Filipa Alexandra de Madureira Peleja

M.Sc.

Learning Domain-Specific Sentiment Lexicons with Applications to Recommender Systems

Dissertação para obtenção do Grau de Doutor em Informática

Orientador: Professor Doutor João Miguel da Costa Magalhães

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Outubro, 2015

Learning Domain-Specific Sentiment Lexicons with Applications to Recommender Systems

Aos meus pais

e querida irmã

Acknowledgments

First of all, I would like to express my gratitude to my supervisor, João Magalhães. He provided me with invaluable advice and helped me comprehend the right research practice making these years of research a real pleasure.

I am very thankful to my PhD panel committee members, Cristina Ribeiro, Pável Calado, Gabriel Lopes and Nuno Marques, who agreed to proofread this dissertation, and who most definitely improved its quality through their comments and suggestions.

The pleasant atmosphere I found at the workplace I owe to my office mates and friends at NOVA-LINCS, Rui Nóbrega, Jorge Costa, Bruno Cardoso, André Sabino, Rui Madeira, Diogo Cabral, Serhiy Moskovchuk, Rossana Santos, Miguel Domingues, Miguel Lourenço, Luisa Lourenço, Inês Rodolfo, Sinan Egilmez, Sofia Gomes, Pedro Centieiro, Pedro Santos, João Santos, Pedro Dias, João Casteleiro, Sofia Reis, Tiago Santos, Flávio Martins, André Mourão, Carmen Morgado, Nuno Correia and Teresa Romão. I really appreciated helpful discussions, much needed coffee breaks and after-work drinks.

I would like to thank my parents Elizabete Peleja and Guilherme Madureira, my sister Joana Peleja, my grandmother Nercínia Andrade, my uncle António Peleja and Rosário Ferreirinha for the unreserved support and love all these years. Thank all my friends who always supported me: Rui Nóbrega, Joana Peleja, Rute Tomaz, Pedro Antunes, Jorge Costa, José Mourinho, Pedro Viegas, Fátima Bernardes, Bruno Magalhães and last but not least, Kisha and Tuxinhas. Finally, I will always owe very important debt to my beloved Luis Costa and mother Elizabete Peleja for patience and encouragement and for helping me on maintaining the body, mind and soul balance throughout this long way.

Abstract

Search is now going beyond looking for factual information, and people wish to search for the opinions of others to help them in their own decision-making. Sentiment expressions or opinion expressions are used by users to express their opinion and embody important pieces of information, particularly in online commerce. The main problem that the present dissertation addresses is how to model text to find meaningful words that express a sentiment. In this context, I investigate the viability of automatically generating a sentiment lexicon for opinion retrieval and sentiment classification applications. For this research objective we propose to capture sentiment words that are derived from online users' reviews. In this approach, we tackle a major challenge in sentiment analysis which is the detection of words that express subjective preference and domain-specific sentiment words such as jargon. To this aim we present a fully generative method that automatically learns a domain-specific lexicon and is fully independent of external sources.

Sentiment lexicons can be applied in a broad set of applications, however popular recommendation algorithms have somehow been disconnected from sentiment analysis. Therefore, we present a study that explores the viability of applying sentiment analysis techniques to infer ratings in a recommendation algorithm. Furthermore, entities' reputation is intrinsically associated with sentiment words that have a positive or negative relation with those entities. Hence, is provided a study that observes the viability of using a domain-specific lexicon to compute entities reputation. Finally, a recommendation system algorithm is improved with the use of sentiment-based ratings and entities reputation.

Contents

List of Figures

List of Tables

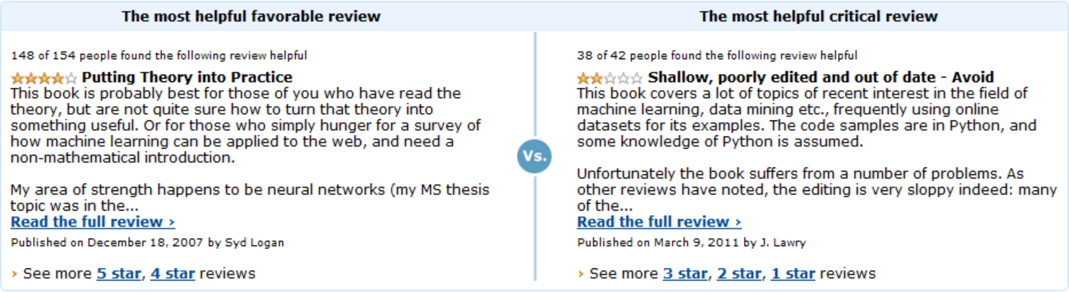
1

Introduction

1.1 Context and Challenges

Communication and interaction among people has significantly changed with the growth of the World Wide Web. Vast amounts of information is available in digital format, and much of this information is stored in unstructured formats and not organised. This has an important impact in users' behaviour. Users have become more demanding and are willing to give a certain amount of effort when trying to find information. More specifically, users have changed their behaviour on different aspects, and in the present thesis we will focus on the fact that nowadays users search for an opinion about different products online instead of only trusting their close circle of friends (Pang and Lee, 2008).

Technological advances in portable devices has facilitated the easy communication and spread of information via popular social-media platforms. People share, comment and publish their opinions on blogs, social networks, forums and other social media channels (Kwak et al., 2010). Social-media mentions to people, public figures, organizations or products emerge constantly and move rapidly over large communities. Nowadays users changed their behaviour from sharing and commenting what is happening around them, to search for recommendations and opinions reported by the other people that also participate in the same social-media platform. This phenomenon created a relationship between people opinions and entities reputation. The information targeting entities is generally controlled by users and consumers (Jansen et al., 2009; Glance et al., 2005). Figure 1 illustrates two comparative reviews from Amazon, a popular Web site that allows users to comment on their purchases. This visualization can help users define their opinion about a product (e.g. movie), and it is based on what other users have commented.



The most helpful favorable review

148 of 154 people found the following review helpful

★★★★☆ **Putting Theory into Practice**

This book is probably best for those of you who have read the theory, but are not quite sure how to turn that theory into something useful. Or for those who simply hunger for a survey of how machine learning can be applied to the web, and need a non-mathematical introduction.

My area of strength happens to be neural networks (my MS thesis topic was in the...

Read the full review ›

Published on December 18, 2007 by Syd Logan

› See more 5 star, 4 star reviews

The most helpful critical review

38 of 42 people found the following review helpful

★★☆☆☆ **Shallow, poorly edited and out of date - Avoid**

This book covers a lot of topics of recent interest in the field of machine learning, data mining etc., frequently using online datasets for its examples. The code samples are in Python, and some knowledge of Python is assumed.

Unfortunately the book suffers from a number of problems. As other reviews have noted, the editing is very sloppy indeed: many of the...

Read the full review ›

Published on March 9, 2011 by J. Lawry

› See more 3 star, 2 star, 1 star reviews

Figure 1. Example of Amazon comparative reviews.

The analysis of humans' viewpoints is known as sentiment analysis. Although this field of study had some research prior to year 2000 (Wiebe, 1994; Hatzivassiloglou and McKeown, 1997; Wiebe, 1990) this was the year in which sentiment analysis emerged as a very active research area. Sentiment analysis deals with an opinion-oriented natural language processing problem and is typically applied to the analysis of structured or unstructured free text documents. These text documents contain users' opinions about one or more entities such as products, organizations, individuals and their features.

The analysis of users' opinions is known by different names: *sentiment analysis*, *emotion analysis*, *opinion mining*, *opinion extraction*, *sentiment mining*, *subjectivity analysis*, *affect analysis*, *emotion analysis* and *review mining* (Liu, 2012). Opinions include appraisals, thoughts and emotions about a product or individual. Usually an opinion is given in a form of a review, comment or purchase evaluation. The sentiment analysis of an opinion can be defined as mapping the text to one of the sentiment classes (labels) from a predefined set. Usually the classes of the predefined set are *negative* and *positive*; *objective* and *subjective*; *negative*, *positive* and *neutral* or in a range of numbers such as 1 to 5 or 1 to 10 (Pang and Lee, 2008).

Sentiment analysis enfolds various techniques to detect words that express a positive and negative feeling or emotion (Liu, 2012). These words are commonly known as *sentiment words* or *opinion words*. Beyond words, n-grams (contiguous sequence of *n* words) and idiomatic expressions are commonly used as sentiment words (e.g. the word *terrible*, the n-gram *quite wonderful* and the idiomatic expression *break a leg*). Sentiment words are able to represent which words are more likely to be valuable in each sentiment class. For this reason sentiment words have proven to be valuable in sentiment classification tasks (Liu, 2012). Consequently, the past decade has witnessed a considerable high volume of research in numerous algorithms working on compiling a set of sentiment words (known as sentiment lexicons) (Takamura et al., 2005; Baccianella et al., 2010; Esuli and Sebastiani, 2006; Rao and Ravichandran, 2009; Velikovich et al., 2010; Ding et al., 2008). In this context, we propose a method to compile a sentiment lexicon. To this end, a set statistical models are used to identify which words are more relevant for each sentiment level. This method shows that is possible to develop a framework that uses topic models in a sentiment analysis problem (Blei et al., 2003; Blei and McAuliffe, 2007). Topic models gained popularity as a tool for automatic corpus summarization and document browsing on large scale data. Such models have been integrated in the context of online commerce and are able to identify important pieces of information (i.e. sentiment expressions) (Moghaddam and Ester, 2011; Ramage et al., 2009; Titov and McDonald, 2008).

Opinionated text also known as subjective text is a set of words, phrases or sentences that express a sentiment. The difference between opinionated text and factual text is centred in the notion of *private state*. As Quirk et al. (1985) define it, a *private state* is a state that is not open to objective observation or verification: "*a person may be observed to assert that God exists, but not to believe that God exists. Belief is in this sense 'private'.*" (p. 1181). More recently the term *subjectivity* for this concept has been adopted by the community (Wilson and Wiebe, 2005; Liu, 2012). Although this area has been researched in academia the problem is still far from being completely solved (Liu, 2012). One of the main challenges is that opinionated language varies over a broad range of discourse, a system with a fixed vocabulary will not be enough to represent users' opinions (Wilson et al., 2004). Also, sentiment words have a natural association to people' opinions and opinions tend to target specific people, organizations or products. For this reason, this thesis deals with the problem of identifying sentiment words and measuring an entity's reputation. In the present thesis we aim to: compute a sentiment lexicon that is human independent and useful for different domains; characterize entities' reputation through their association with a domain sentiment lexicon; and learn how to improve recommendation algorithms with sentiment knowledge. For the mentioned tasks there are a number of challenges to overcome:

- **Sentiment lexicons:** One of the most important indicators in the analysis of subjective text are sentiment words. Researchers have examined the viability of building such lexicons (Velikovich et al., 2010; Rao and Ravichandran, 2009; Weichselbraun et al., 2013; Baccianella et al., 2010), and for this task researchers tackle the problem in three main approaches: manual, dictionary-based and corpus-based. Obtaining a sentiment lexicon is an important and complex step which contains many unsolved questions (Liu, 2012). Depending on the domain, sentiment words may have opposite directions; sentences containing sentiment words may not express any sentiment or (the opposite) sentences without sentiment words may be used to express a sentiment; users' opinions frequently enclose sarcastic and idiomatic sentences; and sentiment words come in different strengths which may be interpreted in a scale with different intensities (Wilson et al., 2004; Liu, 2012).

- **Reference entities:** Intuitively, sentiment words are associated with words or phrases that express a sentiment. For example, *good*, *wonderful*, *poor* and *terrible* represent sentiment words. However, beyond these words there are numerous words that are used to express a sentiment, e.g. *Bollywood* encloses a sentiment value in "*Queen is not another Bollywood movie*." In this example is not obvious the sentiment expressed by *Bollywood*. Specific products, organizations or named entities characteristics are used in subjective sentences to express a sentiment. However, one must keep in mind that these entities might only reveal a sentiment in particular application domains (Ding et al., 2008; Liu, 2010).
- **Reputation of entities:** identify relevant references that influence entities reputation is an ongoing research problem. In a sentiment analysis problem and from an algorithmic perspective, the challenge is to analyse how sentiment words affect entities' public image. Previous work (Zhao et al., 2010; Jo and Oh, 2011; Hu and Liu, 2004b; Liu, 2012) have made significant advances in detecting product aspects or features. However, unlike opinions about products, entities are not structured around a fixed set of aspects or features which imply a more challenging task (Albornoz et al., 2012).
- **Sentiment-based recommendations:** recommendation algorithms have proven their ability to provide valuable recommendations to different users (Koren et al., 2009). However, recommendation algorithms are more likely to use explicit ratings which we believe is a limited metric for assessing specific opinions about different products (e.g. movie). In some cases, such information can prove to be very scarce, especially if the movie is of low quality and users simply do not bother to rate it. In contrast users may discuss, or exchange impressions about the movie, hence the challenge is to be able to detect the opinion about the products and use it in a recommendation system.

These challenges reside at the core of my main research objectives: investigate the lexical characteristics of opinionated texts. In the following section, I discuss the specific objectives of my research.

1.2 Research Objectives

The broad objective of the research proposed in this thesis is to investigate the extraction of sentiment lexicons and the use of sentiment analysis techniques for reputation analysis and recommender systems. Once the challenges are identified I will focus on how to address the problem with the proposed approaches.

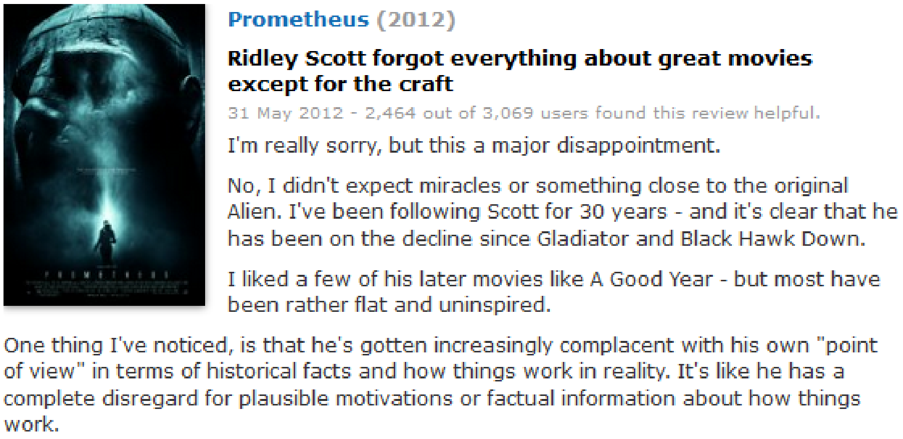
In the context of the analysis of online user reviews: Is it possible to effectively extract sentiment words? More specifically, identify their polarity (positive or negative) and sentiment strength. Also, characterize sentiment words in terms of their sentiment distribution: compute the polarity and sentiment strength fluctuation in respect to different sentiment levels. How to use automatic probabilistic methods with no human annotation to analyse user reviews? What is the most appropriate method to automatically identify sentiment words and their strength? To tackle these questions our first objective can be summarized as:

> *Objective 1: Apply probabilistic techniques to extract sentiment words from online reviews. Departing from the more traditional positive or negative representation, characterize sentiment words in terms of their sentiment distribution.*

Beyond simple words, we also question if it is possible to infer the sentiment of named entities in specific domains? Can the entity sentiment value be used to infer its reputation? In this context, can we visualize in a graph of sentiment the relations between entities and sentiment words? This questions leads us to the second objective of this thesis:

> *Objective 2: Predict the reputation of entities by investigating in a sentiment graph the sentiment words and entities sentiment relations and co-occurrence probability using propagation algorithms.*

Exploiting sentiment relations to improve sentiment tasks has caught the interest of recent research (Calais Guerra et al., 2011; Hu et al., 2013; Tan et al., 2011). For example in Figure 2 – a user review about the movie Prometheus – we observe numerous sentiment words relations with many entities (e.g. *Alien* and *Gladiator*). To express their opinions users apply different sentence syntactic constructions styles, and comparative sentences are frequently observed. In comparative sentences users tend mention other entities to establish a relation.



Prometheus (2012)

Ridley Scott forgot everything about great movies except for the craft

31 May 2012 - 2,464 out of 3,069 users found this review helpful.

I'm really sorry, but this a major disappointment.

No, I didn't expect miracles or something close to the original Alien. I've been following Scott for 30 years - and it's clear that he has been on the decline since Gladiator and Black Hawk Down.

I liked a few of his later movies like A Good Year - but most have been rather flat and uninspired.

One thing I've noticed, is that he's gotten increasingly complacent with his own "point of view" in terms of historical facts and how things work in reality. It's like he has a complete disregard for plausible motivations or factual information about how things work.

Figure 2. Different entities are mentioned as domain specific quality-references.

After these, more fundamental research questions are addressed, we studied how we could enhance a recommendation framework with sentiment analysis techniques. More specifically, we are interested in questions such as: how can we improve a recommendation system with sentiment analysis algorithms? Can we effectively obtain a rating from the analysis of user reviews? Is this rating able to be used in a recommendation system as if it was a rating explicitly given by a user? Thus, we wish to integrate the output of the two first objectives into a closing objective:

> *Objective 3: Investigate two recommendation system problems: first, techniques that embedded sentiment based ratings (Objective 1) in a recommendation system algorithm; and second, apply entities reputation analysis (Objective 2) in a recommendation system.*

Each one of these objectives will be addressed in a different chapter. In the following section we discuss the organization of this thesis.

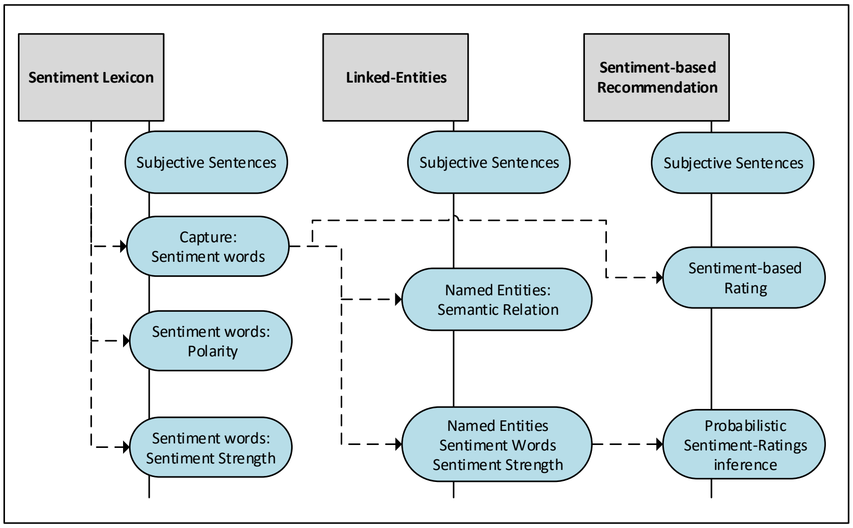

Sentiment Lexicon
Linked-Entities
Sentiment-based Recommendation
Subjective Sentences
Capture: Sentiment words
Sentiment words: Polarity
Sentiment words: Sentiment Strength
Subjective Sentences
Named Entities: Semantic Relation
Named Entities Sentiment Words Sentiment Strength
Subjective Sentences
Sentiment-based Rating
Probabilistic Sentiment-Ratings inference

Figure 3. Process followed in the development of this thesis.

1.3 Research Organization and Contributions

The research conducted in the context of this thesis is organized as depicted in Figure 3. The initial research is centred in understanding state-of-art sentiment analysis algorithms (*Objective 1*). In doing so, we investigated sentiment analysis classification tasks and performed an evaluation of sentiment analysis techniques. This evaluation (Peleja and Magalhães, 2013) has thoroughly examined the PMI-IR technique (Turney, 2002) and the sentiment lexicon SentiWordNet (Esuli and Sebastiani, 2006). We found that spam reviews contain specific domain words which influence the algorithms performance, and that a major challenge in opinion retrieval is the detection of words that express a sentiment and domain related idiosyncrasies where sentiment words are common, i.e. jargon. These experiments were reproduced, reviewed in **chapter 2 – Related Work** and reported at:

> Peleja, F. and Magalhães, J. 2013. "Opinions in User Reviews: An Evaluation of Sentiment Analysis Techniques." *In EPIA 2013 - Local Proceedings of the 16th Portuguese Conference on Artificial Intelligence*, pages 468–79. Angra do Heroísmo, Portugal. doi:10.13140/2.1.3177.1206.

