort Hood", where a (relatively) positive sentiment level is

a part of background level, not associated with any particular events. Anticipated events like "LCROSS", "Harry Potter" or "Leica" also have positive expectations, but in this time connected to events. Another group of anticipated events, like "Nobel Prize", "Beer Summit" and "Transformers" features negative expectations and positive outcomes, while heavily promoted events like "Leica" or "CERN LHC" have a mixture of controversial sentiments during their peaking intervals.

We found that types of discovered correlations and values of time lags correspond to topics of time series. Evidently, different types of events can happen during the course of time series, all having different impacts on sentiment and time lags. Therefore, it makes sense to organize the correlation experiment not per time series, but per events. First, we need to select events of particular kind and determine all the relevant sentiment shifts (of different kinds) in some proximity of each event. Second, we need to compute correlation between events and sentiment shifts of the same kind, determine their optimal time lag, and determine which sentiment measures are best correlated with which kinds of events. Finally, in order to analyze event causality, we need to build event profiles with the help of machine learning.

Topic Name	**PosVolume**			**NegVolume**			**Contradiction**		
Measures	ρ_J	ρ_C	lag	ρ_J	ρ_C	lag	ρ_J	ρ_C	lag
Hangover	0.59	0.75	0.0	0.37	0.54	2.0	0.45	0.62	2.0
LCROSS	0.38	0.55	-1.2	0.36	0.53	0.6	0.43	0.60	0.6
CERN LHC	0.48	0.65	0.0	0.33	0.50	0.0	0.39	0.56	0.0
Ice Age	0.48	0.65	0.0	0.33	0.50	0.0	0.39	0.56	0.0
Michael Jackson	0.35	0.52	0.0	0.32	0.48	-2.3	0.35	0.52	1.8
Swine Flu	0.40	0.57	2.0	0.38	0.55	-2.0	0.33	0.50	0.6
Barack Obama	0.33	0.50	0.0	0.35	0.52	0.0	0.35	0.52	1.9
Harry Potter	0.44	0.61	-1.4	0.28	0.44	0.7	0.40	0.57	-1.6
Neda	0.43	0.60	-2.0	0.36	0.53	-1.9	0.41	0.59	-1.1
Iran Election	0.41	0.58	-2.3	0.34	0.51	2.0	0.42	0.60	-1.7
Iraq	0.36	0.52	-1.9	0.44	0.61	0.6	0.44	0.61	1.6
Fort Hood	0.62	0.77	-1.3	0.39	0.56	1.9	0.37	0.54	-1.4
Follow Friday	0.62	0.77	-1.3	0.39	0.56	1.9	0.37	0.54	-1.4
Google Wave	0.50	0.67	1.7	0.31	0.48	-0.8	0.57	0.74	-2.3
NASA	0.56	0.71	0.7	0.60	0.76	-0.9	0.55	0.72	2.5
Super Bowl	0.51	0.68	0.7	0.38	0.55	0.0	0.44	0.61	0.5
Nobel Prize	0.42	0.62	1.5	0.33	0.51	-1.3	0.33	0.53	-0.5
Beer Summit	0.29	0.45	0.6	0.51	0.68	-1.5	0.60	0.75	-1.5
Transformers	0.39	0.56	0.8	0.39	0.56	-2.4	0.35	0.52	-2.4
Facebook	0.30	0.47	-1.3	0.38	0.55	0.0	0.31	0.47	0.0
Leica	0.44	0.61	-1.6	0.42	0.59	0.0	0.44	0.62	-0.5
Gmail	0.43	0.60	0.0	0.31	0.47	-2.2	0.33	0.50	-2.2
TwitterPeek	0.36	0.60	0.0	0.42	0.60	-2.0	0.25	0.45	0.0

Table 7.3: Sentiment correlation statistics for selected time series from Twitter.

7.6 Conclusion

Our evaluation of news dynamics and their impact on sentiments is the first systematic work in this direction, which applies a thorough and universal modeling of news distribution in various media and studies news interaction with sentiments. We develop a unique model of news event dynamics, which allows to capture meaningful and important characteristics of news event. Our model can accommodate various response functions, suitable for different cases, which should not necessarily be expressed as a differential equation, but can be learned from the data.

The results obtained by applying our methods to several datasets confirm their robustness and universality. Nevertheless, they also reveal that we need to address several more keystone challenges on our way towards the final solution. First, we observe the existence of different kinds of response dynamics for topics in the same media, and even different response functions and their parameters for various events during the same time series. We observe that while different media have preferences for particular response dynamics, it is more often determined by event types. Thus, we need to develop methods of news deconvolution which will automatically determine the best model for every particular event and process the corresponding time interval accordingly. Above all, this involves a refinement of our model of events importance and development of robust and precise deconvolution optimization strategy.

Our analysis shows that different types of events may have different impacts on sentiment and varying time lags. Therefore, not only we need to tailor our correlation methods for particular measurements, but also make them recognizing kinds of events. We then need to analyze causality between discovered sentiment-event correlation pairs and learn event profiles. Finally, we want to build a classifier, predicting changes in sentiment based on event's features, and on the past history of correlation.

Chapter 8

Conclusions and Future Work

We set the objective of our research to aid large scale sentiment analysis and opinion mining by providing efficient and scalable sentiment aggregation and storage methods. Grounding on these methods, we develop an efficient framework for detecting, explaining and predicting sentiment contradictions.

Our aggregated sentiment storage enables approaches to scale to very large data collections and to answer relevant queries in real time. It is incrementally maintained in an online environment, and can outperform a relational database implementation. Moreover, it is uniquely designed to withstand noise and irregularity of online sentiment, thanks to regression analysis and hierarchical thresholding.

The approach of sentiment contradiction detection presented in this work is notable for using only basic sentiment statistics to capture the level of contradiction. We conducted an experimental evaluation with synthetic data, as well as three diverse real-world datasets and evaluated the usefulness of our approach with a user-study. The results demonstrate the applicability, usefulness, and efficiency of the proposed solution.

Following the contradiction detection, we approach the novel problems of characterizing sentiment evolution in a demographic group and identifying correlated demographic groups. We demonstrate that these problems can be solved effectively on a large scale using clever pruning, top-k and compression methods. The proposed approach allows observing sentiment behavior at a much finer level of detail than currently possible, helping to identify cases that are counter-intuitive and can only be observed by processing large amounts of data. For instance, our evaluation identified interesting correlations among real demographic groups in MovieLens, which can be of particular interest to social scientists and social recommender applications.

We outline some interesting problems and extensions of the presented framework, which we plan to work on. We are currently working on constructing a dataset annotated with contradictions, which will permit us to perform more comprehensive experiments, and also serve as a benchmark for other studies in the area.

For demographics analysis, we plan to extend our framework to be able to prune demographics groups based on their conditional dependency. Automatic filtering of high correlations between such groups is possible even without comparing their user bases, since our storage allows analyzing volumes as well as distributions of sentiments for groups.

Our news sentiment analysis reveals the differences in dynamics among various media and a possibility to reconstruct event importance using deconvolution framework. We also identified correlations between news events and changes in sentiment of various kinds. The next steps in this direction lead to a construction of annotated dataset for sentiment and event causality, which will make possible evaluation of different event dynamics for predicting sentiment shifts.

Bibliography

[1] Aggarwal CC, Han J, Wang J, Yu PS (2003) A framework for clustering evolving data streams. In: Proceedings of the 29th international conference on Very large data bases, VLDB '03, pp 81–92, URL http://www.vldb.org/conf/2003/papers/S04P02.pdf

[2] Alm C, Roth D, Sproat R (2005) Emotions from text: machine learning for text-based emotion prediction. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, ACL, Morristown, NJ, USA, HLT '05, pp 579–586

[3] Annett M, Kondrak G (2008) A comparison of sentiment analysis techniques: Polarizing movie blogs. In: Proceedings of the Canadian Society for computational studies of intelligence, 21st conference on Advances in artificial intelligence, Canadian AI '08, pp 25–35, DOI http://dx.doi.org/10.1007/978-3-540-68825-9_3

[4] Antweiler W, Frank MZ (2004) Is all that talk just noise? the information content of internet stock message boards. Journal of Finance 59(3):1259–1294, DOI 10.1111/j.1540-6261.2004.00662.x

[5] Archak N, Ghose A, Ipeirotis PG (2007) Show me the money!: deriving the pricing power of product features by mining consumer reviews. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, KDD '07, pp 56–65

[6] Asur S, Huberman BA, Szabó G, Wang C (2011) Trends in social media: Persistence and decay. CoRR abs/1102.1402, URL http://arxiv.org/abs/1102.1402

[7] Bermingham A, Smeaton AF (2010) Classifying sentiment in microblogs: is brevity an advantage? In: CIKM, pp 1833–1836, DOI http://doi.acm.org/10.1145/1871437.1871741

[8] Bestgen Y (2008) Building affective lexicons from specific corpora for automatic sentiment analysis. In: Proceedings of the 6th International Conference on Language Resources and Evaluation, European Language Resources Association (ELRA), Marrakech, Morocco, LREC '08

[9] Bifet A, Frank E (2010) Sentiment knowledge discovery in Twitter streaming data. In: Proceedings of the 13th International Conference on Discovery Science, Springer, Canberra, Australia, pp 1–15

[10] Bíró I, Szabó J, Benczúr AA (2008) Latent dirichlet allocation in web spam filtering. In: Proceedings of the 4th international workshop on Adversarial information retrieval on the web, ACM, New York, NY, USA, AIRWeb '08, pp 29–32, DOI http://doi.acm.org/10.1145/1451983.1451991

[11] Blei DM, Ng AY, Jordan MI, Lafferty J (2003) Latent dirichlet allocation. The Journal of Machine Learning Research 3:993–1022, DOI http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993

[12] Bodendorf F, Kaiser C (2009) Detecting opinion leaders and trends in online social networks. In: Proceedings of the 2nd ACM workshop on social web search and mining, ACM, SWSM '09, pp 65–68

[13] Bollen J, Mao H, Zeng XJ (2010) Twitter mood predicts the stock market. CoRR abs/1010.3003, URL http://arxiv.org/abs/1010.3003

[14] Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. Computer Networks 30(1-7):107–117, DOI http://dx.doi.org/10.1016/S0169-7552(98)00110-X

[15] Carenini G, Ng RT, Zwart E (2005) Extracting knowledge from evaluative text. In: Proceedings of the 3rd international conference on Knowledge capture, ACM, New York, NY, USA, K-CAP '05, pp 11–18, DOI http://doi.acm.org/10.1145/1088622.1088626

[16] Carenini G, Ng R, Pauls A (2006) Multi-document summarization of evaluative text. In: Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics, pp 3–7

[17] Castellanos M, Dayal U, Hsu M, Ghosh R, Dekhil M, Lu Y, Zhang L, Schreiman M (2011) LCI: a social channel analysis platform for live customer intelligence. In: SIGMOD, pp 1049–1058, DOI http://doi.acm.org/10.1145/1989323.1989436

[18] Chaovalit P, Zhou L (2005) Movie review mining: a comparison between supervised and unsupervised classification approaches. In: HICSS, DOI http://doi.ieeecomputersociety.org/10.1109/HICSS.2005.445

[19] Chen C, Ibekwe-SanJuan F, SanJuan E, Weaver C (2006) Visual analysis of conflicting opinions. In: IEEE Symposium on Visual Analytics Science and Technology, pp 59–66

[20] Chen F, Tan PN, Jain AK (2009) A co-classification framework for detecting web spam and spammers in social media web sites. In: CIKM, pp 1807–1810, DOI http://doi.acm.org/10.1145/1645953.1646235

[21] Chevalier JA, Mayzlin D (2006) The effect of word of mouth on sales: Online book reviews. Journal of Marketing Research 43(3):345–354, DOI http://dx.doi.org/10.1509/jmkr.43.3.345

[22] Chi Y, Tseng BL, Tatemura J (2006) Eigen-trend: trend analysis in the blogosphere based on singular value decompositions. In: CIKM, pp 68–77, DOI http://doi.acm.org/10.1145/1183614.1183628

[23] Choi Y, Kim Y, Myaeng SH (2009) Domain-specific sentiment analysis using contextual feature generation. In: Proceeding of the International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement, ACM, TSA '09, pp 37–44, DOI http://doi.acm.org/10.1145/1651461.1651469

[24] Choudhury MD, Sundaram H, John A, Seligmann DD (2008) Multi-scale characterization of social network dynamics in the blogosphere. In: CIKM, pp 1515–1516, DOI http://doi.acm.org/10.1145/1458082.1458363

[25] Church KW, Hanks P (1989) Word association norms, mutual information, and lexicography. In: Proceedings of the 27th annual meeting on Association for Computational Linguistics, ACL, pp 76–83, DOI http://dx.doi.org/10.3115/981623.981633

[26] Cleveland WS, Loader CL (1996) Smoothing by local regression: Principles and methods. Statistical Theory and Computational Aspects of Smoothing pp 10–49