Next we explored a novel method to automatically capture sentiment vocabularies (Peleja and Magalhães, 2015). The idea is to propose a method that without any manual annotation is able to capture and characterize the sentiment distributions of both generic and domain specific sentiment words. This research is detailed in **chapter 3 – Sentiment-Ranked Lexicons**, and the contributions of this chapter were published in the following papers:

Peleja, F. and Magalhães, J. 2015. "Learning Sentiment Based Ranked-Lexicons for Opinion Retrieval." In *Proceedings of the 37th European Conference on Advances in Information Retrieval (ECIR)*, pages 435-440, Vienna, Austria: Springer. doi: 10.1007/978-3-319-16354-3_47.

Peleja, F. and Magalhães, J. 2015. "Learning Ranked Sentiment Lexicons" In *Proceedimgs 16th International Conference Computational Linguistics and Intelligent Text Processing (CICLing)*, pages 35-48, Cairo, Egypt: Springer. doi: 10.1007/978-3-319-18117-2_3.

Peleja, F. and Magalhães, J. "Learning Ranked Sentiment Lexicons for Opinion Retrieval," *Information Retrieval Journal (under review).*

In a second phase we explored the sentiment expressed by some entities. Entities enclose a sentiment that we believe to be associated to their reputation value (*Objective 2*). In a three step procedure we perform a reputation analysis of domain entities. First, our method extracts the sentiment distribution of entities; second, a sentiment graph is created by analysing cross-citations in subjective sentences; and third, entities reputation are updated through an iterative optimization that exploits a graph of linked entities. The graph is represented in a pairwise Markov Network and represents relations existing in the corpus. This work was presented in Peleja et al. (2014b), Peleja et al. (2014a) and Peleja (2015). This research is detailed in **chapter 4 – A Linked-Entities Reputation Model**, and the contributions of this chapter were published in the following papers:

Peleja, F., Santos, J. and Magalhães, J. 2014. "Reputation Analysis with a Ranked Sentiment-Lexicon." In *Proceedings of the 37th International ACM SIGIR Conference on Research Development in Information Retrieval*, pages 1207–10. SIGIR '14. Gold Coast, Australia: ACM. doi:10.1145/2600428.2609546.

Peleja, F., Santos J. and Magalhães, J. 2014. "Ranking Linked-Entities in a Sentiment Graph." In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 118–25. Warsaw, Poland: IEEE Computer Society. doi:10.1109/WI-IAT.2014.88.

Peleja, F. 2015. "PopMeter: Linked-Entities in a Sentiment Graph." In *Proceedings of the 37th European Conference on Advances in Information Retrieval (ECIR)*, pages 785-788, Vienna, Austria: Springer. doi: 10.1007/978-3-319-16354-3_85.

Finally, we extended state-of-the-art recommender techniques by combining explicit ratings with sentiment ratings in a recommender system (*Objective 3*). With this approach we intended to broaden the usual scope of collaborative recommender systems which focus mainly on explicit ratings. The goal is to use sentiment analysis algorithms to compute more realistic and unbiased user ratings (Peleja et al., 2012; Peleja et al., 2013). This research is detailed in **chapter 5 – Sentiment Analysis Applications**, and the contributions of this chapter were published in the following papers:

Peleja, F., Dias, P. and Magalhães J. 2012. "A Regularized Recommendation Algorithm with Probabilistic Sentiment-Ratings." In *Proceddings on the IEEE 12th International Conference on Data Mining Workshops (ICDMW/SENTIRE)*, pages 701–8. Brussels, Belgium: IEEE Computer Society. doi:10.1109/ICDMW.2012.113.

Peleja, F., Dias P., Martins, F. and Magalhães J. 2013. "A Recommender System for the TV on the Web: Integrating Unrated Reviews and Movie Ratings." *Journal of Multimedia Systems, Springer-Verlag New York*, Volume 19, Issue 6, pages 543–58. Springer. doi:10.1007/s00530-013-0310-8.

Santos, J., Peleja, F. and Magalhães, J. "Monitoring Social-Media for Cold-Start Recommendations", *Multimedia Tools and Applications, Special Issue on Immersive TV*. (*under review*).

Besides the scientific contributions stated above, I worked in turning this state-of-the-art research into industry innovation. A deep analysis of customer preferences allow recommender systems to profile domain-specific linguistic traits and compute the reputation of popular entities. Such extracted information enables several services that engage users to improve their social interaction in a social-media context. I have submitted these ideas to an industry innovation competition and was awarded the first prize:

Peleja, F. 2014. "Social NOS." First prize in DevDays 2014. NOS in collaboration with Microsoft set a challenge to university students – what it will be like "The Television of the Future".

1.4 Thesis Outline

This document follows the following structure:

- **Chapter 2 - Related work**: This chapter reviews state-of-the-art research methods. It also reproduces a set experiments where we examine and discuss several relevant methods.

- **Chapter 3 - Sentiment-ranked Lexicons**: In this chapter we propose a model to learn domain specific sentiment ranked lexicons.
- **Chapter 4 - A Linked-Entities Reputation Model**: In this chapter we extend the sentiment lexicons by identifying entities and model their popularity with sentiment analysis algorithms. Also, a visualization tool is described to examine domain-specific entities in terms of their popularity and relations.
- **Chapter 5 – Sentiment-based Recommendation**: The two previous chapters provide key tools to improve recommender systems. In this chapter we show how recommender systems can be improved with sentiment analysis techniques.

2

Related Work

The research described in this thesis is concerned with the analysis of online reviews. Reviews influence user opinions about products and have a direct impact on product sales and reputation. In this context, research in sentiment analysis started as a field of study that is mainly interested in the analysis of user opinions about products, people and services. Linguistic techniques to process natural language texts (NLP) have a long history, however sentiment analysis research has mainly started in the early 2000s. The strategic importance of monitoring emergent comments that influence products reputation has captured the attention of the research community and e-commerce companies (Martín-Wanton et al., 2013). Consequently sentiment analysis has grown to be a very active research area (Liu, 2012). Throughout this document we will refer to user opinions as reviews or comments.

2.1 Sentiment Analysis

NLP is a field of computer science concerned with the problem of understanding the meaning of a sentence or a document written in natural language. NLP challenges involve natural language understanding and for this reason it is strongly related with sentiment analysis – the central topic of this thesis. In general sentiment analysis applies natural language processing techniques to capture subjective information. Hence sentiment analysis is a NLP research topic that covers many other challenges, as it will be discussed later in this chapter. Although research on NLP has strong roots, only after 2000 has sentiment analysis grown to become one of its most active areas (Turney, 2001; Turney, 2002; Pang et al., 2002; Liu, 2010).

General Inquirer (Stone and Hunt, 1963) is a system developed for content analysis research and it is one of the first introduced methods that among many different NLP tasks aimed at distinguishing between subjective and objective content. Only much later has Hatzivassiloglou and McKeown (1997) proposed a method to identify the positive and negative semantic orientation of adjectives. This was probably the first published work in sentiment analysis. In the early 2000 the burst of social media information led to the development of many other techniques to solve sentiment analysis problems (Turney and Littman, 2003; Turney and Littman, 2002; Turney, 2001; Dave et al., 2003; Hatzivassiloglou and Wiebe, 2000; Das and Chen, 2001). One important aspect of sentiment analysis tasks is that not every word expresses a sentiment, and a common approach is to identify sentiment bearing words by observing the respective words' family. Sentiment bearing words are known as *opinion words*, *polarity words*, *opinion-bearing words* and *sentiment words*.

Figure 4 presents some of the topics related to sentiment analysis that will be discussed in this chapter: sentiment analysis tasks, the most common used sources, and approaches used by researchers to solve this type of research problems.

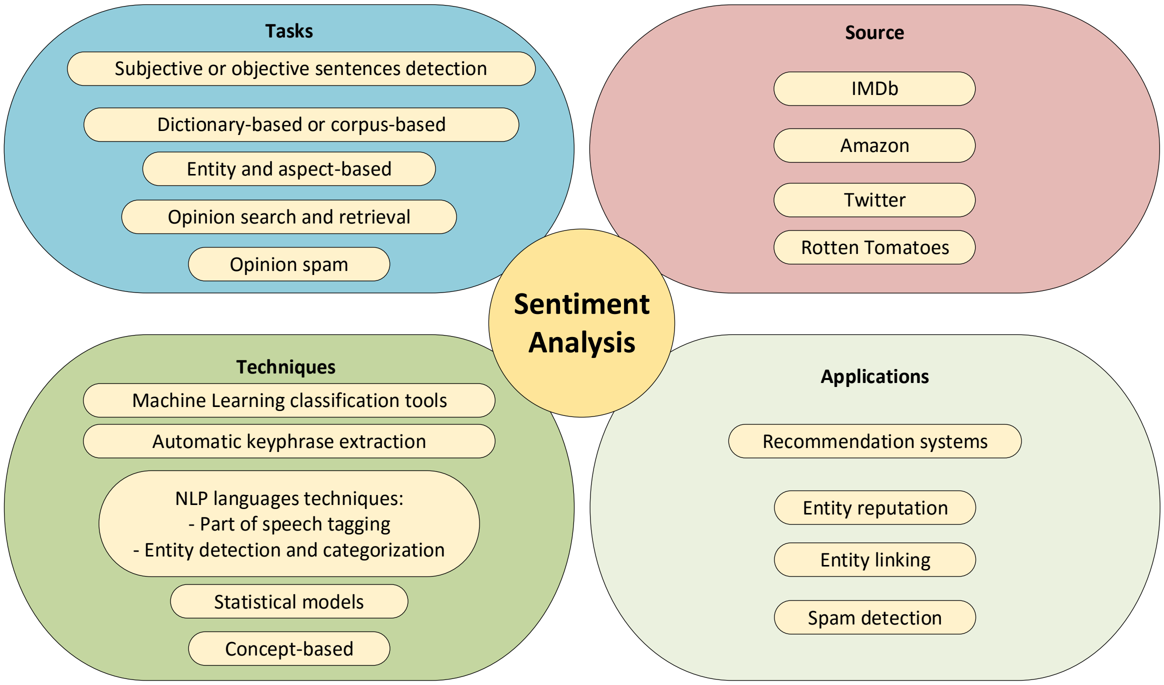


Figure 4. Graphical representation of the sentiment analysis tasks, source and techniques.

2.1.1 Granularity Levels of Sentiment Analysis

Initial work in sentiment analysis aimed at identifying overall positive or negative polarity within full documents (e.g. reviews). Later, works identified that sentiment does not occur only at document-level, or is limited to a single valence or target (Cambria, 2013). Hence, sentiment analysis has been investigated at four granularity levels: document, sentence, word and entity or aspect. Usually entity or aspect level involves extracting product features that are used to express an opinion (Hu and Liu, 2004b; Hu and Liu, 2004a; Popescu and Etzioni, 2005). Finding semantic orientation at word or phrase level differs from entity or aspect level as it is related to specific word families that are mostly used to express a sentiment. At word level researchers have mainly used two methods to automatically annotate sentiment: dictionary-based and corpus-based (Figure 5). Others have also chosen to manually annotate at word level, however, relying in a manual approach is highly time consuming and subjective (Liu, 2012; Ding et al., 2008).

In comparison to document and sentence level, the entity or aspect level allows a finer-grain analysis (Liu, 2010). The latter involves extracting product features that users dislike and like (Hu and Liu, 2004a), while document- or sentence- level sentiment words are commonly used in the task of predicting sentiment classes for users' opinions (Liu, 2012). LDA-based models are considered state-of-the-art for aspect-based sentiment analysis in which topic models have proved to be successful when applied to online reviews such as IMDb or TripAdvisor reviews (Moghaddam and Ester, 2012; Lim and Buntine, 2014). On the document level approach a state-of-the-art approach was introduced by Turney and Littman (2002) who implemented an unsupervised learning algorithm to evaluate review's polarity. For each review, the authors compute the average polarity of its constituent words or phrases. Other works (Pang et al., 2002; Heerschop et al., 2011) have also addressed the sentiment analysis task by using a document-level approach. A common use of sentence-level sentiment analysis is to capture opinionated sentences (Wiebe et al., 1999). To this end, the goal is to distinguish between sentences that express factual information (objective) and sentences that express an opinion (subjective) (Hatzivassiloglou and Wiebe, 2000). Even so, these three levels of granularity require an understanding of "how and which" words express human preferences.

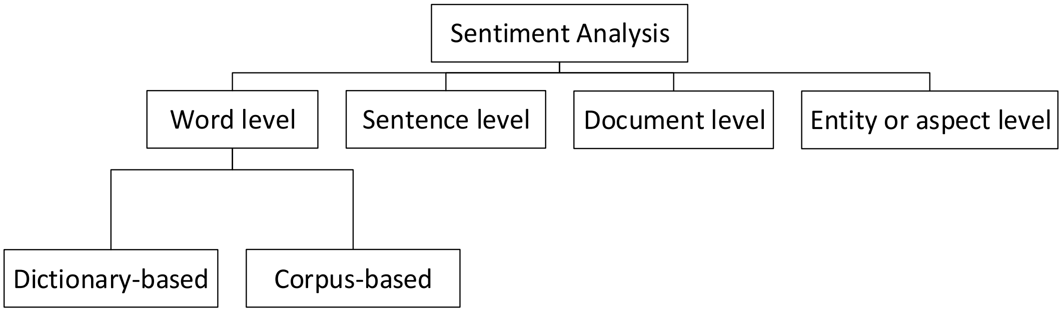


Figure 5. Sentiment analysis granularity levels.

2.1.2 Dictionary-based

Dictionary-based approaches are the most straightforward approaches to obtain a sentiment lexicon. These methods only use a seed list of words or use them in a bootstrap approach to discover new words (Liu, 2012). The strategy is to use a thesaurus or lexical database (e.g. WordNet) as a seed list of words (Ding et al., 2008). A common assumption in such techniques is that semantic relations transfer sentiment polarity to associated words (Kamps et al., 2004; Hu and Liu, 2004a). For instance, using the synonyms semantic relation the sentiment word "lovely" will transfer its positive polarity to its synonyms "adorable", "pretty" etc. (Bross and Ehrig, 2013). Others have chosen to use pre-compiled lists of sentiment words with similar techniques (Hu and Liu, 2004a). The previously pre-compiled lists are known as sentiment lexicons. Previous work has made available to the research community numerous sentiment lexicons: SentiWordNet[1], General Inquirer[2], Urban Dictionary[3], Twitrratr [4] and Multi-perspective Question Answering (MPQA)[5]. A sample of generic positive seed of words can be words such as "good", "nice" and "excellent" and a negative set contain words such as "bad", "awful" and "horrible". Usually dictionary-based methods observe the sentiment word occurrence, and by observing the words proximity its influence towards other words. Moreover, takes into account negation and/or neutralization tokens. The scope of negation aims to detect polarity changes and neutralization, overriding the sentiment polarity effect. Indications of these tokens are words such as "not",

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

[2] http://www.wjh.harvard.edu/~inquirer/

[3] http://www.urbandictionary.com/

[4] https://twitter.com/twitrratr/

[5] http://mpqa.cs.pitt.edu/

“although”, “never” and “would”, “should” and “hope” for negation and neutralization respectively.

A simple and effective dictionary-based approach that is well-known by the community was proposed by Turney and Littman (2003). Here, a seed of words are manually selected as paradigms of positive and negative semantic orientation and applied the Pointwise Mutual Information (PMI) method. PMI has been previous proposed in Turney (2002) and is used to infer semantic orientation from semantic association.

In Hu and Liu (2004a) the authors proposed an iterative process that expands an initial seed of words. The bootstrapping method uses a small list of manually annotated adjectives with positive and negative labels. WordNet list of synonyms and antonyms are used to grow the initial seed of adjectives. Others (Baccianella et al., 2010; Valitutti, 2004) have also used relationships between terms in WordNet to expand positive and negative seed sets. In comparison Rao and Ravichandran (2009) proposed a more elaborated approach. Here, the authors use a three graph-based semi-supervised learning method to identify semantic orientation: Mincut (Blum and Chawla, 2001), Randomized Mincut (Blum et al., 2004) and label propagation (Zhu and Ghahramani, 2002). Similarly, to the aforementioned approaches, Rao and Ravichandran (2009) employed a dictionary-based approach to detect sentiment words by exploiting WordNet synonyms graph. A disadvantage of dictionary-based approach is that they are dependent on pre-built lexicons or manually selected seed of words. Such lexicons or list of words tend to limit the sentiment words scope and as a consequence does not allow to identify domain dependent sentiment words (i.e., jargon).

2.1.3 Corpus-based

Corpus-based approaches have proven to be more successful than using pre-built sentiment dictionaries: as observed by Aue and Gamon (2005) dictionaries usually fail to generalize. Corpus-based approaches can be split into two main groups: (1) using a seed list of sentiment words, usually from a pre-built sentiment dictionary, and later for a specific domain corpus the sentiment word list is used to learn other sentiment words; and (2) implement a method to obtain a sentiment word lexicon for a specific domain corpus. In (1) a straightforward approach is to extract sentiment words through the proximity to an initial seed of words (Liu, 2012). In an early approach, Hatzivassiloglou and McKeown (1997) used a seed of adjectives with a set of linguistic constraints to capture sentiment words. Later, Turney and Littman (2003) and Turney (2002) used Hatzivassiloglou and McKeown (1997) list of adjectives and the General Inquirer dictionary to perform sentiment classification analysis. In Turney (2002) sentiment phrases were captured by evaluating the proximity to an adjective or a verb. In this context Turney and Littman (2002) reported a study where sentiment classification using only adjectives as sentiment words improves the classifier performance. Nonetheless, previous work (Esuli and Sebastiani, 2005; Heerschop et al., 2011; Takamura et al., 2005; Turney and Littman, 2003) have also shown that other word families such as adverbs, nouns and verbs are also qualified with sentiment intensity.

Bethard et al. (2004) devised a supervised statistical classification task to distinguish between opinionated (subjective) and factual documents. With the purpose of obtaining a sentiment lexicon, subjective documents were used to compute the words' relative frequency. The authors used a pre-built lexicon – a seed list of 1,336 manually annotated adjectives (Hatzivassiloglou and McKeown, 1997) – and computed the sentiment lexicon with a modified log-likelihood ratio of the words' relative frequency. Later, Qiu et al. (2009) proposed to obtain a domain specific sentiment lexicon by using an initial seed of sentiment words in a propagation method. Here, the authors used words from a sentiment lexicon as seed in a sentiment word detection process that iterates until no new sentiment words are added to the lexicon. The process detects sentiment words by observing its relation to the initial seed of sentiment words and later the newly extracted sentiment words are used to detect more sentiment words. For this work the authors explored syntactic relations between sentiment words and features. This technique contrasts with Hu and Liu (2004b) who proposed to extract features with distance-based rules. However, as Qiu et al. (2009) comment, Hu and Liu (2004b) proposed a method to detect product features not to expand a sentiment word lexicon.

Peng and Park (2011) proposed to extract a sentiment word lexicon that encloses informal and domain-specific sentiment words. Their technique exploited a matrix factorization method where each entry is the edge weight between two sentiment words. For this task the authors used WordNet relations and conjunction relations to calculate words proximity to WordNet synonyms and antonyms. WordNet synonym and antonym relations has also been the starting point for Kim and Hovy (2004) and Baccianella et al. (2010) SentiWordNet sentiment lexicon. In another approach, Chen et al. (2012) explored the usage of slang and domain-specific sentiment words to extract sentiment expressions from unlabelled tweets. For this task, they used Urban Dictionary in a target-dependent strategy. Urban Dictionary and Twitrratr are dictionaries that contain sentiment words that do not exist in more generic sentiment dictionaries (e.g. MPQA (Wiebe and Cardie, 2005), General Inquirer (Stone et al., 1966) and SentiWordNet (Esuli and Sebastiani, 2006)).The authors used 3,000 tweets and two groups of annotators to manually evaluate the quality of the obtained sentiment expressions. Then, these sentiment expressions are compared with gold standard sentiment dictionaries – MPQA, General Inquirer and SentiWordNet.