[27] Clifton C, Cooley R, Rennie J (2004) Topcat: Data mining for topic identification in a text corpus. TKDE 16(8):949–964, DOI http://doi.ieeecomputersociety.org/10.1109/TKDE.2004.32

[28] Cole R, Shasha D, Zhao X (2005) Fast window correlations over uncooperative time series. In: KDD, pp 743–749, DOI http://doi.acm.org/10.1145/1081870.1081966

[29] Crane R, Sornette D (2008) Robust dynamic classes revealed by measuring the response function of a social system. Proceedings of the National Academy of Sciences 105(41):15,649–15,653, DOI http://dx.doi.org/10.1073/pnas.0803685105

[30] Das M, Amer-Yahia S, Das G, Yu C (2011) MRI: Meaningful interpretations of collaborative ratings. In: VLDB, pp 1063–1074, URL http://www.vldb.org/pvldb/vol4/p1063-das.pdf

[31] Dasgupta S, Ng V (2009) Topic-wise, sentiment-wise, or otherwise?: Identifying the hidden dimension for unsupervised text classification. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, EMNLP '09, pp 580–589

[32] Dave K, Lawrence S, Pennock D (2003) Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th international conference on World Wide Web, ACM, New York, NY, USA, WWW '03, pp 519–528, DOI http://doi.acm.org/10.1145/775152.775226

[33] Denecke K, Brosowski M (2010) Topic detection in noisy data sources. In: Fifth International Conference on Digital Information Management, ICDIM

[34] Devitt A, Ahmad K (2007) Sentiment polarity identification in financial news: A cohesion-based approach. In: 45th Annual Meeting of the Association of Computational Linguistics

[35] Ekman P, Friesen WV, Ellsworth P (1982) What emotion categories or dimensions can observers judge from facial behavior? In: Emotion in the human face, Cambridge University Press, New York, pp 39–55

[36] Ennals R, Byler D, Agosta JM, Rosario B (2010) What is disputed on the web? In: Proceedings of the 4th ACM Workshop on Information Credibility on the Web, WICOW '10

[37] Ennals R, Trushkowsky B, Agosta JM (2010) Highlighting disputed claims on the web. In: Proceedings of the 19th international conference on World wide web, WWW '10

[38] Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, AAAI Press, pp 226–231

[39] Esuli A, Sebastiani F (2006) Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC '06

[40] Fahrni A, Klenner M (2008) Old Wine or Warm Beer: Target-Specific Sentiment Analysis of Adjectives. In: Proceedings of the Symposium on Affective Language in Human and Machine, AISB 2008, pp 60 – 63

[41] Fang Y, Si L, Somasundaram N, Yu Z (2012) Mining contrastive opinions on political texts using cross-perspective topic model. In: WSDM, pp 63–72, DOI http://doi.acm.org/10.1145/2124295.2124306

[42] Fellbaum C (ed) (1998) WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA

[43] Feng S, Wang D, Yu G, Yang C, Yang N (2009) Sentiment clustering: A novel method to explore in the blogosphere. In: Proceedings of the Joint International Conferences on Advances in Data and Web Management, APWeb/WAIM '09, pp 332–344, DOI http://dx.doi.org/10.1007/978-3-642-00672-2_30

[44] Gaikovich K (2004) Inverse problems in physical diagnostics. Nova Science Publishers, URL http://books.google.com/books?id=lXeOOsJcqGwC

[45] Galley M, McKeown K, Hirschberg J, Shriberg E (2004) Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In: ACL '04, Association for Computational Linguistics, pp 669–676, DOI http://dx.doi.org/10.3115/1218955.1219040

[46] Gamon M (2004) Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In: COLING '04, Association for Computational Linguistics, p 841, DOI http://dx.doi.org/10.3115/1220355.1220476

[47] Giampiccolo D, Dang HT, Magnini B, Dagan I, Cabrio E, Dolan B (2008) The fourth pascal recognizing textual entailment challenge. In: Proceedings of the First Text Analysis Conference, TAC '08

[48] Gindl S, Liegl J (2008) Evaluation of different sentiment detection methods for polarity classification on web-based reviews. In: Proceedings of the 18th European Conference on Artificial Intelligence, pp 35–43

[49] Glance NS, Hurst M, Tomokiyo T (2004) BlogPulse: Automated trend discovery for weblogs. In: WWW 2004 Workshop on the Weblogging Ecosystem, ACM

[50] Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. Tech. rep., Stanford, URL `http://cs.wmich.edu/~tllake/fileshare/TwitterDistantSupervision09.pdf`

[51] Godbole N, Srinivasaiah M, Skiena S (2007) Large-scale sentiment analysis for news and blogs. In: Proceedings of the International Conference on Weblogs and Social Media, ICWSM '07

[52] Goldberg A, Zhu X (2006) Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In: TextGraphs Workshop On Graph Based Methods For Natural Language Processing, URL `http://dl.acm.org/citation.cfm?id=1654758.1654769`

[53] Harabagiu S, Hickl A, Lacatusu F (2006) Negation, contrast and contradiction in text processing. In: AAAI'06: Proceedings of the 21st national conference on Artificial intelligence, pp 755–762

[54] He B, Macdonald C, He J, Ounis I (2008) An effective statistical approach to blog post opinion retrieval. In: CIKM, pp 1063–1072, DOI http://doi.acm.org/10.1145/1458082.1458223

[55] Hillard D, Ostendorf M, Shriberg E (2003) Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In: HLT-NAACL, URL `http://acl.ldc.upenn.edu/N/N03/N03-2012.pdf`

[56] Hoffman T (2008) Online reputation management is hot - but is it ethical? Computerworld

[57] Horrigan JA (2008) Online shopping. Pew Internet and American Life Project Report

[58] Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, KDD '04, pp 168–177, DOI http://doi.acm.org/10.1145/1014052.1014073

[59] Hu M, Liu B (2004) Mining opinion features in customer reviews. In: Mcguinness DL, Ferguson G, Mcguinness DL, Ferguson G (eds) AAAI, AAAI Press / The MIT Press, pp 755–760

[60] Hu X, Tang L, Tang J, Liu H (2013) Exploiting social relations for sentiment analysis in microblogging. In: WSDM, ACM, New York, NY, USA, pp 537–546, DOI http://doi.acm.org/10.1145/2433396.2433465

[61] IBM (2012) Social sentiment index. IBM Smarter Analytics
URL `http://www.ibm.com/analytics/us/en/conversations/social-sentiment.html`

[62] Jindal N, Liu B (2008) Opinion spam and analysis. In: WSDM, ACM, New York, NY, USA, pp 219–230, DOI http://doi.acm.org/10.1145/1341531.1341560