The aforementioned corpus-based methods are constrained to the initial seed of sentiment words. Even though for a particular domain a partial number of sentiment words that are used as seed may not reflect the accurate sentiment strength and orientation, their sentiment information is used to help detect new sentiment words. For example, the word "Oscar" can relate the Hollywood Movie's Academy award (implying a positive sentiment) or relate to the given name of a person. An alternative to the abovementioned approaches is to compute algorithms that fully identify sentiment words automatically i.e. in a supervised manner (Jiang et al., 2011; Pang et al., 2002; Pang and Lee, 2005).

Pang et al. (2002) introduced one of the early works to perform a sentiment classification of movie reviews. Here, the authors used well-known machine learning classifiers in a topic-based text categorization strategy. Pang et al. (2002) proposes a corpus-based strategy whereas no initial seed of sentiment words is used. Later, Jiang et al. (2011) establish an important distinction between their work and Pang et al. (2002) work – target-independent strategies may assign irrelevant sentiments to a given target. Hence, Jiang et al. (2011) argue that a sentence where a sentiment is expressed does not contain necessarily the feature that is the target of the sentiment expressed by the user. They additionally point-out the need to observe the review content and its domain. One downside of using a supervised learning task in a corpus-based method over a dictionary-based method, is that corpus-based methods require a greater effort to generalize into different domains (Aue and Gamon, 2005; Pang and Lee, 2008). Nonetheless, from previous contributions, is important to highlight the importance of capturing relevant sentiment words that are strongly associated to a given context (Liu, 2012).

Markov conditional random fields, a class of undirected graphical models, belong to the discriminative family of approaches. This model is trained to maximize the conditional probability of observing a text sequences, leading to a lower number of possible combinations between an observed word and their labels, and for this reason allows to represent better the text in the model. Therefore, Markov conditional random fields became quite popular in natural language processing approaches, and as a consequence in sentiment analysis tasks. Yang and Cardie (2012) proposed to extract sentiment expressions with a semi-Markov conditional random fields (semi-CRFs). This probabilistic approach for segmenting sentiment words has the advantage of being able to detect segment combinations – unigram or N-grams – which provides a better modelling of users' opinions. In this work, the authors study the impact of using syntactic structure segments and syntactic features for capturing opinion expressions. Hence, the authors show that taking into account the syntactic structure helps in the task of detecting opinion expressions but with the weakness of the computational cost associated to parsing all the data.

There are also several works that uses the LDA (Latent Dirichlet Allocation) generative process (Blei et al., 2003) in NLP tasks. LDA has been previously used for a variety of NLP related tasks and has proven to be adequate and hold great results (Harvey et al., 2010; Kang et al., 2013; Blei and McAuliffe, 2007; Zhang et al., 2007). Blei and McAuliffe (2007) propose supervised LDA (sLDA) to predict ratings for movie reviews, this is done by using labelled documents. Reasoning on the rating scale that reflects how much a user liked or disliked a specific movie, in which, ratings represent a sentiment scale, this topic classification task could be seen as a supervised sentiment classification task. Zhang et al. (2007) proposed an LDA based hierarchical Bayesian algorithm, named SSN-LDA. This algorithm is used to discover communities in social networks and the respective associated researchers. Later, Harvey et al. (2010) used users' bookmarking in a LDA hidden topics model to improve page ranking. More recently, Kang et al. (2013) proposed LA-LDA to create user models based on the analysis of social network friends. The authors argue that their method can be useful to capture information related to similar users. Then, automatically filter a chunk of information that might not be useful for a specific user. Hence, is noticeable the ability of this method to produce a set of concepts that are related, not only for topics but also for user preferences.

Lin and He (2009) proposed a fully unsupervised probabilistic model based on LDA, which they named as joint sentiment/topic model (JST). JST model detects sentiment and topics from documents. The authors notice the surprising fact that, the algorithms performance did not improve upon enriching the obtained sentiment words with words from sentiment dictionaries. In fact, it lowered the performance of their method. Their results show that in order to JST achieve its best results, the algorithm required a pre-defined list of sentiment words. Later, Jo and Oh (2011) proposed to apply an LDA generative model to a sentiment analysis problem. Here, the authors propose a method to extract the sentiment pairs {*aspect*, *sentiment*}, where *aspect* refers to product features. To evaluate the obtained pairs the authors had to manually select a list of sentiment words, as for the sentiment classification task the authors chose to perform a binary supervised sentiment classification. For the classification task each review was labelled as positive or negative according to a given probability.

2.1.4 Subjectivity Sentences

Subjectivity in natural language refers to certain combinations of the language used to express an opinion Liu (2010). Early work by Wiebe (1994), defines subjectivity classification as an algorithm that evaluates in a sentence or document the linguistic elements that express a sentiment – sentiment words or, as Wiebe denotes, *subjective elements*. In other words, objective and subjective sentences can be defined as "*An objective sentence expresses some factual information about the world, while a subjective sentence expresses some personal feelings or beliefs.*"(B Liu, 2010). That is, subjectivity in natural language refers to certain combinations of the language that are used to express an opinion (Wiebe, 1994; Tang et al., 2009). It is common to apply a sentiment classifier to evaluate the sentiment polarity and/or strength of sentences labelled as subjective (Liu, 2010). This classification allows to differentiate factual and subjective sentences and for this reason, is commonly known as subjectivity classification. In addition, subjectivity classification has been extensively investigated in the literature (Hatzivassiloglou and McKeown, 1997; Hatzivassiloglou and Wiebe, 2000; Riloff et al., 2006; Riloff and Wiebe, 2003). In the task of creating a sentiment lexicon, subjectivity classification can prove to be a very important step: we should observe sentences that express an opinion (subjective) and ignore sentences that state a fact (objective).

Hatzivassiloglou and Wiebe (2000) claim that adjectives are strong indicators of subjective sentences. Their method uses adjectives to detect potential subjective sentences. Previous work by Wiebe et al. (1999) had a similar method but instead of adjectives, also used words from the family of nouns. More recently, Wiebe and Riloff (2005) introduce a bootstrapping method that learns subjective patterns from unannotated documents. For this method, one needs to define an initial set of rules that are manually annotated and to this aim require a linguist expert (Scheible and Schütze, 2012). Riloff et al. (2006) has also proposed a method that defines subsumed relationships between different elements (unigrams, n-grams and lexicon-syntactic patterns). The idea is that, if an element is subsumed by another, the subsumed element is not needed, thus, can remove redundant elements in the subjectivity classification (Bing Liu, 2012).

In data mining, or machine learning, a classification task uses prior knowledge (e.g. documents or reviews) as training data to learn a model to automatically classify new data. In this context, Pang et al. (2002) claim that machine learning classification methods work well in sentiment analysis tasks. The authors claim that supervised learning fits a sentiment classification task as in a document-level classification. However, one should keep in mind that these models are highly dependent on the quality of the training data. Other researchers have also proposed sentiment classification algorithms for the subjective and sentiment classification problem (Yu and Hatzivassiloglou, 2003; Turney, 2002; Hu and Liu, 2004a; Kim and Hovy, 2004).

Lourenco Jr. et al. (2014) propose an online sentiment classification method and argue that previous approaches lean towards offline classification. This is a critical point that is addressed by the authors – in their approach it is required to produce tweet sentiment judgements in real-time. To this end, an alternative classification strategy is proposed by the authors: to ensure a fast learning time the training sets are kept as small as possible. To this aim the authors describe a set of association rules that are used for sentiment scoring. More formally, a set of rules built from the vocabulary training set $\mathcal{D}_n$ are used to define a classifier $\mathcal{R}(t_n)$ at each time step n. For a given message t_n a rule is valid if applicable to the respective message. Nevertheless, opinions context tend to drift over time and the quality of rules coverage require a reasonable amount of work which might require maintenance rules. As Wiebe and Riloff (2005) notice, rule-based classifiers do not involve learning but merely classify sentences by observing state-of-the-art polarity characteristics that have been previously published. On the other hand in Lourenco Jr. et al. (2014) work at each time step n the classifier updates the vocabulary and sentiment drifts (e.g. polarity changes for the same entity) which slightly differs from traditional rule-based approaches.

2.2 Reputation Systems

Reputation systems address the welfare of e-business communities and individual participants. These systems facilitate decision making hence its importance encourages the community to understand its components and processes (Standifird, 2001). On last years the potential marketing usefulness of reputation analysis has led research to focus extensively on monitoring and profiling relevant issues for market brands and organizations on Twitter, such as Apple and Windows (Villena-Román et al., 2012; Martín-Wanton et al., 2012; Spina et al., 2013; Martín-Wanton et al., 2013).

A reputation system collects and aggregates feedback about users' past behaviours. These systems help users decide who to trust and as a consequence, encourage trustworthy behaviours. In this context trust can be defined as *"a subjective expectation an agent has about another's future behaviour based on the history of their encounters."* (Mui et al., 2002). Reputation systems provide a trust environment for organizations or individuals and discourages the use of reputation systems that have dishonest past behaviour. Resnick et al. (2000) consider reputation to be a community opinion of a particular organization or individual. Reasoning on this concept, a trust environment is built based on the will of each individual to trust. These actions generate a chain of events, indicating if the organization or individual is trustworthy. Hence, a reputation system can be defined as a system based on participants (organizations or individuals), where their behaviour assigns a reputation to participants.

The aforementioned definition for reputation systems excludes the possibility of referring to a reputation system as a collaborative filtering system. To predict products that users were not aware of its existence a collaborative filtering system observes large communities of users that rate products, and then their preferences are matched against other users' preferences (Aciar et al., 2007). At scoring products, collaborative filtering systems assume that all products are trustworthy. Hence, these systems do not consider the reputation of the recommended products and by ignoring the product reputation the results might not match users' opinion. As Clausen (2003) argues, collaborative systems capture a sub-community that the user fits into but that is not based on recommended products reputation. Both systems use large communities to engage peer-review analysis, however collaborative systems are not bi-directional as they only observe communities rating (Rietjens, 2006).

2.2.1 Online Reputation

A well-known reputation system is eBay[6]. Founded in 1995 eBay is an online auction marketplace that allows users to purchase products and give feedback to each other. In this system the previously assigned feedback is used to calculate the reputation score, e.g. positive minus negative feedback. This is a simple system that does not take into account previous behaviours from other platforms. For example, a first-time user has the same reputation as a well-known product manufacture that sells thousands of products in another platform (Standifird, 2001; Rietjens, 2006). A few years after eBay launch, Page et al. (1998) proposed the popular PageRank reputation metric. PageRank calculates page recommendations by the source's incoming links. Pages with a high number of users trust votes are more likely to be recommended, this information propagates through the network. Google used PageRank scores to choose which pages should appear with higher relevance in the search results. More recently, Sabater and Sierra (2001) proposed a research reputation system that estimates the reputation of an individual by selecting the most appropriate individual to evaluate its relevance. This system aims at having one individual that is selected considering its interaction, conflict of interests and social structure.

[6] http://www.ebay.com

Although this system asserts the reputation based on a single individual, the authors claim they have a fairly good approximation to the general opinion about another specific individual. Another proposed reputation system is given by Mui et al. (2002). Here the authors propose a statistical method to assert the organization or individual reputation. This model computes the interactions between users and is not restricted to explicit ratings.

RepLab is a competitive evaluation exercise for online reputation management and RepLab 2013 made available a large collection of Twitter data for reputation monitoring. This data provides a reliable test collection for reputational polarity (Amigó et al., 2013). The collection contains tweets about 61 entities within four domains (automotive, banking, universities and music). The entities were manually chosen according to their inherently relation to the products. Hence the entities transparency and ethical side are highly affected by their products reputation. However, as Spina et al. (2014) notice, reputation monitoring tasks are usually fine-grained and suffer from data sparsity, with some exceptions for popular entities such as "Apple" or "Barack Obama". To automatically capture the relation between entities that have their reputation affected by associated products is a challenging task and very costly as a manual task (Spina et al., 2014).

2.2.2 Sentiment Influence on Online Reputation

Recent studies focused on the idea of exploiting sentiment relations to improve sentiment analysis tasks (Calais Guerra et al., 2011; Hu et al., 2013; Tan et al., 2011). Calais Guerra et al. (2011) analysed two events – presidential elections and national soccer league – which disseminates a large amount of opinionated comments in a social network such as Twitter. Similar to Spina et al. (2014) comments, Calais Guerra et al. (2011) emphasize that topics are not independent from opinion holders and the sentiment expressed. User comments might be influenced by external factors such as new entities or domain related sentiment words. Calais Guerra et al. (2011) perform a sentiment analysis task for Twitter users' comments in a transfer learning strategy. Here, the authors propose a framework that uses Twitter retweets to create a graph of transitive opinions. The authors observe that is possible to improve named-entities reputation when observing (during a period of time) named entities associated sentiment and the respective domain deviations. Other works, such as the ones from Hu et al. (2013) and Tan et al. (2011) have also used Twitter comments to discover transitive features. In the aforementioned works the argument is that it is possible to build a graph of users' relations based on comments analysis, also users tend to befriend with users with similar opinions.

Research efforts in reputation analysis have focused not only on summarizing the overall reputation, but also in predicting the reputation of other instances or events (Oghina et al., 2012; Joshi et al., 2010). Joshi et al. (2010) explored the popularity of old movies among online critic reviews to predict opening weekend revenues for new movies. For this task, the authors observed the metadata similarity between classic movies – highly rated –and recent movies. In a similar approach, Asur and Huberman (2010) exploited bursty keywords on Twitter streams to predict box-office revenue for movies. For this purpose, the authors study how positive and negative comments propagate in the social network and how influences people opinions about movies. More recently, Oghina et al. (2012) predicted IMDb movie ratings by performing an analysis over their popularity on social media, more specifically Youtube and Twitter. The authors investigate textual tweets, comments and likes that are associated to a specific movie. This analysis leverages on the movie reputation, which is then translated into ranking scale from 1 to 10. Unlike Calais Guerra et al. (2011) and Hu et al. (2013) methods Oghina et al. (2012) and Joshi et al. (2010) did not use graph methods to explore the influence of users comments over movies reputation. Both Oghina et al. (2012) and Joshi et al. (2010) perform a feature engineering task and use it in a linear regression algorithm. The authors (Oghina et al., 2012; Joshi et al., 2010) choose a diverse number of features in which several features are extracted from the product metadata, for example: number of views, number of comments, likes, favourites, genre, running time among many others. Regression analysis allows the authors of the aforementioned works to depict a relationship between independent and dependent variables in a graph and regression is a statistical process that is popular for its usage in forecasting tasks.

Martín-Wanton et al. (2012) explored different methods to identify relevant emerging topics that influence an organization reputation. Here, the content of each tweet was translated as a set of Wikipedia concepts and then, to capture relevant topics, applied to a LDA generative model. The authors performed a standard feature engineering task: term occurrence, TFIDF (term frequency-inverse document frequency), content-based, time-aware, and many others. More recently, in a similar task, Villena-Román et al. (2012) proposed to improve the reputation predictions with a generated domain-specific semantic graph. The semantic graph expands the sentiment word thesaurus, and this task can prove to be highly important to ascertain about entities reputation, since one should capture sentiment words that are highly attached to the domain (Liu, 2012).

2.3 Recommendation Systems

Recommender systems also known as recommendation systems tackle the problem of content overload. These systems emerged with the intent of obtaining personalized and meaningful recommendations based on user preferences and history. Although the increase of online information captured the attention of the recommender systems research community, these systems had their popularity peak in 2007 with the Netflix Prize contest[7]. This contest awarded $1M to the recommender algorithm with a minimum of 10% improvement.

Early work Resnick and Varian (1997) on recommendation systems defined these systems as: "*(…) people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients.*" The aforementioned definition describes recommender systems as supporting the collaboration between users. Later work expanded this definition to include systems that recommend products regardless of how the recommendations are produced (Burke, 2002).

Table 1 presents how Burke (2002) split recommendation approaches into five main techniques: collaborative, content-based, demographic, utility-based and knowledge-based. In Table 1, U is a list of known users, I is of list of known products (or items), u is a user that will receive recommendations and i the recommended product. Collaborative filtering systems aggregate user preferences history to provide new recommendations (Schafer et al., 2007). On the other hand content-based systems only observe and match user profiles (preferences) (Adomavicius and Tuzhilin, 2005). Demographic systems analyse recommendations based on demographic categorization (Pazzani, 1999). Utility-based systems provide their recommendations based on the user profile. In comparison to content-based, utility-systems have the advantage of using non-product attributes (e.g. product availability) in the recommendation computation (Guttman, 1998). Finally, knowledge-based systems evaluates the product requirements to provide a user recommendation. Hence knowledge-based systems learn how a particular product meets user's needs with functional knowledge (e.g. case-based reasoning) (Burke, 2007).

[7] www.netflixprize.com

Table 1. Recommendation systems techniques (Burke, 2002)

Techniques	**Background**	**Input**	**Process**
Collaborative filtering	Ratings from $\boldsymbol{U}$ of items in $\boldsymbol{I}$.	Ratings from $\boldsymbol{u}$ of items in $\boldsymbol{I}$.	Identify users in $\boldsymbol{U}$ similar to $\boldsymbol{u}$, and extrapolate from their ratings of $\boldsymbol{i}$.
Content-based	Features of items in $\boldsymbol{I}$.	$\boldsymbol{u}$'s ratings of items in $\boldsymbol{I}$.	Generate a classifier that fits $\boldsymbol{u}$'s rating behaviour and use it on $\boldsymbol{i}$.
Demographic	Demographic information about $\boldsymbol{U}$ and their ratings of items in $\boldsymbol{I}$.	Demographic information about $\boldsymbol{u}$.	Identify users that are demographically similar to $\boldsymbol{u}$, and extrapolate from their ratings of $\boldsymbol{i}$.
Utility-based	Features of items in $\boldsymbol{I}$.	A utility function over items in $\boldsymbol{I}$ that describes $\boldsymbol{u}$'s preferences.	Apply the function to the items and determine $\boldsymbol{i}$'s rank.
Knowledge-based	Features of items in $\boldsymbol{I}$. Knowledge of how these items meet a user's needs.	A description of $\boldsymbol{u}$'s needs or interests.	Infer a match between i and $\boldsymbol{u}$'s need.

2.3.1 Ratings-only recommendations

Recommendation algorithms have proven their ability to influence user's future purchases by observing the available user ratings. Two popular families of recommendation algorithms are content-based filtering and collaborative filtering, and hybrid approaches that combine content-based and collaborative filtering.

Content-based filtering aims at performing an analysis of the users' personal information and product preferences. Therefore, this type of analysis originally began in text processing applications and information retrieval (Belkin and Croft, 1992). Content-based filtering approaches have two main short comings: first, makes the assumption that similar users like the same products, and users who consumed a given product are willing to consume similar products; the second short coming concerns a limitation known as *overspecialization* (Adomavicius and Tuzhilin, 2005). *Overspecialization* lies in the fact that users are restricted to get only recommendations of products with similar characteristics of those they have consumed.

Collaborative-filtering (CF) attempts to infer implicit ratings based on the pattern analysis of user preferences and consuming history. An early application with CF was introduced by Resnick et al. (1994) which aimed at filtering netnews based on the ratings given by users. Hence, this approach introduced the concept of user explicit feedback in the form of ratings. More recently, Hu et al. (2008) provided a recommendation system that only relies on implicit feedback, thus, feedback obtained from users' activity analysis. Moreover, Koren (2008) has successfully proposed to blend explicit and implicit feedback in a CF approach.

2.3.2 Review-based recommendations

Sentiment analysis and recommendation systems (RS) have similar goals. Generally in sentiment analysis the main goal is to identify the users' likes/dislikes by evaluating the overall sentiment or specific feature oriented sentiment. In contrast, RS algorithms aim at learning users' likes to suggest new products. However, as Jakob et al. (2009) point out, most of RS algorithms focus on the explicit ratings and users/products characteristics disregarding the information enclosed in the free-text reviews. A few studies have proposed to integrate sentiment analysis with RS (Aciar et al., 2007; Jakob et al., 2009; Moshfeghi et al., 2011; W. Zhang et al., 2010).

Recommendations systems emerged with the intent of tackling the problem of choosing from a large set of products the sub-set of product(s) that provide more helpful recommendations. However, users provide comments on mostly everything, thus, the exchanged information goes beyond explicit ratings and past purchases. Also, several web applications only support comments (e.g. blogs and online forums). Moreover, traditional RS approaches that rely on a rating-only approach can prove to be an inadequate metric for assessing a user opinion about a product, and in some cases such information can prove to be very scarce. Hence, to handle the sparsity of ratings Y Moshfeghi et al. (2011) proposed to improve a RS algorithm by considering not only ratings but also emotions and semantic spaces to better describe the movies' and users' space. The Latent Dirichlet Allocation is used to compute a set of latent groups of users. Y Moshfeghi et al. (2011) evaluation showed that such hybrid approach (combining ratings with additional spaces extracted from metadata) outperforms ratings-only approaches and reduces the effects of cold-start.

In the movie domain Jakob et al. (2009) present the advantages of improving the RS quality with the sentiment extracted from user reviews. According to the authors, the sentiment should be split into clusters where each cluster corresponds to different movie aspects. Hence, the overall sentiment regarding a movie is measured by observing the sentiment within these clusters. In comparison to Jakob et al. (2009) approach where the recommendations always need explicit ratings, we propose to infer ratings from reviews. In addition, Jakob et al. (2009) use a semi-automatic clustering method to infer movie aspects upon which users express some opinion. Wang et al. (2012) propose a framework similar to Jakob et al. (2009). More specifically, in Wang et al. (2012) framework a CF recommendation algorithm is improved with an aspect-based sentiment analysis approach. Unlike Jakob et al. (2009) Wang et al. (2012) approach does not require the explicit rating to predict a sentiment-based rating. Additionally, with the use of two semantic spaces (movie and emotion) Wang et al. (2012) have successfully compared their approach with Y Moshfeghi et al. (2011). However, Y Moshfeghi et al. (2011) framework was evaluated in a larger set of semantic spaces, also in the evaluation dataset Wang et al. (2012) had a sample of 53.353 reviews whilst Y Moshfeghi et al. (2011) evaluated their framework in two datasets with 100.000 and 1 million ratings respectively.

Leung et al. (2006) suggested to infer ratings from user reviews and integrate them in a CF approach. The authors tackle the extraction of multilevel ratings by proposing a new method to identify opinion words, semantic orientation and its corresponding sentiment strength. This method allows different semantic orientation values to similar words. For example, the words *terrible* and *frightening* may seem similar but in some domains (e.g. movies), *frightening* is likely to be applied in a positive context. However, in contrast to the present work, Leung et al. did not perform any evaluation of the recommendation part, thus, having only assessed the opinion words sentimental strength and orientation.

In a more recent study Zhang et al. (2010) proposes a comprehensive approach to a sentiment-based recommendation algorithm on an online video service. Their system computes recommendations based on the analysis of users' reviews and textual facial expressions, and the video description and the respective comments. In Zhang et al. (2010) approach the inferred prediction is based on an unsupervised sentiment classification. For this task, a sentiment dictionary, an expression face set, and a negation word list is used to decide the sentiment polarity of each sentence. In addition, Zhang et al. (2010) identifies a list of keywords that are combined in an users' matrix, a products' matrix, and a ratings' matrix.

In a different study, Aciar et al. (2007) propose to analyse the user reviews by developing an ontology to translate users' reviews content. Aciar et al. (2007) presents an early work in a recommender system that uses the review text content. The ontology proposed by the authors relies on observing the review positiveness, negativeness and users' skill level. The related-word concepts allow the identification of co-related product characteristics (features). For instance, on the photographic cameras domain the concept "carry" is related to the concept of "size" (Aciar et al., 2007). However, an important part of their work relies on a manually created ontology that captures related-words and the training examples are manually collected and labelled. Consequently, Aciar et al. (2007) ontology measures the quality of the several features within a product to create the user recommendations.

In Table 2 we provide a comparative summary of some approaches in a review-based recommendation. In the table N.A. (not applicable) refers to a RS framework proposed by Leung et al. (2006) in which the experiments were not clearly introduced and not completed.

Table 2. Comparative summary of review-based recommendations frameworks.

	Requires Explicit Ratings	**RS experiments**	**Manually built opinion word lexicon**	**SA dictionary**	**Supervised SA**	**Unsupervised SA**	**Aspect based sentiment analysis**	**Semantic based sentiment analysis**	**Rating scale beyond *like* and *do not like***	**RS with a CF approach**
Wang et al. (2012)	yes	yes	no	yes	no	yes	yes	no	yes	yes
Moshfeghi et al. (2011)	yes	yes	no	no	yes	no	no	yes	no	no
Zhang et al. (2010)	yes	yes	yes	yes	no	yes	no	no	no	yes
Jakob et al. (2009)	yes	yes	no	no	no	yes	yes	yes	yes	yes
Aciar et al. (2007)	no	no	yes	no	no	no	no	yes	no	no
Leung et al. (2006)	N.A.	N.A.	no	no	yes	no	no	no	yes	yes

2.4 Experimental Comparison of Sentiment Classification Methods

SentiWordNet is a popular linguistic dictionary that was introduced by Esuli and Sebastiani (2006) and recently revised by Baccianella et al. (2010). This lexicon is created semi-automatically by means of linguistic classifiers and human annotation in which each synset is annotated with its degree of positivity, negativity and neutrality. Moreover the same synset can express opposite polarities. Previous studies using the sentiment lexicon SentiWordNet in sentiment classification tasks have shown promising results (Denecke, 2009; Ohana and Tierney, 2009). Ohana and Tierney (2009) used Support Vector Machine (SVM) classification and observed the distribution of the positive vs. negative opinion words in users' reviews. Their evaluation over a dataset of users' reviews from the Internet Movie Database (IMDb) website point out SentiWordNet as an important resource for sentiment analysis tasks. However, the best accuracy obtained with the authors approach is 69.35%. As will be seen in this survey, the results obtained with an alternative framework are considerable higher than the ones obtained by Ohana and Tierney (2009). Denecke (2009) for a sentiment analysis task have also applied the SentiWordNet sentiment lexicon. Here, the authors chose to evaluate the performance of a rule based classifier in three different domains: products, drugs and news articles. With an accuracy of 82% the authors show that the news articles domain presents the best performance over the remaining domains. However, one must keep in mind that, according to the domain there are considerable linguistic differences in the structure of users' reviews. The users' reviews from the movies domain are commonly written in natural language in which it is observable the usage of slang and internet acronyms. For example, spelling errors or writing styles (i.e. "greeeat" for the word "great") frequently occur in reviews from this domain. While a general linguistic resource such as SentiWordNet might not be able to capture the movie domain specific jargon, the same does not apply to news articles where text tends to be written in a more formal manner. Turney (2002) show that movie reviews prove to be more challenging than reviews about automobiles or bank. Here, the author obtained an accuracy of 80%, 84% and 66% for the automobiles, bank and movie domain respectively. However, for the author' reported results 120 movie reviews were used. For a better understanding of the difficulty of this task our survey performance a study using a larger corpora of movie reviews. As it will be observed, we report a better performance when using movie reviews *polarity*[8] dataset Pang et al. (2002) than with reviews from the books and music domain.

[8] A popular sentiment dataset that contains 2,000 movie reviews from IMDb.

The most elementary representation of an opinion word is the bag-of-words representation. Pang and Lee (2004) argues that this representation delivers fairly good results, in particular when comparing to bigram and adjective representation. Others, such as Liu (2010), stress that unigram representation simplicity might add a few doubts on its ability to describe a sentiment. For instance, the unigram representation might fail to capture strong opinions. Words from the words' family of adjectives are commonly observed paired with other opinion words. Riloff et al. (2006) reports a sentiment analysis study that combines a variety of representations: unigrams, multiword n-grams, phrases and lexicon-syntactic patterns. Here, a subsumption hierarchy is used to identify the opinion semantic orientation associated with each representation. Although Riloff et al. (2006) present a study free of opinion word sentiment lexicons (e.g. SentiWordNet) our comparative survey, for the *polarity* dataset, achieved a better performance.

For a sentiment analysis study an important step is the task of defining the sentiment word semantic orientation, in other words whether a sentiment word is positive or negative. Turney (2002) and Turney (2001) proposed a metric to estimate the orientation of a phrase using the concept of the PMI-IR (Pointwise Mutual Information-Information Retrieval). PMI-IR is known for its ability to measure words' semantic association strength. This metric measures the degree of statistical dependence between the candidate word and two reference words (i.e. positive and negative word reference). Turney argues that a high co-occurrence between a candidate word and a positive word is a suggestion of a positive sense. For example, high co-occurrence between "ice-cream" and the reference word "excellent". However, Turney's chosen reference words seems to some extent subjective (Mullen and Collier, 2004). To this end it is conducted a throughout evaluation of PMI-IR using different reference words.

In a sentiment classification task humans seem to be able to distinguish a positive from a negative feeling regardless their familiarization with the topic. Depending on the topic, for topic classification the same cannot be as easily said. Highly co-related topics can become a serious challenge, even for humans. In this context machine learning classification methods have been implemented having in mind a sentiment classification tasks (Pang et al., 2002; Pang and Lee, 2004). However, as Pang et al. (2002) argue machine learning techniques do not perform as well as in topic classification tasks. Although the noticeable similarities with topic classification a sentiment classification task requires a more comprehensive approach. In addition, Pang et al. (2002) argue that opinion words from the word family of adjectives provide less useful information than the unigrams (in which other words families are also considered). Nonetheless, considering the frequent usage of adjectives when expressing an opinion, in the present survey we combine adjectives with unigrams as a sentiment word bigram.

Initial studies in sentiment classification tackled the problem with binary classifiers (Liu, 2012). However, specific characteristics of different type of products or rating scales more closely related to the domain suggest multiclass classification (Sparling, 2011). Pang et al. (2002) proposed a binary classification approach. The authors analyse the performance of the binary sentiment classification for three well-known machine learning classifiers: Naive Bayes, maximum entropy and support vector machines (SVM). Prabowo and Thelwall (2009) argue on the advantages of using a rule-based classifier in semantic analysis. In comparison to the performance obtained by Prabowo and Thelwall (2009), in the present survey, the inductive rule-based classifier (RIPPER) presents competitive results. Similarly to Denecke (2009) observations the rule-based classifier accuracy is lower than the obtained with the logistic regression classification model.

2.4.1 Sentiment Analysis Framework

For the purposes of this survey it is implemented a sentiment analysis framework that aims to analyse users' reviews about different products (e.g. movie) and infer a preference in the form of ratings. To formulate the problem, a set of reviews and their associated rating $\mathcal{D} = \{(re_1, ra_1), \ldots, (re_n, ra_n)\}$ are analysed. A review re_i is rated according to the rating range of the dataset. For instance, in the Amazon dataset each review is rated with value of $ra_i \in \{1,2,3,4,5\}$. A review is represented by a set of opinion words $re_i = \left(ow_{i,1}, \ldots, ow_{i,m}\right)$ where each component $ow_{i,j}$ represents the opinion word j of the review i. The sentiment analysis framework aims to learn the following classification function,

$$\Phi(re_i) \mapsto [0,1], \qquad (1)$$

to infer the rating of a review. Following a machine learning approach, this function is learnt as a probabilistic model $p(ra_i|re_i)$ that is estimated from a labelled training set.

An overview of the sentiment analysis framework is shown in Figure 6. In this framework it is chosen a dictionary-based approach and the influence of negative and neutral expressions will be considered.

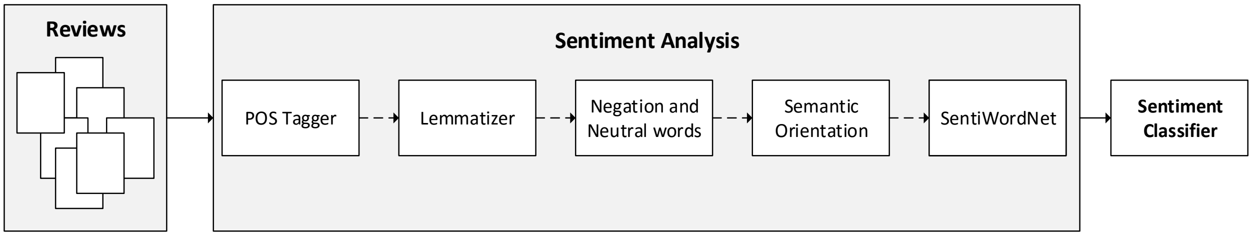


Figure 6. Overview of the Sentiment Analysis Framework

The most elementary representation of an opinion word is the single word (unigram). Pang and Lee (2004) argues that this representation presents fairly good results in relation to bigrams or adjectives-only representation. Considering the simplicity of the unigram representation we stress that this representation might fail to capture numerous opinion words (i.e. "basket case") Liu (2010). For that reason, two sentiment words representations were used: unigram and adjective-word pair (bigram). Regarding the bigram representation the following points were considered:

i. Adjectives influence the following word (s) by increasing and decreasing the level of positive or negative sentiment intensity.
ii. A word will pair with a preceding adjective if it occurs in the same sentence. The adjective and the word must be within a distance of 3 words.

The full set of possible bigrams might become too large and not very valuable in capturing sentiment associated bigrams. To this aim we propose to use the mutual information criterion to capture the relevant sentiment associated bigrams.

$$\mathrm{MI}(adjective - word) = \frac{freq(adjective - word)}{freq(adjective) \cdot freq(word)}, \tag{2}$$

where $freq(\cdot)$ represents the occurrence frequency. Here, an adjective-word pair is relevant if $MI(adjective - word)$ is above a pre-defined threshold. The minimum threshold set to capture relevant sentiment associated bigrams is set to 1E-5[9].

2.4.2 Orientation and intensity of sentiment words

The semantic orientation (SO) of an opinion word details the words' polarity. In other words, if the word is positive or negative. In Turney (2001) and Turney (2002) work is introduced a metric to estimate the degree of statistical dependence between two words (PMI-IR). In this metric is

[9] The mutual information criterion was tested with different threshold values.

observed the probability of two words co-occurring together and individually. In this context, the metric proposed by Turney is computed by observing the co-occurrence between a negative, and a positive, reference word and the candidate word on the Web corpus,

$$\mathrm{SO}(word) = \log_2\left(\frac{hits(word, "excellent") \cdot hits("poor")}{hits(word, "poor") \cdot hits("excellent")}\right), \tag{3}$$

where $hits(word)$ and $hits(word, "excellent")$ are given by the number of hits a search engine returns using these keywords as search queries. In Turney (2002), the words *excellent* and *poor* were chosen as reference words. However, the author reports that in the movie domain their results were unsatisfactory. To further investigate this metric is proposed a set of alternative reference words (Table 3).

Table 3. PMI pos/neg references

	PMI pos/neg references
T: Turney (Turney, 2002)	"excellent" / "poor"
G: Generic	"good" / "bad"
DS: Domain Specific	"best movie" / "worst movie"
DS+T	"excellent movie" / "poor movie"

When computing the $SO(adjective - word)$ one cannot control the distance between the *adjective* and the *word* in the search engine. To this end the SO of the pair *adjective-word* is given by the SO of the *adjective*. This assumption proves to be correct in several human expressions. For example, in the sentences "That movie is a waste of time" and "The great aggression where nation confronts nation", the SO of the adjectives *waste* and *great* enclose the correct SO.

In a sentiment analysis task to attain the SO of an opinion word is an important task. But, one should not diminish the importance to compute the sentiment intensity enclosed within each sentiment word. Hence, the semantic orientation determines the polarity of a word but does not weight the intensity expressed. For example, "sad" versus "depressed" or "contented" versus "ecstatic" (Liu, 2010). Here, we retrieve the sentiment words intensities from the lexical resource SentiWordNet (Esuli and Sebastiani, 2005; Esuli and Sebastiani, 2006; Baccianella et al., 2010). The intensity of an opinion word (ow) is defined as,

$$\mathrm{swn}(ow) = \begin{cases} posSWN(ow), & SO > 0 \\ negSWN(ow), & SO \leq 0' \end{cases} \tag{4}$$

where for a given ow, $posSWN(ow)$ corresponds to the SentiWordNet positive and $negSWN(ow)$ will correspond to the negative score respectively. For the bigram $(adjective - word)$ representation the swn sentiment word intensity value is given by:

$$\mathrm{swn}(adjective - word) = \mathrm{swn}(adjective) + \mathrm{swn}(word) \tag{5}$$

To investigate if a binary sentiment classification task (positive versus negative) is satisfactory for a sentiment classification problem we also analyse the performance of a multiple Bernoulli and multiclass classifier.

2.4.3 Sentiment classification models

Binary or Bernoulli: A sentiment classifier that detects the overall polarity of a text. Each review is classified as positive or negative. The classifier is defined as follows,

$$\mathrm{R} = \{(re_1, ra_j), (re_2, ra_j), \dots, (re_{i-1}, ra_j), (re_i, ra_j)\}, \tag{6}$$

$$\Phi\colon re_u \longrightarrow ra_j \in \{1,0\}, \tag{7}$$

where $u \in \{1, \dots, i\}$. Each review re_u is labelled (as positive or negative) according to the classifier function Φ inferred rating value ra.

Multiple Bernoulli: A multiple Bernoulli classification is performed for each rating in one-against-all scenario. Considering reviews with a rating range from 1 to 5 the multiple Bernoulli classification will perform 5 binary classifications and chose the prediction with the higher confidence value. The classifier is defined as follows,

$$\mathrm{p}_{re_{uj}} = \frac{re_{uj}}{\sum_{i=1}^{n=k} re_{ui}}, \qquad k \in \{1,2,\dots,maxRating\}, \qquad n = maxRating, \qquad j \leq maxRating, \tag{8}$$

$$\Phi\colon re_u \longrightarrow max\left(p_{re_{uj}}\right), \qquad \mathrm{j} \in \{1,2,\dots,\mathrm{maxRating}\}. \tag{9}$$

The sum of all the predictions within each rating (e.g. 1 to 5) the rating prediction is normalized, and is chosen the prediction with higher probability for each review u.

Classifiers are available in numerous approaches in which proved their applicability in the NLP domain. In this survey three classifiers were selected: Support Vector Machines, RIPPER and a generative sentence level classifier.

Support Vector Machines (SVM): The SVM algorithm aims at linearly divide the features with decision surfaces. The features projected near the surface limits will be selected. Support vectors define the optimal division between the categories (Joachims, 1998).

RIPPER: This algorithm identifies the class (or category) by building a set of decision rules (Cohen and Singer, 1999). RIPPER uses the technique of *direct representation* where each document is represented by a list of features without, as SVM, selecting a subset of the more relevant features. Also, this algorithm contemplates the absence and presence of a feature.

Sentence Level Classifier (SL): This classifier is proposed as a generative classifier in which the sum of the polarity of each feature (opinion word) within a sentence is observed. Each sentence is analysed individually which results in a polarity value for each sentence,

$$\text{sentencePol}(s) = \begin{cases} +1, & if \sum_{f \in s} polarity(f) > 0 \\ -1, & if \sum_{f \in s} polarity(f) \leq 0 \end{cases} \tag{10}$$

where $sentencePol(s)$ represents the polarity of the sentence s. Sentence s is composed by a set of features f where $polarity(f)$ represents the polarity of each feature. The polarity of a review is computed as,

$$\Phi(rev_i) = \sum_{s_i \in rev_i} sentencePol(s_j), \tag{11}$$

where the function $\Phi(rev_i)$ is greater than zero for positive reviews and less or equal to zero for negative reviews.

Finally, the evaluation of the sentiment analysis framework is given by the standard evaluation metrics precision (*p*), recall (*r*) and F-score, which is the harmonic mean between p and r,

$$\text{Fscore} = \frac{2 \cdot p \cdot r}{p + r}. \tag{12}$$

2.4.4 Datasets and Pre-processing Steps

The reviews are split at sentence level using the tools from Natural Language Toolkit[10] (NLTK). The stemming and identification of the adjectives, adverbs, verbs and nouns is performed with Freeling 3.0[11]. At sentence level for each word is observed the influence of negative and neutral

[10] http://nltk.org/

[11] http://nlp.lsi.upc.edu/freeling/

expressions, the sentiment expression must be in a maximum distance of 3 words. The collection of the chosen negative and neutral words are as follows: *not*, *however*, *rather*, *hardly*, *never*, *nothing*, *scarcely*; and *if*, *though*, *without*, *despite*, respectively.