[63] Jo Y, Oh AH (2011) Aspect and sentiment unification model for online review analysis. In: WSDM, pp 815–824, DOI http://dx.doi.org/10.1145/1935826.1935932

[64] Johansson R, Moschitti A (2010) Reranking models in fine-grained opinion analysis. In: Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '10, pp 519–527

[65] Kamps J, Marx M, Mokken RJ, Rijke MD (2004) Using wordnet to measure semantic orientation of adjectives. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC '04, vol IV, pp 1115–1118

[66] Kim HD, Zhai C (2009) Generating comparative summaries of contradictory opinions in text. In: Proceedings of the 18th ACM conference on Information and knowledge management, ACM, New York, NY, USA, CIKM '09, pp 385–394, DOI http://doi.acm.org/10.1145/1645953.1646004

[67] Kim SM, Hovy E (2004) Determining the sentiment of opinions. In: Proceedings of the 20th international conference on Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA, COLING '04, p 1367, DOI http://dx.doi.org/10.3115/1220355.1220555

[68] Koppel M, Schler J (2006) The importance of neutral examples for learning sentiment. Computational Intelligence 22(2):100–109, DOI http://dx.doi.org/10.1111/j.1467-8640.2006.00276.x

[69] Ku LW, Lee LY, Wu TH, Chen HH (2005) Major topic detection and its application to opinion summarization. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, SIGIR '05, pp 627–628, DOI 10.1145/1076034.1076161

[70] Ku LW, Liang YT, Chen HH (2006) Opinion extraction, summarization and tracking in news and blog corpora. In: AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs

[71] Ku LW, Lo YS, Chen HH (2007) Using polarity scores of words for sentence-level opinion extraction. In: Proceedings of NTCIR-6 Workshop Meeting, pp 316–322

[72] Langford E, Schwertman N, Owens M (2001) Is the property of being positively correlated transitive? TAS 55(4):322–325, DOI http://www.tandfonline.com/doi/abs/10.1198/000313001753272286

[73] Lehmann J, Gonçalves B, Ramasco JJ, Cattuto C (2012) Dynamical classes of collective attention in twitter. In: WWW, pp 251–260, DOI http://doi.acm.org/10.1145/2187836.2187871

[74] Lerman K, Blair-Goldensohn S, Mcdonald R (2009) Sentiment summarization: Evaluating and learning user preferences. In: Proceedings of 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL '09

[75] Leskovec J, Backstrom L, Kleinberg J (2009) Meme-tracking and the dynamics of the news cycle. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, KDD '09, pp 497–506, DOI http://doi.acm.org/10.1145/1557019.1557077

[76] Leung CWK, Chan SCF, Chung FL (2006) Integrating collaborative filtering and sentiment analysis: A rating inference approach. In: ECAI 2006 Workshop on Recommender Systems, pp 62–66

[77] Li F, Huang M, Zhu X (2010) Sentiment analysis with global topics and local dependency. In: AAAI, URL http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1913

[78] Li F, Huang M, Yang Y, Zhu X (2011) Learning to identify review spam. In: IJCAI, pp 2488–2493, URL http://ijcai.org/papers11/Papers/IJCAI11-414.pdf

[79] Lim EP, Nguyen VA, Jindal N, Liu B, Lauw HW (2010) Detecting product review spammers using rating behaviors. In: Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10, pp 939–948, DOI http://doi.acm.org/10.1145/1871437.1871557

[80] Lin C, He Y (2009) Joint sentiment/topic model for sentiment analysis. In: Proceeding of the 18th ACM conference on Information and knowledge management, ACM, New York, NY, USA, CIKM '09, pp 375–384, DOI http://doi.acm.org/10.1145/1645953.1646003

[81] Lin YR, Margolin D, Keegan B, Lazer D (2013) Voices of victory: A computational focus group framework for tracking opinion shift in real time. In: Practice and Experience Track, WWW 2013

[82] Liu B (2010) Sentiment analysis and subjectivity. In: Indurkhya N, Damerau FJ (eds) Handbook of Natural Language Processing, Second Edition, CRC Press, Boca Raton, FL

[83] Liu B, Hu M, Cheng J (2005) Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of the 14th international conference on World Wide Web, ACM, New York, NY, USA, WWW '05, pp 342–351, DOI http://doi.acm.org/10.1145/1060745.1060797

[84] Liu H, Lieberman H, Selker T (2003) A model of textual affect sensing using real-world knowledge. In: Proceedings of the 8th International Conference on Intelligent User Interfaces, IUI '03, pp 125–33

[85] Liu J, Birnbaum L, Pardo B (2009) Spectrum: Retrieving different points of view from the blogosphere. In: Proceedings of the Third International Conference on Weblogs and Social Media

[86] Lu Y, Tsaparas P, Ntoulas A, Polanyi L (2010) Exploiting social context for review quality prediction. In: WWW, ACM, New York, NY, USA, pp 691–700, DOI http://doi.acm.org/10.1145/1772690.1772761

[87] Lu Y, Castellanos M, Dayal U, Zhai C (2011) Automatic construction of a context-aware sentiment lexicon: an optimization approach. In: WWW, pp 347–356, DOI http://doi.acm.org/10.1145/1963405.1963456

[88] Mandel B, Culotta A, Boulahanis J, Stark D, Lewis B, Rodrigue J (2012) A demographic analysis of online sentiment during hurricane Irene. In: Proceedings of 2nd Workshop on Language in Social Media, pp 27–36

[89] de Marneffe MC, Rafferty AN, Manning CD (2008) Finding contradictions in text. In: Proceedings of ACL: HLT, Association for Computational Linguistics, Columbus, Ohio, ACL '08, pp 1039–1047

[90] McArthur R (2008) Uncovering deep user context from blogs. In: Proceedings of the 2nd workshop on Analytics for noisy unstructured text data, AND '08, pp 47–54, DOI http://doi.acm.org/10.1145/1390749.1390758

[91] Mei Q, Ling X, Wondra M, Su H, Zhai C (2007) Topic sentiment mixture: modeling facets and opinions in weblogs. In: WWW, ACM, pp 171–180, DOI http://doi.acm.org/10.1145/1242572.1242596

[92] Melville P, Gryc W, Lawrence RD (2009) Sentiment analysis of blogs by combining lexical knowledge with text classification. In: KDD '09, pp 1275–1284, DOI http://doi.acm.org/10.1145/1557019.1557156

[93] Miao Q, Li Q, Dai R (2009) Amazing: A sentiment mining and retrieval system. Expert Syst Appl 36(3):7192–7198, DOI http://dx.doi.org/10.1016/j.eswa.2008.09.035