Considering the absence of available labelled data one additional dataset was extracted from IMDb. This web resource contains a high amount of data in which several users only provide a review for a few number of movies. To overcome this constraint, it was implemented an extractor that crawls reviews by combining the top rated movies and users with a high value of *helpfulness* (Algorithm 1). The reason to obtain this additional dataset is because many well-known available datasets for sentiment analysis tasks contain no information regarding the rating of a review (Pang and Lee, 2004; Turney, 2002). Furthermore, to evaluate the proposed sentiment analysis survey the crawled dataset (IMDb-Extracted) and three state-of-the-art datasets have been chosen:

i. **polarity** (Pang and Lee, 2004): This dataset is frequently used for sentiment analysis tasks. Contains 2,000 movie reviews from IMDb and it's evenly split in positive and negative reviews. The dataset was split by 1,400 training and 600 test reviews respectively.

ii. **AmazonS1**[12]: This dataset contains reviews for three domains: books, dvds and music, with 4,000, 4,010 and 4,008 reviews respectively (Qu et al., 2010).

iii. **AmazonS2**[13]: This large-scale amazon dataset contains 698,210 amazon reviews (Jindal and Liu, 2008).

iv. **IMDb-Exctrated**: This dataset contains 671,950 reviews collected from IMDb. The reviews rating range is from 1 to 10 rating stars.

Originally, each Amazon review is labelled from 1 to 5 rating stars, and IMDb-Extracted from 1 to 10 rating starts. Also, unlike the *polarity* and AmazonS1 dataset the other datasets do not offer a proportional number of positive versus negative reviews. Once considered what inspires users to offer their' insights about a movie should foreseeable the lack of proportionality between positive and negative reviews. Amazon is related to users' purchases and intuitively we may say that the odds of a user acquiring a movie that is displeasing is smaller than to be pleased with the purchased. Yet, as regards to the movie domain this notion is not as intuitive. Table 4 presents the datasets details.

[12] http://www.mpi-inf.mpg.de/~lqu

[13] http://131.193.40.52/data

For the binary classification we have followed Bespalov et al. (2011) and Qu et al. (2010) reasoning: Amazon ratings of 3 rating stars or higher are labelled as positive, otherwise negative. As for IMDb, ratings of 6 rating starts or higher are labelled as positive, otherwise negative.

Algorithm 1: IMDb Extractor

```
Inputs: reviews, user_ids, movies_ids, maxReviews
Outputs: reviews(user_id, movie_id)
Steps:
begin
  addMovies(users_ids, movie_ids, reviews):
    begin
      foreach  movie_id$_i$ from movie_ids do
        Extract 20 top users (IMDb measures each user helpfulness)
        foreach user_id$_j$ from 20 top users do
          AddTo(user_id$_j$,users_ids)
          AddTo(user_id, movie_id$_i$, reviews)
    end
  addUsers(users_ids, movie_ids, reviews):
    begin
      foreach  user_id$_j$ from user_ids do
        Extract 5 top movies
        foreach movie_id$_i$ from 5 top movies do
          AddTo(movie_id$_i$,movie_ids)
    end
  Extract 250 top rated movies to movie_ids
  while len(reviews) < maxReviews
    addMovies(users_ids, movie_ids, reviews)
    addUsers(users_ids, movie_ids, reviews
end
```

Table 4. Detailed information of the datasets

Dataset	#Reviews	#Users	#Movies
polarity (Pang and Lee, 2004)	2,000	-	-
AmazonS1 (Qu et al., 2010)	12,018		
AmazonS2 (Jindal and Liu, 2008)	698,210	3,700	8,018
IMDb-Extracted	1,729,293	453,857	21,507

2.4.5 Results

Table 4 shows the evaluation results for the binary sentiment classification with the SVM classifier. Using the polarity, IMDb-Extracted and AmazonS2 datasets the sentiment analysis algorithm presents an F-score of 0.84, 0.81 and 0.84 respectively. In comparison to the *AmazonS2* and IMDb-Extracted datasets, the performance shown with the *polarity* dataset it a more balanced outcome since with the other datasets is observable a slight shift between precision and recall. The polarity dataset contains positive and negative IMDb reviews carefully chosen. In contrast, both AmazonS2 and IMDb-Extracted datasets contain reviews from a wider variety, thus, reviews with a not-so-obvious positive or negative polarity. Additionally, these dataset contain a much higher volume of reviews. Regarding the obtained performance with the AmazonS2 it's observed that recall outperforms precision. Considering the nature of this dataset this outcome should be expected since in the Amazon platform users can acquire products and provide a review regarding the purchase. Consequently Amazon reviews have a high probability of being spam. Hence, a large volume of Amazon reviews do not contemplate the respective rating. Spam reviews can be misleading to sentiment analysis algorithms since this algorithms evaluate the reviews associated text.

Figure 8 illustrates the F-score, recall and precision of the SVM classifier on the AmazonS1 dataset. Unlike AmazonS2 dataset, in the AmazonS1 dataset the sentiment evaluation is performed according to the domain. It is clearly observed that the overall performance is considerably lower than the one obtained with the other three datasets (Figure 7). The best F-score (0.63) is obtained in the DVDs domain. The lower performance with this dataset is also a consequence of the nature of the Amazon reviews since this dataset encloses many reviews with unrelated content regarding the explicit rating. Additionally, for each domain (DVDs/Books/Music) the dataset contains a low volume of reviews. In comparison the AmazonS2 is approximately 98% percent larger than AmazonS1.

Experimental results with different classifiers had SL classifier with the worst performance (Figure 7, Figure 8 and Figure 9). The SVM classifier outperforms RIPPER with only one exception – *AmazonS1* (DVDs). The amazon reviews contain a low volume of negative reviews which entails a greater challenge for the sentiment classifier. Additionally RIPPER classifier is able to correctly classify more positive reviews than SVM. However, RIPPER classifier misclassifies a higher number of negative reviews.

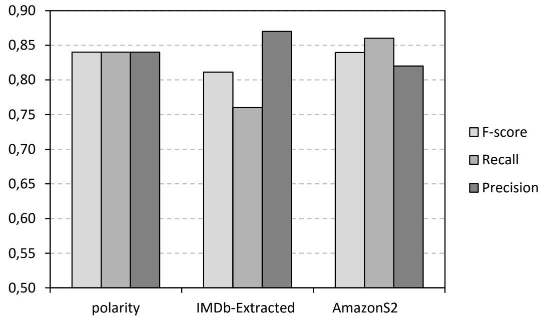


Figure 7: Binary classification for polarity, IMDb-Extracted and AmazonS2 datasets.

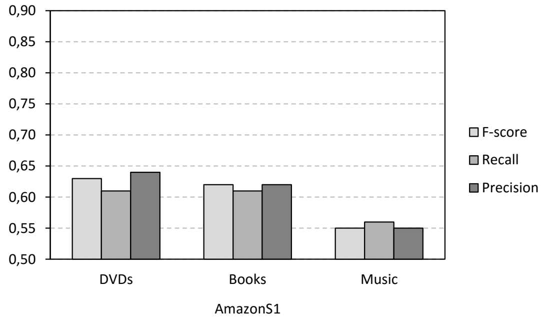


Figure 8: Binary classification for the multi-domain AmazonS1 dataset.

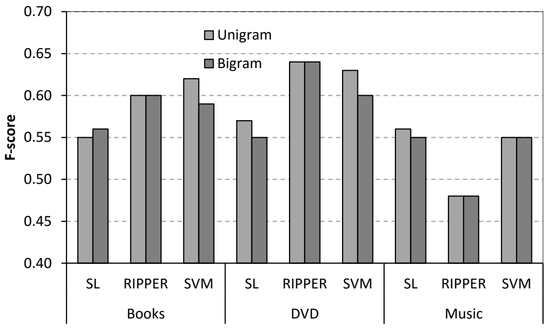


Figure 9: Unigram and bigram F-score with AmazonS1 dataset.

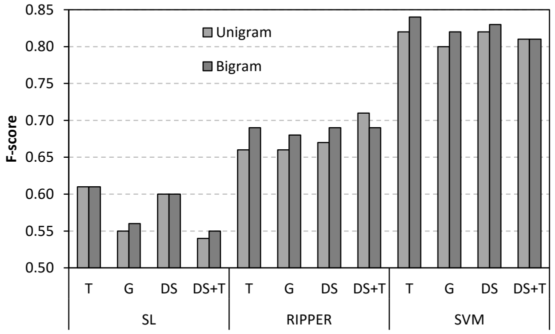


Figure 10: Sentiment analysis F-score with *polarity* dataset.

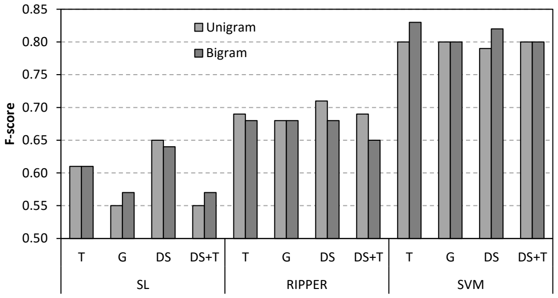


Figure 11: Sentiment analysis F-score with the multi-domain *AmazonS2* dataset.

Figure 9, Figure 10 and Figure 11 show the evaluation on the polarity, AmazonS2 and AmazonS1 dataset, respectively. These figures compare the unigram to the adjective based bigrams. The acronyms for the semantic orientation references are detailed in Table 3. SL, RIPPER, and SVM, refers to the sentence level, rule-based, and support vector machines classifiers, respectively.

Using the polarity dataset Pang et al. (2002) observed a considerably lower performance when representing sentiment words with only unigram adjectives. Figure 10 illustrates how the adjective-word bigrams representation improved the classifier performance, even though the comments from Pang et al. (2002) indicate that adjectives-only representation were unable to improve the sentiment analysis performance. In addition, Figure 10 shows that when combining with other sentiment bearing words the overall performance is improved. This last observation does not hold for the *AmazonS1* dataset, for the *AmazonS1* dataset the bigram representation has no effect or decreases the classifier performance (Figure 11). Should be taken into consideration that the *polarity* and *AmazonS1* datasets contain many linguistic differences whilst one focus on a single-domain (movie) and the other is multi-domain. The *AmazonS1* has an unbalanced positive/negative number of reviews which creates an uneven number of false positives and false negatives, which affects the classifiers performance. *AmazonS1* dataset is associated with products purchases which implies a completely different review structure and sentiment bearing words expressions.

In Figure 10 and Figure 11 is shown the semantic word references for PMI-IR (T, G, DS and DS+T) influence on the overall performance. The best word references results are observed with domain specific word references (DS or DS+T), and the original word references proposed by Turney (2002).

The multi Bernoulli classification entails a greater challenge than the binary classification (Sparling, 2011). Frequently, users reasoning when providing a rating, and the associated review, differs. On rating a product, some users can prove to be more demanding, or generous, than others. Figure 12 illustrates that in comparison to the binary classification, the performance decreases with the multiple Bernoulli classification. Yet, considering an IMDb review where the rating scale ranges from 1 to 10, in a multiple Bernoulli classification the classifier should be able to evaluate the difference from a review with a rating of 9 in relation to a rating 10. The multiple Bernoulli classification obtained a mean precision, recall, and F-score of 0.72, 0.68 and 0.65 respectively (Figure 12). In comparison the performance for lower ratings is not as good as for higher ratings which is consistent with W. Zhang et al., 2010 observations. Moreover, in the *IMDb-Extracted* dataset the volume of negative reviews is considerable lower than positive reviews.

The incorrect predictions performed by the multiple Bernoulli classifier have a tendency to be the neighbour ratings. Figure 13 shows the confusion matrix of the multiple Bernoulli classification. Considering the predicted rating and the actual rating, the confusion matrix illustrates the cross-rating inference. For example, in Figure 13 for rating 8 in the diagonal has 0.038 which is followed by 0.039 (rating 7) and 0.034 (rating 9). It is observable that the matrix diagonal, and the surrounding elements hold a higher accuracy, showing a low interference across distant ratings. Additionally, both datasets contain an unbalanced distribution of positive vs. negative reviews as most ratings are between 8, 9 and 10 (Figure 13) ratings. Hence the greater confusion among the low ratings.

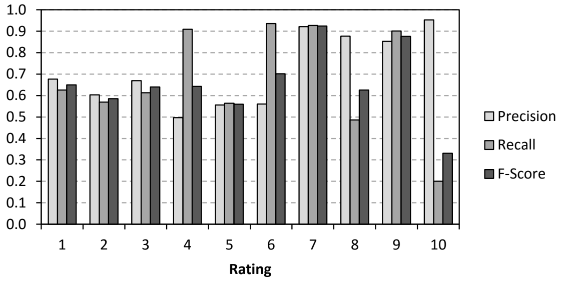
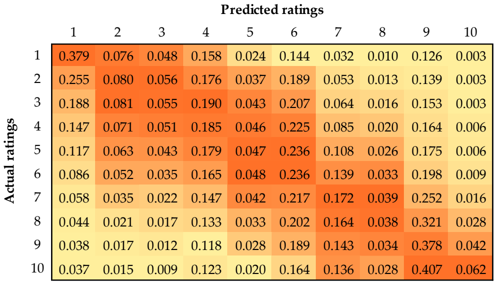


Figure 12: Multiple Bernoulli sentiment analysis evaluation for the *IMDb-Extracted* dataset.



		Predicted ratings									
		1	2	3	4	5	6	7	8	9	10
Actual ratings	1	0.379	0.076	0.048	0.158	0.024	0.144	0.032	0.010	0.126	0.003
	2	0.255	0.080	0.056	0.176	0.037	0.189	0.053	0.013	0.139	0.003
	3	0.188	0.081	0.055	0.190	0.043	0.207	0.064	0.016	0.153	0.003
	4	0.147	0.071	0.051	0.185	0.046	0.225	0.085	0.020	0.164	0.006
	5	0.117	0.063	0.043	0.179	0.047	0.236	0.108	0.026	0.175	0.006
	6	0.086	0.052	0.035	0.165	0.048	0.236	0.139	0.033	0.198	0.009
	7	0.058	0.035	0.022	0.147	0.042	0.217	0.172	0.039	0.252	0.016
	8	0.044	0.021	0.017	0.133	0.033	0.202	0.164	0.038	0.321	0.028
	9	0.038	0.017	0.012	0.118	0.028	0.189	0.143	0.034	0.378	0.042
	10	0.037	0.015	0.009	0.123	0.020	0.164	0.136	0.028	0.407	0.062

Figure 13: Predicted ratings distribution for the IMDb-Extracted dataset.

2.5 Summary

There are a variety of existing methods for sentiment lexicons, but most of them are either simple binary polarity or too generalist approaches. For more specific sentiment analysis problems a binary polarity (positive versus negative) might not be enough, and a sentiment word that is transversal to different domains – generic sentiment lexicons – may not enclose the correct sentiment for all the domains. On top of that generic sentiment lexicons miss to capture highly specific sentiment words (e.g. *oscar* for the movie domain).

On subjective text users' tend to influence other entities reputation and it is noticed that previous work on reputation systems have been taking a different approach other than look into how sentiment words relate to entities reputation. Moreover, in recommendation algorithms the sentiment influence that can be achieved from users' reviews is still an object of research as prior work has given more attention to other type of user generated content (e.g. explicit ratings).

3

Sentiment-Ranked Lexicons

The increasing popularity of the WWW led to profound changes in people’s habits. In this new context, sentiment expressions became important pieces of information, particularly in the context of online commerce. As a result, modelling text to find the vocabulary that is meaningful at expressing a sentiment has emerged as an important research direction. Here, we notice that existing work for sentiment lexicons lean towards generic sentiment words (Turney, 2002; Pang and Lee, 2008; Hu and Liu, 2004b; Liu, 2012). Words from these generic lexicons may not be designed for ranking tasks. Usually words from generic lexicons have fixed sentiment word weights (sometimes are simply positive/negative or have more than one sentiment weight). For this reason such lexicons do not handle domain specific words and do not capture sentiment word interactions. This underlines the need for a new breed of models that automatically generate domain specific sentiment lexicons with key properties for opinion analysis tasks. These models should deliver both a general lexicon and a domain specific one, with sentiment polarity and sentiment weight for their constituent words.

The proposed method aims at providing IR (Information Retrieval) tasks with a sentiment resource lexicon that is specifically designed for rank-by-sentiment tasks. The two main steps in building such resource, concerns the identification of the lexicon words and the words sentiment weight (we argue that a simple weight is not enough). The proposed algorithm is related to Labelled LDA introduced by Ramage et al. (2009) algorithm and LDA for re-ranking from Song et al. (2009). However, a fundamental difference is that we add an extra hierarchical level to smooth sentiment word distributions across different sentiment relevance levels.

The contributions presented in this chapter are: first, we propose a fully generative automatic method to learn a domain-specific lexicon from a domain-specific corpus, which is fully independent of external sources: there is no need for a seed vocabulary of positive/negative sentiment words. Second, a hierarchical supervised method is used to enhance the ability of learning sentiment word distributions in specific contexts. The uncertainty that arises from the sentiment word polarities used in previous works (Baccianella et al., 2010; Wilson et al., 2005) , are naturally mitigated in our proposal by ensembles of sentiment word distributions that co-occur in the same context.

The chapter is organized into 7 sections: Section 1 presents an overview of existing methods to obtain sentiment lexicons. Section 3 discusses the background of topic modelling techniques while Section 2 introduces the mathematical formulation used in this chapter. Section 4 describes the proposed Rank-LDA sentiment lexicon. Section 5 describes the proposed methodology for computing sentiment lexicons. Section 6 presents the experimental setting and Section 7 presents the discussion of the results.

3.1 Sentiment Lexicons

Previous works have proposed different methods to cope with the sentiment analysis problem (Zhang and Ye, 2008; Jo and Oh, 2011; Gerani et al., 2010; Aktolga and Allan, 2013). Zhang and Ye (2008) described how to use a generic and fixed sentiment lexicon to improve opinion retrieval through the maximization of a quadratic relation model between sentiment words and topic relevance. Other methods, as Gerani et al. (2010), applied a proximity-based opinion propagation technique to calculate the opinion density at each point in a document. More recently Jo and Oh (2011) proposed a unified model of products and services aspects and the respective associated sentiment. The model hypothesis is that each sentence concerns one aspect and all sentiment words in that sentence refer to that sentence. Later, Aktolga and Allan (2013) proposed to diversify search results by observing sentiment aspects. The common element among these works (Zhang and Ye, 2008; Jo and Oh, 2011; Gerani et al., 2010; Aktolga and Allan, 2013) is the SentiWordNet sentiment lexicon (Baccianella et al., 2010), as this is a popular and quite successful sentiment lexicon. An alternative method to capture additional sentiment words is to expand existing sentiment lexicons or manually annotated sentiment words lists (Hu and Liu, 2004b).

We notice that the research community has actively contributed to the sentiment analysis problems (Liu, 2012), overlooking the task of automatically learning domain sentiment vocabularies (Chen et al., 2012). One of the major challenges in sentiment analysis is the detection of the words that express a subjective preference, and domain related idiosyncrasies for which specific sentiment words are strongly related. Additionally, we notice that popular domain specific named entities frequently enclose important sentiment weights (in this document we refer to these named entities as sentiment anchors). For example, in the following sentences,

"If you liked *Requiem for a Dream* or *Blue Velvet*. Consider this one."

"Just like *Se7en* there is a huge twist that makes your blood curdle."

the named entities *Requiem for a Dream*, *Blue Velvet* and *Se7ven* are being used as a positive reference. In this context, to capture domain specific sentiment words and sentiment anchors might prove to be highly valuable. However, it is also a particularly challenging task: domain dependencies are constantly changing and opinions are not binary. The problem of sentiment anchors will be addressed in more detail in the next chapter, and in the present chapter we will give more emphasis to the detection of sentiment words and provide a qualitative discussion about sentiment anchors.

3.2 Topic Modelling Notation

In this chapter we propose Rank-LDA a novel method that uses topic modelling notation. To this end, we follow a similar notation as Blei et al. (2003):

- A *word* or *term* represents a unique word type of a fixed length vocabulary indexed by $\{1, \ldots, W\}$. We represent each word as unit-basis vector of length $\mathcal{W}$ that has a single component equal to one and all the other components equal to zero. The k-th word in the vocabulary is represented by a vector w such that $w^k = 1$ and $w^i = 0$ for $i \neq k$.
- A *document* is a sequence of N words denoted by $d = (w_1, w_2, \ldots, w_N)$, where w_i is the i-th word in the sequence. Note that since it is not required for the word sequence to match the original word order of the document, this is also known as bag-of-words representation.
- A *corpus* is a collection of D documents denoted by $D = \{d_1, d_2, \ldots, d_N\}$.
- $P(z|d)$ denote a document d distribution over topics z, $P(w|z)$ denote the probability distribution over words w given a topic z, and $P(w|d)$ the distribution over words within the document d.

In topic modelling, for a corpus with D documents and W words a topic model learns a relation between words and topics T, and a relation between topics and documents. Usually, observed variables are highlighted using shaded nodes while latent variables are denoted by unshaded nodes. The arrows between nodes indicate conditional dependency and the plates (boxes) inclosing nodes indicate repetitions of sampling steps. Finally, in the plate bottom right corner there is a number that indicates the number of samples (repetitions). In Figure 14 presents an example where x is an observed variable and y is a latent variable.

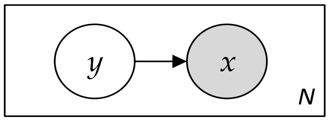


Figure 14: Example of plate notation.

3.2.1 Background: Probabilistic Topic Models

A fundamental problem in NLP is finding ways to represent large amounts of text in a compact way. Prior to 1988 the most popular text representation model in information retrieval tasks was the Vector Space Model (VSM), proposed by Salton et al. (1975). The main drawback of this technique is that VSM model assumes that terms are statistically independent, and it is known that words have dependencies such as synonyms and polysemy. As a result, its low semantic sensitivity fails to correctly evaluate documents with similar context but different term vocabulary.

In 1988 Dumais et al. (1988) proposed a method that uses the mathematical technique Singular Value Decomposition (SVD) to take into account term dependencies. The most popular name for this method is Latent Semantic Indexing (LSI), also known as Latent Semantic Analysis (LSA). Formally, the term document matrix $C = W \times D$ is a type of semantic space, in which, W represents the terms weight in the D documents. LSI method decomposes matrix C into three other matrices as follows,

$$C = U\Sigma V^T \tag{13}$$

where U is a $W \times W$ matrix of word vectors and its columns are eigenvectors CC^T, Σ is a diagonal $W \times D$ matrix that contains the singular values, V is a $D \times D$ matrix of document vectors and its columns are eigenvectors of $C^T C$. LSI reduces the dimensionality of the SVD by deleting coefficients in the diagonal matrix Σ.

LSI has proved its ability to overcome some of the VSM limitations, such as synonyms and polysemy (Landauer et al., 1998). However, for a generative model of text an algorithm such as maximum likelihood LSI might fit the problem as well. Additionally, the topics learned by LSI are not easily interpretable. The reason for this is based on the nature of the vectors that assign topics to each document. These vectors are linear combinations of the term-document frequencies, and for this reason it is not possible to identify important terms that are more relevant for each topic (Stevens et al., 2012). To overcome LSI shortcomings, Hofmann (1999) proposed the Probabilistic Latent Semantic Indexing (PLSI) model. In PLSI, each word is observed in a document as a sample from a mixture model and the mixture components are multinomial random variables (topics).

Given T topics, PLSI aims to find the probability distribution of words in a topic and the probability of topics in a document. Here, topics are latent variables and words are the observed variables. Following a generative process PLSI is computed as follows:

1. For each document $d \in D$ with probability $P(\theta_d)$
    a. Select a latent topic z with probability $P(z|d)$,
    b. Generate a word w with probability $P(w|z)$.

The mathematical definition of PLSI is obtained by following the abovementioned process as a jointly probability between a word and a document:

$$P(\theta_d, w) = P(\theta_d)P(w|\theta_d) \tag{14}$$

where,

$$\begin{aligned} P(w|\theta_d) &= \sum_{z \in Z} P(w|z)P(z|\theta_d) \\ &= P(\theta_d)\sum_{z \in Z} P(w|z)P(z|\theta_d). \end{aligned} \tag{15}$$

The graphical model representation of PLSI is shown in Figure 15. This model satisfies the topic models assumption, which is that a document consists of multiple topics. Here, $P(z|\theta_d)$ contains the weight of a topic z ($z \in T$) in a document d, symmetric Dirichlet priors θ on the distribution over topics for a given document and the distribution ϕ over words for a given topic.

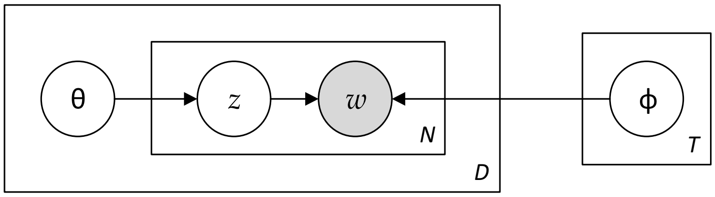


Figure 15: Graphical model representation of PLSI.

PLSI model represents each document as a list of topic weights. Hence, as Blei et al. (2003) notice, not using a generative probabilistic model prompts two main drawbacks in PLSI model:

1. Overfitting problems: the number of parameters grows linearly with the number of documents in the corpus.
2. The model does now allows to assign topic probabilities to unseen documents.

To overcome the abovementioned PLSI limitations, Blei et al. (2003) introduced LDA (Latent Dirichlet Allocation). LDA takes into account De Finetti (1990) representation theorem, which states that any collection of exchangeable random variables has a representation as a mixture distribution. The authors emphasize that unlike VSM, the assumption of exchangeability is not equivalent to the notion that random variables are independent and identically distributed. Here, exchangeability is with respect to an underlying latent parameter of a probability distribution. To this aim, LDA captures significant intra-document statistical structure by using mixing distribution of the conditional joint distribution of random variables and the joint distribution of random variables over the latent parameter.

LDA is an extension of PLSI which introduces symmetric Dirichlet priors θ on the distribution over topics for a given document and the distribution ϕ over words for a given topic. In Blei et al. (2003) the LDA generative process is described as follows:

1. For each topic, choose a distribution over words $\phi \sim Dir(\beta)$.
2. Choose N.

   A document d in a corpus D is represented by latent topics using the following generative process:
3. Choose $\theta \sim Dir(\alpha)$.
4. For each of the N words w_n:
   a. Choose a topic $z_n \sim Multinominal(\theta)$.
   b. Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

In the described generative process N is the number of words in a document, z_n is the n topic for the word w_n, θ is the topic distribution for a document, α is the parameter of the Dirichlet prior on the per-document topic distributions and β is the parameter of the Dirichlet prior on the per-topic word distribution. Figure 16 presents the graphical representation of LDA.

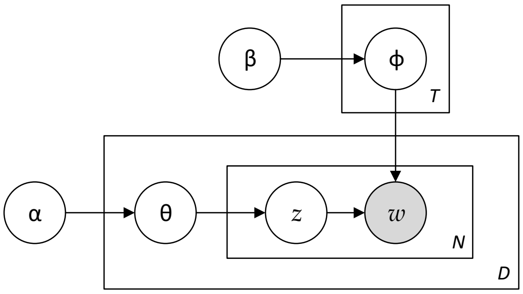


Figure 16: Graphical model representation of LDA.

The parameters α and β are corpus-level parameter, assumed to be sampled once in the process of generating a corpus. As noticeable in Figure 16, LDA model involves three levels: latent topics, documents and words. The joint probability of the corpus D given the hyperparameters α and β is given as follows:

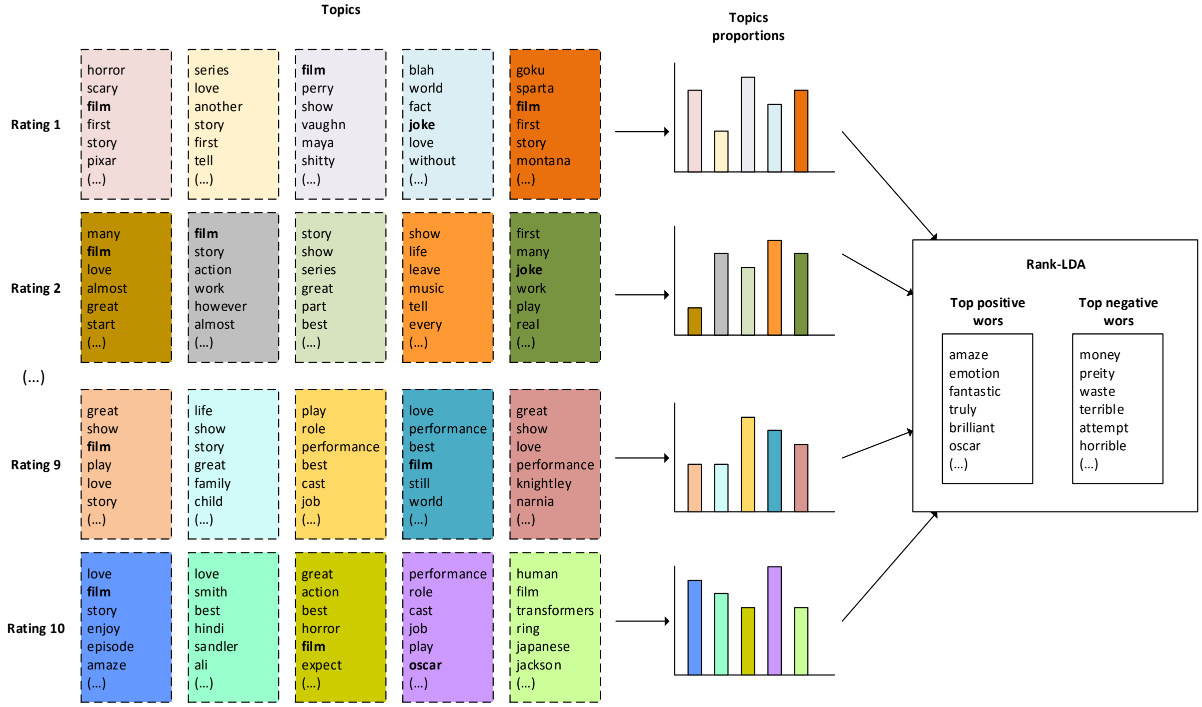
$$P(D|\alpha,\beta) = \prod_{t=1}^{T}\prod_{d=1}^{D}\prod_{n=1}^{N} P(\phi_t|\beta)\, P(\theta_d|\alpha) P(z_{dn}|\theta_d)\, P\big(w_{dn}\big|\phi_{z_{dn}}\big)\,. \tag{16}$$

3.3 Rank-LDA

LDA generative topic model is based on the exchangeability assumption for words and topics in a document (Blei et al., 2003) and performs as a dimensionality reduction technique that observes the generative probabilistic words' semantics, which is a requirement in a topic modelling problem. However we will not discuss topic modelling, in this section we propose a novel method: Rank-LDA. The proposed method aims to help in the problem of sentiment lexicon coverage limitations. To this end, the proposed method applies the LDA model in a sentiment analysis task.

Figure 17 reflects the intuition that reviews exhibit multiple topics with different proportions. Latent topics produced from different sentiment levels exhibit different topic proportions. More specifically, we observe words such as *film* that tend to have a similar distribution throughout different latent topics and sentiment levels; and *oscar* or *joke* that exhibit a more evident latent topic distribution according to its sentiment level. The proposed model (Rank-LDA) adds to the

LDA model a new variable associated to user reviews (document). This variable is associated to the overall opinion about the product that each review targets (e.g. a movie). To find the latent topics that best predict the chosen variable Rank-LDA jointly models the user reviews and the associated variables. Here, the user ratings correspond to the variables. The intuition behind LDA is that documents exhibit multiple topics (as seen on the left of Figure 17) and each topic represents a distribution over a fixed vocabulary. With Rank-LDA we will observe word probabilities by accommodating its distribution over a variable. In the context of Rank-LDA, we will refer to the variable associated to each user review as sentiment level or rating level. These terms (sentiment level or rating level) refer to the same variable that was added to the LDA method.



Topics
Topics proportions
Rating 1
horror scary **film** first story pixar (...)
series love another story first tell (...)
film perry show vaughn maya shitty (...)
blah world fact **joke** love without (...)
goku sparta **film** first story montana (...)
Rating 2
many **film** love almost great start (...)
film story action work however almost (...)
story show series great part best (...)
show life leave music tell every (...)
first many **joke** work play real (...)
(...)
Rating 9
great show **film** play love story (...)
life show story great family child (...)
play role performance best cast job (...)
love performance best **film** still world (...)
great show love performance knightley narnia (...)
Rating 10
love **film** story enjoy episode amaze (...)
love smith best hindi sandler ali (...)
great action best horror **film** expect (...)
performance role cast job play **oscar** (...)
human film transformers ring japanese jackson (...)
Rank-LDA
Top positive wors
amaze emotion fantastic truly brilliant oscar (...)
Top negative wors
money preity waste terrible attempt horrible (...)

Figure 17: (Left) The top 5-topics for lower and higher ratings. (Right) Top Rank-LDA sentiment words for movie reviews data.

In Rank-LDA, we first treat sentiment level as non-exchangeable with the words and assume that words are generated by topics, and the topics infinitely exchangeable within a document. Given a sentiment level s and by marginalizing over the hidden topics z, the sentiment distribution of a word w computed by Rank-LDA is,

$$p(w_i \mid s) = \int p(\theta) \cdot \prod_{n=1}^{N} p(z_n \mid \theta, s) p(w_n \mid z_n) \, d\theta + \tau \qquad (17)$$

where we compute the marginal distribution of a word given a sentiment level, over the T latent topics of the Rank-LDA model. The variable θ is the random parameter of a multinomial over topics and τ is a smoothing parameter that we set to 0.01[14].

A key characteristic of Rank-LDA intuition is that reviews from different sentiment levels share numerous words but each sentiment level exhibits those words in different proportion (Figure 17). Figure 18 illustrates a sample of the density distribution of words according to the sentiment level. This graphs shows the inner distributions of the Rank-LDA for different sentiment words (e.g. awful, emotion and wonderful). While the distributions of Figure 18 depict the marginal distributions of each sentiment word, the distributions of sentiment word interactions are also embedded in the hierarchical model structure but these are not so easy to visualize graphically. However, in the experiments section we discuss this model's property.

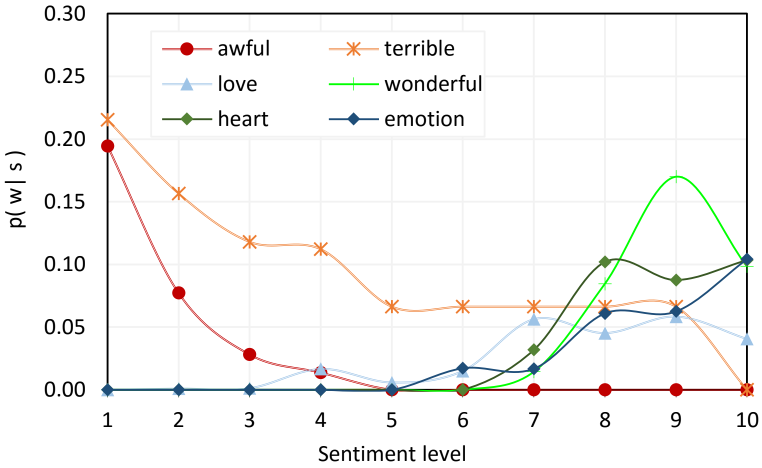


Figure 18. The sentiment distributions of words *emotion*, *love*, *wonderful*, *awful*, *heart* and *terrible*.

In PLSI model each word is a sample from a mixture model. However, as seen in Section 3.2.1, it does not provide a probabilistic model at the level of reviews (documents). Note that this is an important aspect for our model: PLSI computational complexity increases linearly with the size of the learning corpus, which can be critical when applied to a large dataset and leads to overfitting problems. LDA method overcomes this limitation by adding a Dirichlet prior to the per-document topic distribution. The generative nature of LDA method allows to detect the

[14] Smoothing parameter tested with different values.

words' probabilities at the level of latent topics and reviews (documents), which is valuable to unveil the sentiment words distribution. However, LDA model does not captures relevant sentiment words and evaluate its' polarity and weight. More specifically, LDA defines a topic as a distribution over a fixed vocabulary while Rank-LDA computes the distribution of words over topics that best describe an association to a sentiment. At its core, Rank-LDA links latent topics to the sentiment level of each document. Hence, in this hidden structure a set of hidden topics are activated for each sentiment level.

3.3.1 Graphical model for Rank-LDA

We address the problem of creating a sentiment lexicon based on user reviews without human supervision and propose to identify the sentiment words using a multi-level generative model of users' reviews. Intuitively, we use a generative probabilistic model that ties words to different sentiment levels, creating a sentiment rank over the entire sentiment lexicon. The main contribution of the proposed approach is that the model infers a sentiment lexicon by analysing user reviews as sentiment ranked sets of documents.

Problem formalization: consider a set of D documents (reviews) $\mathcal{D} = \{d_1, \dots, d_l\}$ containing user opinions towards a given product. According to the domain, a review is rated in a rating range from 1 to the maximum rating value R. In Rank-LDA each review d_i is represented by a tuple (w_i, s_i), where $w_i = (w_{\mathrm{i},1}, \dots w_{i,N})$ is a vector of N word counts. Then we add the variable sentiment level $s_i \in \{1, \dots, R\}$, responsible for quantifying the user opinion about the product (it corresponds to the user rating), and associate it to each word.

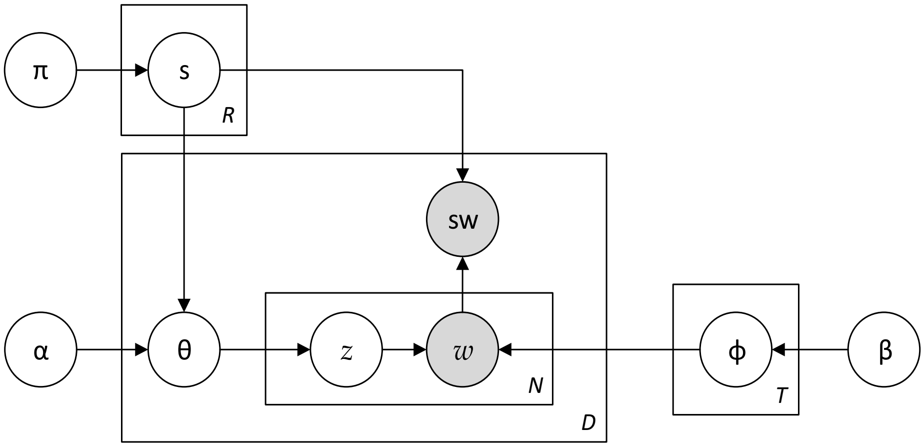


Figure 19. The Rank-LDA graphical model.