[94] Missen MM, Boughanem M (2009) Using wordnet's semantic relations for opinion detection in blogs. In: ECIR '09, pp 729–733, DOI http://dx.doi.org/10.1007/978-3-642-00958-7_75

[95] Morinaga S, Yamanishi K, Tateishi K, Fukushima T (2002) Mining product reputations on the web. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, KDD '02, pp 341–349, DOI http://doi.acm.org/10.1145/775047.775098

[96] Mueen A, Nath S, Liu J (2010) Fast approximate correlation for massive time-series data. In: SIGMOD, pp 171–182, DOI http://doi.acm.org/10.1145/1807167.1807188

[97] Mullen T, Malouf R (2006) A preliminary investigation into sentiment analysis of informal political discourse. In: AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs

[98] Nadeau D, Sabourin C, de Koninck J, Matwin S, Turney P (2006) Automatic dream sentiment analysis. In: Workshop on Computational Aesthetics at 21st National Conference on Artificial Intelligence, AAAI '06

[99] Nowson S (2009) Scary films good, scary flights bad: topic driven feature selection for classification of sentiment. In: Proceeding of the International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement, TSA '09, pp 17–24, DOI http://doi.acm.org/10.1145/1651461.1651465

[100] O'Connor B, Balasubramanyan R, Routledge BR, Smith NA (2010) From tweets to polls: Linking text sentiment to public opinion time series. In: ICWSM

[101] O'Hare N, Davy M, Bermingham A, Ferguson P, Sheridan P, Gurrin C, Smeaton AF (2009) Topic-dependent sentiment analysis of financial blogs. In: Proceeding of the International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement, TSA '09

[102] Osherenko A, André E (2007) Lexical affect sensing: Are affect dictionaries necessary to analyze affect? In: Proceedings of the 2nd international conference on Affective Computing and Intelligent Interaction, ACII '07, pp 230–241, DOI http://dx.doi.org/10.1007/978-3-540-74889-2_21

[103] Pado S, de Marneffe MC, MacCartney B, Rafferty AN, Yeh E, Manning CD (2008) Deciding entailment and contradiction with stochastic and edit distance-based alignment. In: Proceedings of the First Text Analysis Conference, TAC '08

[104] Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining. In: LREC, URL `http://www.lrec-conf.org/proceedings/lrec2010/summaries/385.html`

[105] Pang B, Lee L (2004) A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting of ACL, pp 271–278

[106] Pang B, Lee L (2005) Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: ACL

[107] Pang B, Lee L (2008) Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2):1–135

[108] Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? sentiment classification using machine learning techniques. In: EMNLP, pp 79–86

[109] Papadimitriou S, Sun J, Faloutsos C (2005) Streaming pattern discovery in multiple time-series. In: VLDB, pp 697–708, URL http://www.vldb2005.org/program/paper/thu/p697-papadimitriou.pdf

[110] Papadimitriou S, Sun J, Yu PS (2006) Local correlation tracking in time series. In: ICDM, pp 456–465, DOI http://doi.ieeecomputersociety.org/10.1109/ICDM.2006.99

[111] Paul MJ, Girju R (2010) A two-dimensional topic-aspect model for discovering multi-faceted topics. In: AAAI, URL http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1730

[112] Paul MJ, Zhai C, Girju R (2010) Summarizing contrastive viewpoints in opinionated text. In: EMNLP, pp 66–76, URL http://www.aclweb.org/anthology/D10-1007

[113] Popescu AM, Pennacchiotti M (2010) Detecting controversial events from twitter. In: CIKM, pp 1873–1876, DOI http://dx.doi.org/10.1145/1871437.1871751

[114] Prabowo R, Thelwall M (2009) Sentiment analysis: A combined approach. Journal of Informetrics 3(2):143–157, DOI http://dx.doi.org/10.1016/j.joi.2009.01.003

[115] Read J, Carroll J (2009) Weakly supervised techniques for domain-independent sentiment classification. In: Proceeding of the International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement, ACM, New York, NY, USA, TSA '09, pp 45–52, DOI http://doi.acm.org/10.1145/1651461.1651470

[116] Riloff E, Wiebe J, Phillips W (2005) Exploiting subjectivity classification to improve information extraction. In: Veloso MM, Kambhampati S (eds) AAAI, AAAI Press / The MIT Press, pp 1106–1111

[117] Sakurai Y, Papadimitriou S, Faloutsos C (2005) BRAID: Stream mining through group lag correlations. In: SIGMOD, pp 599–610, DOI http://doi.acm.org/10.1145/1066157.1066226

[118] Shimada K, Endo T (2008) Seeing several stars: A rating inference task for a document containing several evaluation criteria. In: PAKDD, pp 1006–1014

[119] Sornette D, Deschâtres F, Gilbert T, Ageon Y (2004) Endogenous versus exogenous shocks in complex networks: An empirical test using book sale rankings. Phys Rev Lett DOI 10.1103/PhysRevLett.93.228701

[120] Spruill NL, Gastwirth JL (1982) On the estimation of the correlation coefficient from grouped data. JASA 77(379):614–620, DOI http://www.tandfonline.com/doi/abs/10.1080/01621459.1982.10477860

[121] Stoyanov V, Cardie C (2008) Annotating topics of opinions. In: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco

[122] Stoyanov V, Cardie C (2008) Topic identification for fine-grained opinion analysis. In: Proceedings of the 22nd International Conference on Computational Linguistics, Manchester, UK, Coling '08, pp 817–824

[123] Taboada M, Anthony C, Voll K (2006) Methods for creating semantic orientation dictionaries. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC '06, pp 427–432

[124] Taboada M, Gillies MA, McFetridge P (2006) Sentiment classification techniques for tracking literary reputation. In: Proceedings of LREC Workshop Towards Computational Models of Literary Analysis, pp 36–43

[125] Tang H, Tan S, Cheng X (2009) A survey on sentiment detection of reviews. Expert Syst Appl 36(7):10,760–10,773, DOI http://dx.doi.org/10.1016/j.eswa.2009.02.063

[126] Thelwall M, Buckley K, Paltoglou G (2011) Sentiment in twitter events. JASIST 62(2):406–418

[127] Thet TT, Na JC, Khoo CS, Shakthikumar S (2009) Sentiment analysis of movie reviews on discussion boards using a linguistic approach. In: Proceeding of the International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement, TSA '09

[128] Thomas M, Pang B, Lee L (2006) Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In: EMNLP, pp 327–335, URL http://dl.acm.org/citation.cfm?id=1610075.1610122

[129] Titov I, McDonald RT (2008) A joint model of text and aspect ratings for sentiment summarization. In: ACL, pp 308–316, URL http://www.aclweb.org/anthology/P08-1036