In Figure 19 we present the graphical model of Rank-LDA. The model is structured as follows: ϕ is the parameter of the multinomial distribution over topics, θ is the per-document topic Dirichlet$(\cdot\,|\alpha)$ distribution, z is the per-word latent topic assignment following a Multinomial$\left(\cdot\,|\theta^{(d)}\right)$ distribution, w correspond to the set of words observed on each document, N is the number of words in a document, T is the number of topics, θ is the topic distribution for a document and s is the per-document sentiment level Dirichlet Dirichlet$(\cdot\,|\pi)$ distribution. Finally, $s_i \in \{1, \dots, R\}$ is the per-document sentiment level and sw is the per-word random variable corresponding to the words sentiment distributions across the different sentiment levels. The random variables α, β and π are distribution priors: α is the Dirichlet parameters of the Dirichlet topic prior, β is parameters for the word prior while π is the label prior for documents sentiment level. Furthermore, algorithm 2 describes the generative process of the Rank-LDA model.

```
ALGORITHM 2.    The Rank-LDA generative process
For each topic $k \in \{1, \dots, T\}$
  Generate $\phi_k = \left(\phi_{k,1}, \dots, \phi_{k,N}\right) \sim \mathrm{Dir}(\cdot\,|\beta)$
For each document $d$:
  For each topic $k \in \{1, \dots, T\}$
    Generate $s_k^{(d)} \in \{1, \dots, R\} \sim \mathrm{Mult}(\cdot\,|\pi)$
  Generate $\alpha^{(d)} = L^{(d)} \cdot \alpha$
  Generate $\theta^{(d)} = \left(\theta_{k,1}, \dots, \theta_{k,N}\right) \sim \mathrm{Dir}(\cdot\,|\alpha)$
  For each $i \in \{1, \dots, N_d\}$:
    Generate $z_i \in \left\{\lambda_1^{(d)}, \dots, \lambda_{M_d}^{(d)}\right\} \sim \mathrm{Mult}\left(\cdot\,|\theta^{(d)}\right)$
    Generate $w_i \in \{1, \dots, N\} \sim \mathrm{Mult}\left(\cdot \mid \beta_{z_i}\right)$
For each sentiment word $w_i$:
  Compute the marginal distribution $\int p(sw_i \mid w_i, s)ds$
```

Computationally, in Rank-LDA, reviews are rated in a particular scale (usually 1 to 10 or 1 to 5). Iteratively, a set of topic distributions per sentiment level are computed and this is repeated until all sentiment levels are incorporated in the Rank-LDA structure. In this hierarchical approach, the ratings information are imposed over the topic distributions rendering distributions of words that will allow the identification of words used to express different sentiment relevance levels.

The sentiment word distribution function can be used to rank words by its positive/negative weight and to calculate a word sentiment relevance at different sentiment levels. A straightforward way of achieving this conversion is through the function

$$RLDA\left(w_{i,j}\right) = \frac{p(w|s=i) - p(w|s=j)}{min\left(p(w|s=i), p(w|s=j)\right)} \tag{18}$$

where $p(w|s=i)$ and $p(\mathrm{w}|s=j)$ denote the word w sentiment level in ratings i and j. The obtained sentiment lexicon with Rank-LDA is denoted as RLDA.

3.3.2 Relations to other LDA extensions

In this Section we will briefly discuss how Rank-LDA differs from similar methods available in the literature. LDA's Dirichlet distributions over topics and words detects words semantic associations. However, as Blei and McAuliffe (2007) notice, this is not a supervised approach. For this reason, Blei and McAuliffe (2007) introduced sLDA (supervised latent Dirichlet allocation) in which the authors propose to add an extra layer to the LDA model. In sLDA each document is associated with a response variable (e.g. rating given to a movie). The top left of Figure 20 shows the graphical model representation of sLDA model. Here, $\alpha,\ \phi,\ \eta$ and σ^2 are unknown constants to be estimated which are used in sLDA instead of the random variables of the original LDA model. Also, η and σ^2 are the response variable parameters and y corresponds to the response variable (e.g. rating level).

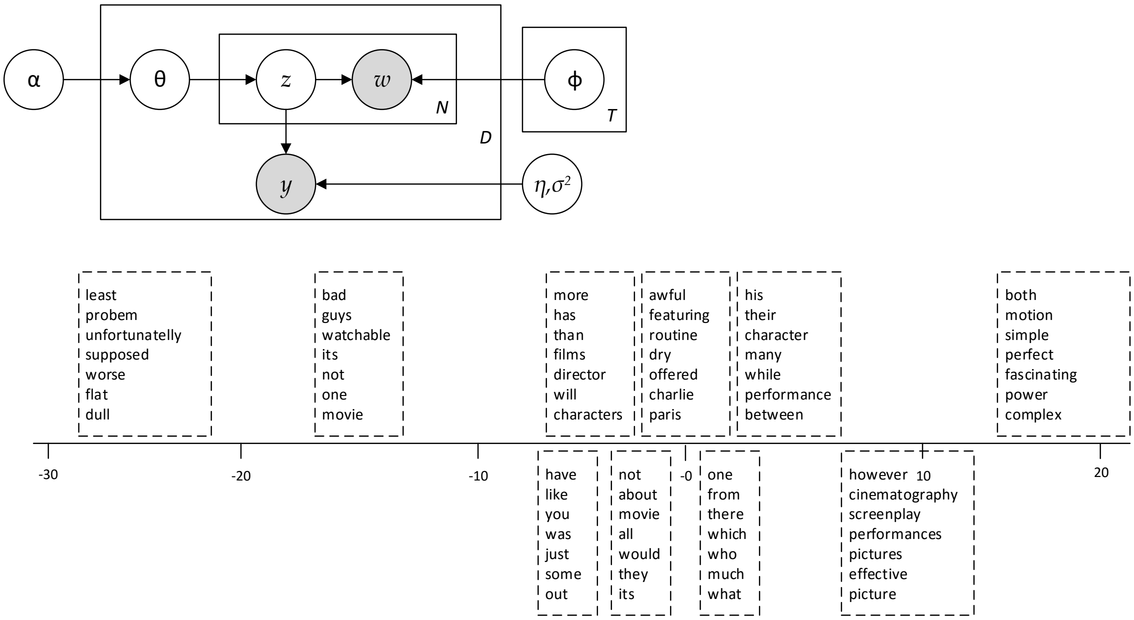


Figure 20: (Top) A graphical model representation of sLDA. (Bottom) The topics of a 10-topic sLDA model for movie reviews data (Blei and McAuliffe, 2007).

We find that it is important to distinguish Rank-LDA from sLDA. Similar to the experiments that will be detailed in the Results and Discussion Section of this chapter, Blei and McAuliffe (2007) propose an algorithm for a sentiment analysis problem. In the bottom of Figure 20 is noticeable that the words within the sample documents appear to correlate to sentiment. For example, in the document most to the left we have the words *worse* and *dull*, and in the document most to the right *fascinating* and *complex*. However, we also notice that there any many other words that do not as clearly correlate to sentiment. The reason for that sLDA aims to detect which words best describe the documents by using rating level as a class. Therefore sentiment words are depicted within those classes (rating levels). Moreover, sLDA adds an extra layer for the topic probability while Rank-LDA adds an extra layer for the word probability.

Ramage et al. (2009) propose Labelled LDA (L-LDA), a model that associates each label with one topic in direct correspondence. Similar to LDA the L-LDA models each document as a mixture of latent topics and generates each word from one topic. However, in contrast to LDA the L-LDA only incorporates latent topics that are within the documents' label set. Figure 21 shows L-LDA graphic model representation, where the multinomial mixture representation θ is affected by the Dirichlet prior α and by a newly introduced variable, the topic presence indicators Λ. The topic presence indicator is the additional dependency introduced by L-LDA. Here, each document is represented by a list of binary topic presence/absente indicators $\Lambda = (l_1, \dots, l_T)$. Additionally, L-LDA sets the number of latent topics to be the number of unique labels T in the corpus.

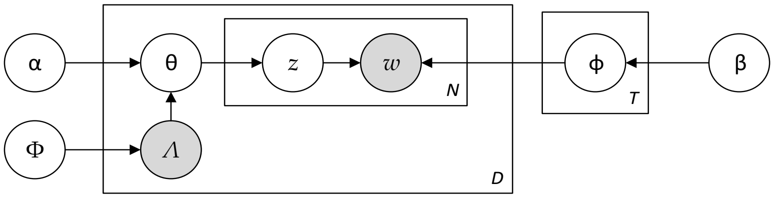


Figure 21: Graphical model of Labelled LDA.

Unlike traditional LDA and Rank-LDA, L-LDA restricts the multinomial mixture representation θ to be defined only over the unique labels that are activated for a document d. To this end, Ramage et al. (2009) define for each document d a document-specific label projection matrix: $L^{(d)}$ of size $M_d \times T$ where $M_d = \left|\lambda^{(d)}\right|$ and T is the number of latent topics. Here, $L^{(d)}$ is the matrix to project the parameter vector of the Dirichlet topic prior α. Each position in the matrix has an entry of 1 if and only if the document label in that entry is equal to the latent topic k. Therefore, the role of the projection matrix is to activate and de-activate topics. In Rank-LDA we further extend the projection matrix $L^{(d)}$ to link the latent topic variables k to sentiment relevance levels $s_k^{(d)}$ in which the rows of the projection matrix will correspond to a set of topics, as Rank-LDA topics are associated to a sentiment level. For example, consider the case where we have 3 sentiment levels and 2 latent topics per sentiment level. If a given document d has a rating level equal to 2, then $s_k^{(d)} = (0,0,1,1,0,0)$ and the projection matrix would be:

$$\begin{pmatrix} 0\,0 & 1\,0 & 0\,0 \\ 0\,0 & 0\,1 & 0\,0 \end{pmatrix}.$$

This answers our requirement that a document is represented by a set of sentiment ranked words.

3.4 Sentiment Analysis Tasks

Sentiment classification, also commonly known as the document-level sentiment classification (see Section 2.1), is the most extensively studied sentiment analysis task (Pang and Lee, 2008). A less addressed task, but also popular, is sentiment ranking. Note the difference between these methods, while sentiment classification aims to answer the question *"is this review positive or negative?"* sentiment ranking aims to order a collection of reviews, *"rank these reviews by how positive they are"* (Pang and Lee, 2008). We emphasize that for sentiment classification and sentiment ranking models it is particularly important to correctly capture the relevant sentiment words polarity and weights.

3.4.1 Sentiment Classification

For the sentiment classification task, we use three different approaches: binary or Bernoulli (B), multiple Bernoulli (MB) and one-against-all (OAA). The default learning algorithm from the Vowpal Wabbit[15] (VW) library was chosen for this task – an online gradient descent method which optimises the square loss on a linear representation. An important VW aspect for the performed classification is that the algorithm is able to rapidly handle large datasets while adjusting feature weights in an online manner. Hence it is easily applied on learning problems with sparse tera-features (Yuan et al., 2011).

Binary or Bernoulli

The reviews are classified according to a binary classification algorithm. The intuition behind this is simple: the review sentiment level is adapted to a positive versus negative viewpoint. To this end, each review is labelled as positive or negative as follows,

$$R = \{(re_1, ra_j), (re_2, ra_j), \dots, (re_{i-1}, ra_j), (re_i, ra_j)\} \tag{19}$$

[15] https://github.com/JohnLangford/vowpal_wabbit

$$\Phi\colon re_u \longmapsto ra_j \in \{1,0\}, \tag{20}$$

where $u \in \{1, \dots, i\}$ and each review re_u is labelled as positive or negative according to the classifier function Φ inferred rating value ra.

Multiple-Bernoulli (MB)

Starting from the word distributions $p(w_i \mid s)$, we designed a straightforward classifier that identifies the sentiment level of a review. For this task we implement a multi-class classifier that aims to find the most probable sentiment level $s = ra$ of a given review re_j. This classifier benefits from observing each sentiment level individually. The most probable sentiment level is obtained as follows,

$$\underset{ra}{arg\,max}\left[p\left(s = ra \mid re_j\right) = \frac{p\left(\, re_j \mid s = ra\right) \cdot p(s = ra)}{\int_l \; p\left(s = l\,, re_j\right)}\right]. \tag{21}$$

One-Against-All (OAA)

VW provides one-against-all implementation that internally reduces the multiclass classification problem in K binary classification problems, where K is the number of sentiment levels. OAA differs from MB in the metric used to learn the optimal class. VW implements a multiclass log loss while MB metric is described in the MB Equation details.

3.4.2 Sentiment Ranking

Sentiment classification can be naturally formulated as a regression problem because ratings are ordinal. An ordinal regression problem might fit best to the problem as for each rating reviews' semantics may not correspond to a point in scale (i.e. 4 in a scale from 1 to 5). However, instead of proposing a regression algorithm we propose to address this problem by ranking reviews by its sentiment level (rank level). Nevertheless, the intuition is: review semantics may not correspond to a fixed point in scale. For this task, we assume that each sentiment level has its own distinct vocabulary. Additionally, reviews exhibit multiple words and Rank-LDA studies each word sentiment distribution, in which the sentiment weight is used in the sentiment ranking method.

The goal is to retrieve reviews that satisfy a given query. To this end, we consider a query $Q(q_1 \dots q_n)$ that contains a set of keywords $q_1 \dots q_n$ that correspond to the review content.

For each query the reviews that are returned should contain a high similarity with the search query, where the query corresponds to a given review. A set of queries (reviews) were manually selected that represent the sentiment level. Finally, to rank reviews by its sentiment level we compute a ranking algorithm that given a review re_j minimizes the distance between the query sentiment level qs_i and the inferred sentiment level $p\left(s = qs_i \mid \mathrm{re}_j\right)$. Reviews are ranked as follows,

$$p\left(s = qs_i \mid re_j\right) \propto exp\left(\sum\nolimits_k \gamma_k \cdot p\left(sw_{j,k} | s_i\right)\right) \tag{22}$$

where $sw_{j,k}$ is the sentiment word weight for a review re_j given the sentiment level k.The parameters γ_k are optimized to minimize the expected cost between the observed rating and the inferred cost. Having in mind that when ranking opinions, one wishes to retrieve reviews that are in a close range to the query, it is computed the squared error cost function to minimize the penalty over close sentiment levels and maximizes the penalty over more distant sentiment levels.

3.5 Evaluation

This section describes the experiments to assess the effectiveness of the proposed method. The first set of experiment concerns sentiment ranking by rating level while the other experiments detail the Rank-LDA performance in sentiment classification tasks. As evaluation metrics, P@5, P@30, NDCG and MAP are used in the retrieval experiments and precision, recall and F1 in the classification experiments.

3.5.1 Datasets

IMDb-Extracted: This dataset contains over 703,000 movie reviews. Reviews are rated in a scale of 1 to 10. We crawled this dataset because most of the existing review datasets either lacked the rating scale information, targeted multiple domains, did not capture cross-item associations or were limited to small numbers. In Section 2.4.4 it is described in more detail the motivation and how the dataset was extracted.

TripAdvisor: This dataset contains 189,921 reviews and each review is rated in a scale of 1 to 5. This dataset was made available by Wang et al. (2010). The dataset was split into 94,444 documents for training and 95,477 documents for testing.

3.5.2 Evaluation Metrics

Sentiment classification methods are evaluated according to its performance in classifying review sentiment level. For positive reviews, negative reviews and a specific sentiment level (from 1 to maximum rating) we compute precision, recall and F1-measure. For a given sentiment level, precision weights the proportion of the correct classifications but does not observe the missed reviews from that sentiment level (FN, false negatives). In contrast, recall observes FN but does not weigh reviews that were incorrectly classified as belonging to that sentiment level (FP, false positives). Here, F1-measure resolves this constraint by performing a harmonic mean between precision and recall. Furthermore, the classifier performance is evaluated with the micro-averaging precision, recall and F1. These methods are described as follows,

$$MicroP = \frac{\sum_{ra=1}^{R} TP}{\sum_{ra=1}^{R}(TP_{ra} + FP_{ra})} \tag{23}$$

$$MicroR = \frac{\sum_{i=1}^{R} TP}{\sum_{ra=1}^{R}(TP_{ra} + FN_{ra})} \tag{24}$$

$$MicroF1 = \frac{2 \times MicroP \times MicroR}{MicroP + MicroR}. \tag{25}$$

To compute the relevance of a review to a query (selected representative review) we use P@5, P@30, MAP and NDCG. P@5 and P@30 corresponds to the precision at the top 5 and 30 retrieved reviews. MAP (mean average precision) is defined as the mean of the average precision (AP) for each sentiment level, where AP is the average of the precision at each recall point in a ranked list of relevant reviews. MAP is described as follows,

$$MAP = \frac{1}{R} \times \sum_{i=1}^{R} \frac{\sum_{j=1}^{m} P_j}{m} \tag{26}$$

where R is the number of sentiment levels, m is the number of recall points in a ranked list and P_j is the precision at the j^{th} recall point. Finally, NDCG is the normalized DCG (discounted cumulative gain) which is a popular evaluation metric to measure ranking quality. For a position k in the retrieved reviews for a query q NDCG is defined as follows,

$$NDCG(k) = DCG_{max}^{-1}(max) \sum_{j:\pi_i(j)\leq k} \frac{2^{y_{i,j}} - 1}{\log_2\left(1 + \pi_i(j)\right)} \qquad (27)$$

where $DCG_{max}(k)$ is the normalizing factor, $y_{i,j}$ is the sentiment level label of $re_{i,j}$ in ranking list π_i, $\pi_i(j)$ is the position of review $re_{i,j}$ in the retrieved ranking list π_i.

3.5.3 Baselines: Dictionary-based Sentiment Lexicons

The obtained sentiment lexicon is compared to three well-known sentiment lexicons:

SentiWordNet (SWN) (Esuli and Sebastiani, 2006): this lexicon was built with a semi-automatic method where some manual effort was used to curate the output. It was selected the top 2,290 positive words and the bottom 4,800 negative words, corresponding to a sentiment weight greater than 0.6 (on a scale of 0.0 to 1.0).

MPQA (Wilson et al., 2005): this lexicon provides a list of words that have been annotated for intensity (weak or strong) in the respective polarity – positive, negative or neutral. The lexicon was obtained manually and an automatic strategy is employed afterwards. Contains 2,718 positive and 4,912 negative words.

Hu-Liu (Hu and Liu, 2004a): this lexicon contains no numerical scores. Based on the premise that misspelled words frequently occur in users' reviews these words are deliberately included in the lexicon. The lexicon contains 2,006 positive and 4,683 negative words.

3.5.4 Baselines: Corpus-based Sentiment Lexicons

LLDA (Ramage et al., 2009): Labeled LDA is a topic model that constrains LDA by defining one-to-one correspondence between LDA's latent topics and user tags. In the present work, tags will correspond to user ratings.

Web GP (Velikovich et al., 2010): A method based on graph propagation algorithms to construct polarity lexicons from lexical graphs.

Full vocabulary baselines. The standard TFIDF weighting scheme was used in the recently proposed D-TFIDF sentiment word weighting scheme (Martineau and Finin, 2009). D-TFIDF combines TFIDF with a weight that measures how a word is biased to a dataset.

In Table 5 shows a description of the number of words captured for the corpus-based sentiment lexicons in the IMDb and TripAdvisor datasets.

3.5.5 Rank-LDA Lexicons

The proposed methods introduced in this chapter are: Rank-LDA (**RLDA**); **D-RLDA** which applies Martineau and Finin (2009) D-TFIDF weighting scheme adapted to the RLDA method. Following the strategy described in Section 3.3.1, we compute Rank-LLDA (**RLLDA**) and Rank-Web GP (**RWGP**) lexicons. The number of words within each of these lexicons is described in Table 5, while

Table 6 shows the number of sentiment words (sw) detected by the RLDA sentiment lexicon when observing 100, 500, 1000, 2.000 and 5.000 words (w) in each latent topic. Notice that there is a high percentage of sentiment words (captured by RLDA) that are not within Hu-Liu and SentiWordNet lexicons. For example, in the first row 82% of the sentiment words captured by RLDA are not in Hu-Liu lexicon while for SentiWordNet, 24% of the sentiment words captured by RLDA are unknown to this sentiment lexicon.

Table 5: Number of words in lexicons built from IMDb and TripAdvisor datasets.

	IMDb	**TripAdvisor**
RLDA/D-RLDA	9,510	4,936
RLLDA	55,428	15,086
RWGP	1,406	875
D-TFIDF	367,691	123,678
LLDA	97,808	44,248
Web GP	3,647	2,261

Table 6: Detected Sentiment Words not found in Hu-Liu and SentiWordNet lexicons (IMDb).

RLDA	**Hu-Liu**	**SentiWordNet**
$w = 5{,}000$ / $sw = 9{,}510$	7,827 (82%)	2,237 (24%)
$w = 2{,}000$ / $sw = 3{,}715$	3,074 (83%)	666 (18%)
$w = 1{,}000$ / $sw = 1{,}644$	1,379 (84%)	208 (13%)
$w = 500$ / $sw = 806$	694 (86%)	72 (9%)
$w = 100$ / $sw = 160$	144 (90%)	16 (10%)

To obtain an analysis of the sentiment anchors[16] (described in Section 1 and will be discussed in the Experiments Section) sentiment distribution, three subsets of the RLDA lexicon were analysed: (1) for the IMDb-Extracted dataset we obtain: RLDA-A corresponds to the base lexicon without the actor names; (2) RLDA-TA is the base lexicon without the movie title and actor names; and (3) RLDA-TCA is the base lexicon without the movie titles, actor names and character names.