[130] Tsytsarau M, Palpanas T (2011) Survey on mining subjective data on the web. Data Mining and Knowledge Discovery, Special Issue on 10 Years of Mining the Web pp 1–37, DOI http://dx.doi.org/10.1007/s10618-011-0238-6, iSSN 1384-5810

[131] Tsytsarau M, Palpanas T (2011) Towards a framework for detecting and managing opinion contradictions. In: ICDM Workshops, pp 1219–1222, DOI http://doi.ieeecomputersociety.org/10.1109/ICDMW.2011.167

[132] Tsytsarau M, Palpanas T, Denecke K, Brosowski M (2009) Scalable Discovery of Contradicting Opinions in Weblogs. Tech. Rep. DISI-09-038, DISI, University of Trento

[133] Tsytsarau M, Palpanas T, Denecke K (2010) Scalable discovery of contradictions on the web. In: Proceedings of the 19th international conference on World wide web, WWW '10, pp 1195–1196, DOI 10.1145/1772690.1772871

[134] Tsytsarau M, Palpanas T, Denecke K (2011) Scalable detection of sentiment-based contradictions. In: The First International Workshop on Knowledge Diversity on the Web, Colocated with WWW 2011

[135] Tsytsarau M, Palpanas T, Castellanos M, Hsu M, Dayal U (2012) Identifying news events that cause a shift in sentiment. US Patent HP-82962988 (pending)

[136] Tsytsarau M, Amer-Yahia S, Palpanas T (2013) Efficient sentiment correlation for large-scale demographics. In: SIGMOD (accepted for publication), pp 1–12

[137] Tumasjan A, Sprenger TO, Sandner PG, Welpe IM (2010) Predicting elections with twitter: What 140 characters reveal about political sentiment. In: ICWSM, URL http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1441

[138] Turney P, Littman M (2003) Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems 21:315–346

[139] Turney PD (2002) Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on ACL, Association for Computational Linguistics, Morristown, NJ, USA, ACL '02, pp 417–424, DOI http://dx.doi.org/10.3115/1073083.1073153

[140] Varlamis I, Vassalos V, Palaios A (2008) Monitoring the evolution of interests in the blogosphere. In: ICDE Workshops, pp 513–518, DOI http://dx.doi.org/10.1109/ICDEW.2008.4498371

[141] Voorhees EM (2008) Contradictions and justifications: Extensions to the textual entailment task. In: Proceedings of ACL: HLT, Association for Computational Linguistics, Columbus, Ohio, ACL '08, pp 63–71

[142] Wang F, Helian N, Yip YJ (2002) Run-length encoding applied to grid space storing and indexing. In: Proceedings of the IADIS International Conference WWW/Internet, ICWI 2002, pp 461–471

[143] Wang H, Lu Y, Zhai C (2010) Latent aspect rating analysis on review text data: a rating regression approach. In: KDD, pp 783–792, DOI http://doi.acm.org/10.1145/1835804.1835903

[144] Wartena C, Brussee R (2008) Topic detection by clustering keywords. In: DEXA, pp 54–58

[145] Wiebe J, Riloff E (2005) Creating subjective and objective sentence classifiers from unannotated texts. In: CICLing-2005, pp 486–497, DOI http://dx.doi.org/10.1007/978-3-540-30586-6_53

[146] Wiebe J, Wilson T, Bell M (2001) Identifying collocations for recognizing opinions. In: Proceedings of the ACL Workshop on Collocation: Computational Extraction, Analysis, and Exploitation, pp 24–31

[147] Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP

[148] Wu F, Huberman BA (2007) Novelty and collective attention. Proceedings of the National Academy of Sciences 104(45):17,599–17,601, DOI http://dx.doi.org/10.1073/pnas.0704916104

[149] Xia Y, Prabhakar S, Lei S, Cheng R, Shah R (2005) Indexing continuously changing data with mean-variance tree. In: SAC, pp 1125–1132, DOI http://doi.acm.org/10.1145/1066677.1066932

[150] Xing D, Girolami M (2007) Employing latent dirichlet allocation for fraud detection in telecommunications. Pattern Recognition Letters 28(13):1727–1734, DOI http://dx.doi.org/10.1016/j.patrec.2007.04.015

[151] Yi J, Nasukawa T, Bunescu R, Niblack W (2003) Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In: Proceedings of the IEEE International Conference on Data Mining, ICDM '03, DOI http://doi.ieeecomputersociety.org/10.1109/ICDM.2003.1250949

[152] Yu H, Hatzivassiloglou V (2003) Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In: EMNLP, pp 129–136, URL `http://portal.acm.org/citation.cfm?id=1119355.1119372`

[153] Zhang J, Kawai Y, Kumamoto T, Tanaka K (2009) A novel visualization method for distinction of web news sentiment. In: WISE, pp 181–194, DOI http://dx.doi.org/10.1007/978-3-642-04409-0_22

[154] Zhang W, Yu CT, Meng W (2007) Opinion retrieval from blogs. In: CIKM, pp 831–840, DOI http://doi.acm.org/10.1145/1321440.1321555

[155] Zhou L, Chaovalit P (2008) Ontology-supported polarity mining. J Am Soc Inf Sci Technol 59:98–110, DOI http://dx.doi.org/10.1002/asi.v59:1

[156] Zhu J, Zhu M, Wang H, Tsou BK (2009) Aspect-based sentence segmentation for sentiment summarization. In: Proceeding of the International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement, ACM, TSA '09, pp 65–72, DOI http://doi.acm.org/10.1145/1651461.1651474

[157] Zhu Y, Shasha D (2002) Statstream: statistical monitoring of thousands of data streams in real time. In: VLDB, pp 358–369, URL `http://dl.acm.org/citation.cfm?id=1287369.1287401`

Appendix

Year	Authors	Type	Topic	Algorithms	Range	Scope	Data	Scale
’02	Morinaga et al	A	Y	RuleBased+Dictionary	C	Reviews (P)	N/A	L
’02	Turney	C	N	Statistic	C	Reviews(M,P,S)	EP	L
’02	Pang et al	C	N	NB, ME, SVM	2	Reviews (M)	IMDB	L
’03	Liu et al	C	N	NLP + Dictionary	C	Texts	N/A	S
’03	Turney and Littman	C	N	LSA, Statistic (PMI)	C	Words	GI, HM	S
’03	Dave et al	A	N	NB	2	Reviews (P)	AZ, CN	L
’03	Yi et al	C	Y	Dictionary	3	Reviews (M,P)	N/A	L
’03	Yu and Hatzivassiloglou	C	N	Statistic	3	News	TREC	L
’04	Kim and Hovy	C	Y	Semantic	2	News	DUC	S
’04	Galley et al	C	N	ME, CMM	2	Transcripts	N/A	S
’04	Hu and Liu	A	Y	Semantic + RuleBased	2	Reviews (P)	AZ, CN	L
’04	Gamon	C	N	SVM	4	Reviews(S)	N/A	L
’04	Kamps et al	C	N	Semantic	C	Texts	GI	S
’05	Alm et al	C	N	Linear Classifier	2	Fairytales	N/A	S
’05	Ku et al	A	Y	Dictionary	2	News	TREC	L
’05	Chaovalit and Zhou	C	N	ML, Statistic (PMI)	2, C	Reviews (M)	IMDB	L
’05	Liu et al	A	Y	Semantic + RuleBased	2	Reviews (P)	N/A	M
’05	Pang and Lee	C	N	SVM OVA, SVR + ML	3, 4	Reviews (M)	IMDB	S
’06	Thomas et al	C	N	Multi SVM	2	Transcripts	GovTrack	S
’06	Leung et al	C	N	Statistic	3	Reviews (M)	N/A	S
’06	Taboada et al	C	N	Statistic (PMI)	2	Reviews	EP	L
’06	Carenini et al	A	Y	Semantic	2	Reviews (P)	N/A	M
’06	Ku et al	A	Y	Statistic	C	News, Blogs	TREC, NTCIR	L
’06	Goldberg and Zhu	C	N	Graph, SVR	4	Reviews (M)	IMDB	L
’06	Taboada et al	C	N	Dictionary	C	Reviews (B)	N/A	S
’07	Godbole et al	C	Y	Semantic	C	News, Blogs	N/A	L
’07	Osherenko and André	C	N	SVM + Dictionary	4	Texts	SAL	L
’07	Zhang et al	C	Y	SVM	2	Blogs	EP, RA	S
’07	Devitt and Ahmad	C	N	Semantic	2	News	News	L
’07	Mei et al	C	Y	HMM	2	Blogs	N/A	L

Continued. . .

Year	Authors	Type	Topic	Algorithms	Range	Scope	Data	Scale
’07	Ku et al	C	N	Statistic	C	News	NTCIR	L
’07	Chen et al	A	N	DT, SVM	2	Reviews (B)	N/A	S
’08	Annett and Kondrak	C	N	SVM, NB, ADT	2	Reviews (M)	IMDB	M
’08	He et al	C	N	Statistic (IR)	C	Blogs	TREC	M
’08	Bestgen	C	N	Statistics (SO-LSA)	2	Words	N/A	L
’08	Fahrni and Klenner	C	Y	Statistic	C	Reviews (S)	N/A	L
’08	Shimada and Endo	C	Y	SVM OVA, ME, SVR	3, 6	Reviews (P)	N/A	L
’09	Zhang et al	A	Y	Corpus	C	News	N/A	L
’09	Miao et al	A	Y	Dictionary	2	Reviews (P)	N/A	M
’09	Zhu et al	A	Y	Dictionary	3	Reviews (S)	N/A	M
’09	Nadeau et al	C	N	LR, NB + Dictionary	4	Dreams	N/A	S
’09	Bodendorf and Kaiser	C	N	SVM OVA	3	Blogs	N/A	M
’09	Choi et al	C	Y	Clustering +Dictionary	3	News	NTCIR	S
’09	Lin and He	C	Y	LDA + Dictionary	2	Texts	IMDB	S
’09	Nowson	C	Y	SVM	2	Reviews (P)	N/A	S
’09	Melville et al	C	N	NB + Dictionary	2	Blogs	N/A	L
’09	Thet et al	C	Y	Dictionary	C	Reviews (M)	IMDB	L
’09	Prabowo and Thelwall	C	N	RuleBased, Dictionary, Statistic, SVM	2	Reviews(M,P)	IMDB,N/A	L
’09	Feng et al	A	Y	Dictionary	2	Blogs	N/A	L
’09	Lerman et al	A	N	Semantic	C	Reviews(P)	N/A	L
’09	O’Hare et al	C	Y	MNB, SVM	2, 3	Blogs	N/A	L
’09	Dasgupta and Ng	C	N	SVM + Clustering	2	Texts	IMDB,AZ	S
’09	Missen and Boughanem	C	Y	Semantic	C	Blogs	TREC	M
’09	Read and Carroll	C	N	Statistic	2	News,Reviews(M)	IMDB, GI	L, S
’09	Go et al	C	N	NB, ME, SVM	2	Microblogs	Twitter	L
’10	Bollen et al	A	N	OpinionFinder, Statistic (PMI)	C	Microblogs	Twitter	L
’10	Tumasjan et al	A	Y	Dictionary(LIWC)	C	Microblogs	Twitter	L
’10	Bifet and Frank	C	N	MNB, SGD, Hoeffding tree	2	Microblogs	Twitter	L
’10	Pak and Paroubek	C	N	MNB	3	Microblogs	Twitter	L

Table 1: An overview of the most popular sentiment extraction methods, used in Subjectivity Analysis.

Paper	Dataset	Sentiment Algorithm (Precision, %)
Dave et al [32]	AZ, CN	SVM (85.8 - 87.2) NB (81.9 - 87.0)
Hu and Liu [58]	AZ, CN	Semantic (84.0)
Turney [139]	EP	Statistics (74.4)
Taboada et al [123]	EP	PMI (56.8)
Turney and Littman [138]	HM	SO-LSA (67.7 - 88.9) PMI (61.8 - 71.0)
	GI	SO-LSA (65.3 - 82.0) PMI (61.3 - 68.7)
Kamps et al [65]	GI	Semantic (76.7)
Read and Carroll [115]	GI	PMI (71.7) Semantic Space (83.8) Similarity (67.6)
	SemEval*	PMI (46.4) Semantic Space (44.4) Similarity (53.1)
	IMDB	PMI (68.7) Semantic Space (66.7) Similarity (60.8)
Gindl and Liegl [48], average	AZ (N/A)	Dictionary (59.5 - 62.4) NB (66.0) ME (83.8)
	TA (N/A)	Dictionary (70.9 - 76.4) NB (72.4) ME (78.9)
	IMDB	Dictionary (61.8 - 64.9) NB (58.5) ME (82.3)
Pang et al [108]	IMDB	NB (81.5) ME (81.0) SVM (82.9)
Chaovalit and Zhou [18]	IMDB	N-Gram (66.0 - 85.0) PMI (77.0)
Goldberg and Zhu [52]	IMDB	SVR (50.0 - 59.2) Graph (36.6 - 54.6)
Annett and Kondrak [3]	IMDB	NB (77.5) SVM (77.4) ADTree (69.3)
Thet et al [127]	IMDB	Dictionary (81.0)
Ku et al [71]	NTCIR	Statistics (66.4)
Choi et al [23]	NTCIR	Dictionary + Clustering ($\sim$70.0)
Osherenko and André [102]	SAL*	SVM + Dictionary (34.5)
Yu and Hatzivassiloglou [152]	TREC	Statistics (68.0 - 90.0)
Ku et al [69]	TREC	Dictionary (62.0)
Missen and Boughanem [94]	TREC	Semantic (MAP 28.0, P@10 64.0)
Yi et al [151]	N/A	Dictionary (87.0 Reviews, 91.0 - 93.0 News)
Gamon [46]	N/A	SVM (69.0 nearest classes, 85.0 farthest classes)
Kim and Hovy [67]	N/A	Semantic (67.0 - 81.0)
Thomas et al [128]	N/A	Multiple SVM (71.0)
Nadeau et al [98]	N/A*	LR (35.0 - 50.0) NB + Dictionary (38.0)
Chen et al [19]	N/A	DT (71.7) SVM (84.6) NB (77.5)
Devitt and Ahmad [34]	N/A	Semantic (50.0 - 58.0, f-measure)
Shimada and Endo [118]	N/A*	SVM OVA (58.4) ME (57.1) SVR (57.4) SIM (55.7)
O'Hare et al [101]	N/A	MNB (75.1) SVM (74.4)
Zhu et al [156]	N/A	Dictionary (69.0)
Bodendorf and Kaiser [12]	N/A	SVM OVA (69.0)
Melville et al [92]	N/A	NB + Dictionary (63.0 - 91.0)
Prabowo and Thelwall [114]	N/A	SVM-only (87.3) SVM + RuleBased + Dictionary + Statistics (91.0)
Feng et al [43]	N/A	Dictionary (65.0)
Go et al [50]	TS	NB (82.7) ME (83.0) SVM (82.2)

Continued…

Paper	Dataset	Sentiment Algorithm (Precision, %)
Bifet and Frank [9]	TS	MNB (82.5) SGD (78.6), Hoeffding tree (69.4)
	N/A	MNB (86.1) SGD (86.3) Hoeffding tree (84.8)
Pak and Paroubek [104]	N/A	MNB (70.0) at recall value 60.0

Table 2: Precision of sentiment extraction for different implementations according to the data reported by authors. In this table we only report best-run results for the available datasets (which are also listed in Table 3). "N/A" means that the dataset is not publicly available. 3(5)-classes accuracy marked with *.

Name (and URL)	Year	Type	Pos	Neg	Neu	Range
GI - General Inquirer content analysis system *www.wjh.harvard.edu/~inquirer/*	2002	Words	N/A	N/A	N/A	D
IMDB - Movie Review Data v2.0 *www.cs.cornell.edu/People/pabo/movie-review-data/*	2004	Reviews	1,000	1,000	0	B
TREC - Blog Track *http://ir.dcs.gla.ac.uk/test_collections/blog06info.html*	2006	Blogs	3,215,171 total			N/A
AZ - Amazon Reviews *www.cs.uic.edu/~liub/FBS/sentiment-analysis.html*	2007	Reviews	4,554K	759K	525K	D
SemEval-2007 Affective Text Task *www.cse.unt.edu/~rada/affectivetext/*	2007	News	561	674	15	D
NTCIR-MOAT - *http://research.nii.ac.jp/ntcir/*	2008	News	26K total			D
SAL - Sensitive Artificial Listener corpus *www.vf.utwente.nl/~hofs/sal/*	2008	Speech	672 turns total			D
Irish Economic Sentiment Dataset *www.mlg.ucd.ie/sentiment/*	2009	News	2,608	3,915	3,080	D
Multi-Domain Sentiment Dataset v2.0 *www.cs.jhu.edu/~mdredze/datasets/sentiment/*	2009	Reviews	10,771	10,898	0	D
TS - Twitter Sentiment *http://twittersentiment.appspot.com/*	2009	Micro-blogs	800K	800K	0	B
TA - TripAdvisor.com *http://times.cs.uiuc.edu/~wang296/Data/*	2010	Reviews	183K	37K	26K	D
EP - Epinions - *www.epinions.com*	2010	Reviews	N/A	N/A	N/A	D
CN - C|net - *www.cnet.com*	2010	Reviews	N/A	N/A	N/A	D
RA - RateItAll - *www.rateitall.com*	2010	Reviews	N/A	N/A	N/A	D
ZD - ZDnet - *www.zdnet.com*	2010	Reviews	N/A	N/A	N/A	D

Table 3: An overview of the most popular opinion mining datasets and data sources. Under the columns "Pos", "Neg" and "Neu" we list the approximate numbers of positive, negative and neutral labels respectively. The range of these labels is either binary, marked as "B", or discrete, marked as "D". "N/A" means that the dataset is not publicly available.

Attribute	Type	Description
`topic`	SMALLINT	Topic identifier
`timeBegin`, `timeEnd`	INT	Start and end of the time window
`granularity`	BYTE	A level of granularity
`n`	FLOAT	The number of posts within interval
`M1`, `M2`	FLOAT	1^{st} and 2^{nd} moments of sentiments

Table 4: Attributes of database storage Cdb.

***AdaptContr main view*: Outputs nodes of specific granularity using adaptive threshold.**
`create view 'AdaptContr' as select c1.topic, c1.timeBegin, c1.timeEnd from Contr c1 join Contr c2 on c1.topic = c2.topic and c2.granularity = c1.granularity + 1 and c1.timeBegin is between c2.timeBegin and c2.timeEnd where c1.contradiction` $\geq$ `c2.contradiction * @threshold and c1.granularity = @window and (c1.timeBegin is between startDate and endDate or c1.timeEnd is between startDate and endDate);`
***Query 1*: For the Problem 2, with a restriction on a specific topic.**
`select * from AdaptContr c1 where c1.topic = @topic;`
***Query 2*: For the Problem 3, identifying contradictions for all topics.**
`select * from AdaptContr c1 order by c1.contradiction;`
***Query 3*: For the Problem 3, distinguishing between sync. and async. types.**
`select * from AdaptContr c1 join Contr c2 on c2.topic = c1.topic and c2.timeBegin = c1.timeEnd and c2.granularity = c1.granularity join Contr c3 on c3.topic = c1.topic and c3.timeEnd = c1.timeBegin and c3.granularity = c1.granularity where c2.m1 * c3.m1` $\leq$ `0;`

Table 5: Queries used in the evaluation of Cdb performance.