3.6 Results and Discussion

3.6.1 Sentiment Ranking

In this section we present the evaluation results in a task of sentiment retrieval by rating level. Table 7 shows the opinion retrieval performances. The table shows that the proposed methods RLDA, D-RLDA, RLLDA and RWGP, LLDA and Web GP are consistently effective across the four evaluation metrics (P@5, P@30, MAP and NDCG).

Table 7: Sentiment ranking. P@5, P@30, MAP and NDCG for two datasets. * is the best result, the statistical significance t-test showed that the differences in retrieval results between D-RLDA and SentiWordNet are statistically significant.

	IMDb				**TripAdvisor**			
	P@5	**P@30**	**MAP**	**NDCG**	**P@5**	**P@30**	**MAP**	**NDCG**
RLDA	92.00	90.67	56.33	78.17	92.00	98.67	65.34	81.34
DRLDA	90.00	**91.67***	56.37	78.18	96.00	98.67	65.33	81.31
RLLDA	**94.00***	91.00	55.12	77.16	**100.00***	98.00	65.92	81.50
RWGP	92.00	88.67	56.64	**79.21***	100.00*	98.00	64.02	81.47
Hu-Liu	82.00	76.67	43.85	72.44	92.00	90.00	55.76	78.12
MPQA	82.00	81.67	46.22	73.61	**100.00***	87.33	57.99	78.95
SWN	88.00	89.00	53.52	76.77	92.00	96.00	63.70	80.89
D-TFIDF	76.00	81.67	54.72	77.01	96.00	**99.33***	66.51	**81.81***
LLDA	92.00	90.34	55.04	77.13	**100.00***	98.67	**66.61***	**81.87***
Web GP	88.00	89.67	**57.20***	79.11	96.00	96.00	65.36	**81.83***

[16] Popular domain specific named entities that enclose sentiment weights.

In general the proposed sentiment lexicons outperform baseline lexicons. However, we note that for the TripAdvisor dataset the metric D-TFIDF presents fairly good results. More specifically, for both MAP and nDCG evaluation metrics. Here, we would like to recall that D-TFIDF presents a weight for all words: 367,691 and 123,678 words in the IMDb and TripAdvisor datasets respectively. This is a considerable difference in comparison to all the other sentiment lexicons (Table 5). Therefore, the D-TFIDF metric would not be as useful as the proposed approach for creating sentiment lexicons. For instance, in the TripAdvisor dataset the most relevant positive and negative words obtained with the D-TFIDF metric are {*cevant*, *untrained*, *unconcerned*, *enemy*} and {*leonor*, *vaporetto*, *unpretentions*, *walter*}, respectively. In contrast, the most relevant positive and negative words obtained with the D-RLDA lexicon are {*full*, *great*, *excellent*, *wonderful*} and {*tell*, *call*, *dirty*, *bad*}, respectively. These examples illustrate the discriminative nature of D-TFIDF and the generative nature of D-RLDA.

LLDA introduced by Ramage et al. (2009) is a model of multi-labelled corpora that addresses the problem of associating a label (a rating, in our case) with one topic. In particular LLDA is strongly competitive with discriminative classifiers in multi-label classification tasks. However, we note that despite presenting equally good results LLDA requires a higher number of words to correctly perform the opinion retrieval tasks than RLDA. Intuitively, the proposed task could be approximated to a topic classification task for which LLDA is more appropriate. However, LLDA is not capturing sentiment words. Indeed, similar to D-TFIDF, it is capturing words that best describe each rating level. On the other hand, Velikovich et al. (2010) proposed the web-derived lexicon (Web GP) which performs at a similar level although with a considerable lower number of words – approximately 50% lower than the ones captured by RLDA. Web GP constructs a polarity lexicon using graph propagation techniques whereas the graph captures semantic similarities between two nodes. In contrast, our method relies on LDA generative model to capture semantic similarities of words. Nonetheless, unlike Web GP the proposed sentiment lexicon (RLDA) does not require a seed of manually constructed words to produce the lexicon. In addition, asserting the ideal number of sentiment words that are required for a sentiment lexicon can be highly challenging. As a consequence, in a sentence level classification tasks the sentiment words selected by Web GP may not be enough to discriminate sentiments at sentence level.

3.6.2 Sentiment Classification

To evaluate the gains of using the proposed method in a supervised sentiment classification task we measured the performance of the lexicons in a binary (B), multilevel (MB) and one-against-all

(OAA) sentiment classification task (Table 8). The MB classifier predicts the rating that presents the highest probability in a rating range of 1 to 10 (IMDb) or 1 to 5 (TripAdvisor). MB entails a greater challenge than positive vs. negative, or vs. all, unlike the other classifiers the MB classifier attempts to distinguish between similar ratings (Sparling, 2011). In Table 8, we can verify that with the IMDb dataset the MB classifier was outperformed by the B and OAA classifiers. However, notice that mid-range ratings represent a greater challenge than high or low ratings. We found that the TripAdvisor dataset has a lower rating range, thus, lower uncertainty between mid-range opinions. In other words, users tend to be blunter when writing a highly positive or negative review. Obviously these mid-range reviews negatively affect the overall performance. For instance, Jo and Oh (2011) opt to remove all ratings from borderline reviews from the classification task. However, in this experiment we chose to remain as close to the real data as possible. When analysing the results for both datasets, we see that our method has a good performance consistently outperforming other lexicons or being as good as the best.

Martineau and Finin (2009) proposed the metric D-TFIDF to weight words scores. In their study the authors found that in comparison to the Pang et al. (2002) subjectivity dataset[17] D-TFIDF shows an improvement over the accuracy. Moreover, variants of our proposed method outperformed dictionary-based sentiment lexicons, while D-TFIDF presents a similar performance. But, an important difference is that RLDA lexicon only required 2.6% of the words used by D-TFIDF. This entails a very aggressive and effective feature selection. In Figure 22 we observe the size (number of words) of different lexicons (Table 5) and the respective precision obtained with the binary sentiment classification. This illustrates the impact of the feature selection in the performance of the sentiment classifier. It is clear that although D-TFIDF presents a comparable performance the other lexicons fit better to a sentiment classification problem.

Weichselbraun et al. (2013) observations about hotel reviews vocabulary were also noticed in our study. The vocabulary used in hotel reviews is more “contained” than the one used in movie reviews. In particular, in the latter users tend to be more creative and less concise (IMDb data). Users create longer documents discussing different topics and frequently recur to the use of synonyms to avoid boring the reader with repetition (Turney, 2001; Martineau and Finin, 2009). This domain characteristic is reflected in the classification performance, which performs better in domains where both the vocabulary and the documents’ length are more concise. Results also

[17] http://www.cs.cornell.edu/people/pabo/movie-review-data/

show that generic sentiment lexicons (e.g. SWN) can perform quite well on sentiment analysis tasks, however almost always below other finer-grained lexicons.

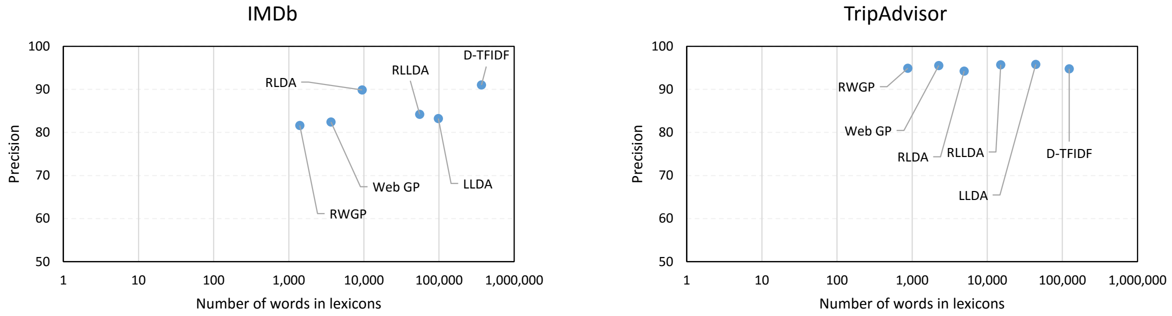


Figure 22: Precision for the binary sentiment classifier. The results with the IMDb dataset are on the left and on the right the TripAdvisor dataset results. Considering the number of words within each lexicon precision shows the results in the logarithmic value.

Table 8: Sentiment classification. Micro-averaging precision (P), recall (R) and F1-measure for binary classification, P for multiple Bernoulli (MB) and one-against-all (OAA) for two datasets. * is the best result, significance was tested using t-test and all classifiers differ from the baseline with a value of $p < 0.01$.

	IMDb					**TripAdvisor**				
	Binary			**MB**	**OAA**	**Binary**			**MB**	**OAA**
Method	P	R	MicroF1	P	P	P	R	F1	P	P
RLDA	89.05	88.59	88.82	73.02	70.29	94.47	93.41*	93.94*	88.40	90.89
D-RLDA	89.87*	86.85	88.33	73.09	70.98	94.24	93.12*	93.68*	94.24	90.87
RLLDA	84.21	97.04	90.17*	73.67*	80.80*	95.73*	91.56	93.60	95.58*	91.30*
RWGP	81.61	96.63	88.48	69.39	76.61	94.90	88.78	91.73	93.52	88.40
Hu-Liu	73.46	94.82	82.78	61.43	65.87	94.52	65.11	77.11	94.52	83.12
MPQA	75.59	93.52	83.60	62.04	66.05	94.27	73.56	82.64	94.27	84.42
SWN	73.90	99.50*	84.81	68.59	68.77	94.38	91.55	92.95	94.38	88.48
D-TFIDF	91.05	85.76	88.33	70.36	73.68	94.78	92.47	93.61	94.78	90.77
LLDA	83.21	97.04	89.60	73.67*	80.62	95.79*	87.69	91.56	95.58*	91.38*
Web GP	82.44	97.12	89.18	71.53	78.53	95.53	91.46	93.45	94.85	89.85

3.6.3 Qualitative Results

One of the key properties of the proposed method is the sentiment word distributions for specific domains. Rank-LDA leverages on the rating scale assigned to reviews to learn a structured and generative model that represents the entire domain. This generative quality of the model, guarantees that words are represented by probability distributions across the entire range of sentiment levels. Figure 23 depicts examples of sentiment word distributions. In these figures the conditional probability density functions for each word is presented. We selected a sample of sentiment words to illustrate the probability of using a word given the sentiment level.

In Figure 23 the first two graphs illustrate the sentiment word distributions for the IMDb domain. The words *love* and *excellent* are general sentiment words that are used from a mid-range to a top-level sentiment value. However, it is interesting to note that in this domain the domain-specific sentiment word *oscar* tends to be only used to express a highly positive sentiment. On the other hand, the second graph illustrates words that are mostly used to express negative sentiment. We note that the sentiment word *watch* is used across the entire range of sentiment expressivity. This is an important feature, because the RLDA does not categorize a word as neutral (or positive/negative), instead it creates a fine-grain model of how likely is this word to occur at different sentiment levels. This is a critical feature to learn more elaborate sentiment word-interactions and to build more effective opinion retrieval systems. In the third and fourth graphs we turn our attention to the sentiment word distributions in the TripAdvisor dataset. In this domain we observed an interesting phenomena: the most positive words were quite general and not highly domain-specific. However, this was not true for the most negative sentiment word distributions: the word *dirty* is highly relevant in this domain (for obvious reasons), but the words *carpet* and *smell* are highly relevant because they are key for this particular domain. In Table 9 and Table 10 we observe the sentiment words with highest and lowest sentiment weight in the IMDb and Tripadvisor datasets, respectively. This illustrates the generative quality of the RLDA model in which our method captures both general and domain-specific sentiment words, thereby generating adequate lexicons. Moreover, the words observed in the graph *friendly*, *helpful* and *fantastic* are mostly used to express positive sentiments, but they also occur in negative sentiment with other words (not or but).

Table 9: Top sentiment words detected with the Rank LDA sentiment lexicon extracted from IMDb data.

(a) Top positive sentiment words for IMDb.

amaze	emotion	great	always	wonderful
excellent	fantastic	strong	along	heart
perfect	truly	although	screen	bring
beautiful	viewer	michael	awesome	season
best	role	favorite	stand	perfectly
brilliant	play	definitely	enjoy	finally
performance	oscar	throughout	experience	believable

(b) Top negative sentiment words for IMDb.

money	aamir	terrible	nothing	attempt
preity	waste	shanti	ghajini	horrible
bore	bad	dutt	abhishek	suppose
shetty	sanjay	sgu	mj	bachchan
srk	worst	nancy	arjun	roshan
even	awful	tudor	stupid	aishwarya
crap	quantum			

Table 10: Top sentiment words detected with the Rank LDA sentiment lexicon extracted from TripAdvisor data.

(a) Top positive sentiment words for TripAdvisor.

full	shop	value	fun	definitely
great	helpful	highly	especially	bus
excellent	location	station	fabulous	quiet
wonderful	de	friendly	special	fantastic
love	best	recommend	modern	spacious
comfortable	amaze	super	easy	
perfect	lovely	enjoy		

(b) Top negative sentiment words for TripAdvisor.

tell	charge	pay	stain	desk
call	worst	reservation	poor	smell
dirty	finally	told	awful	ask
bad	another	carpet	rude	sheet
manager	terrible	already	someone	toilet
say	horrible	credit	believe	management
never	move	happen		

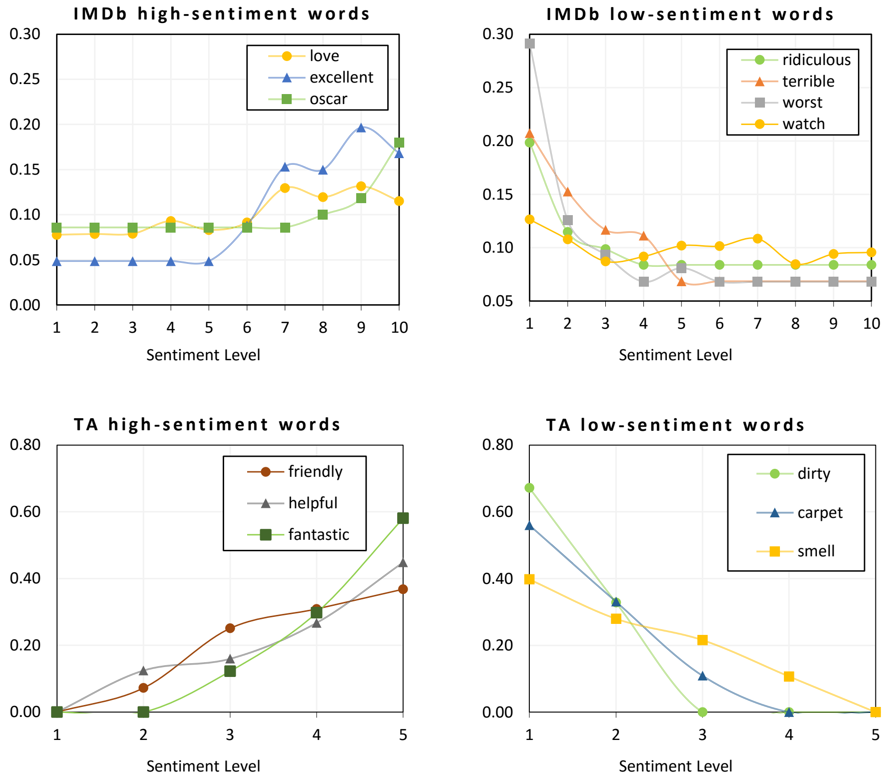
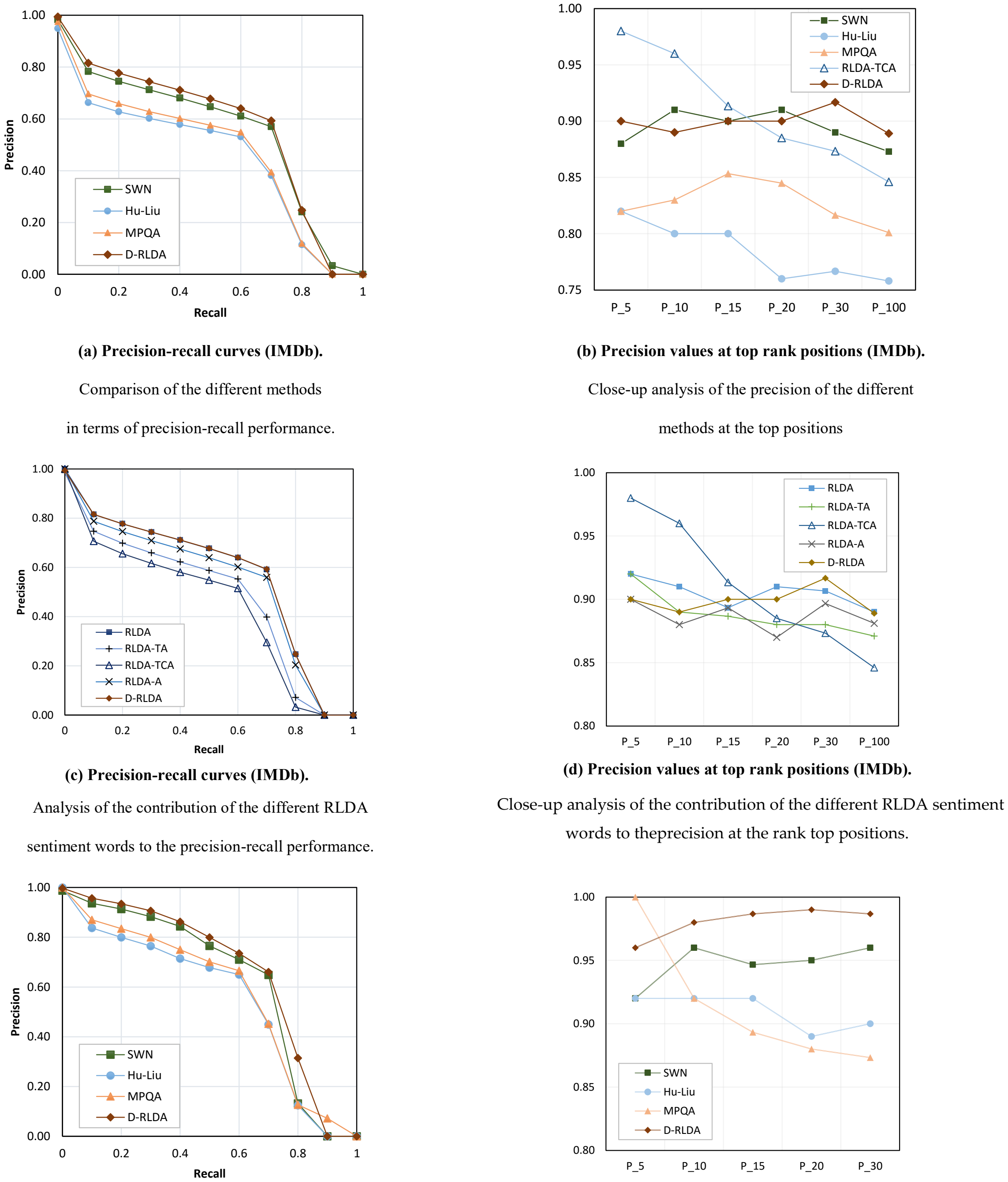


Figure 23: Sentiment word distributions for the datasets IMDb and TripAdvisor (TA).



(a) Precision-recall curves (IMDb).

Comparison of the different methods in terms of precision-recall performance.

(b) Precision values at top rank positions (IMDb).

Close-up analysis of the precision of the different methods at the top positions

(c) Precision-recall curves (IMDb).

Analysis of the contribution of the different RLDA sentiment words to the precision-recall performance.

(d) Precision values at top rank positions (IMDb).

Close-up analysis of the contribution of the different RLDA sentiment words to theprecision at the rank top positions.

(e) Precision-recall curves (TA).

Comparison of the different methods in terms of precision-recall performance.

(f) Precision values at top rank positions (TA).

Close-up analysis of the precision of the different methods at the top positions.

Figure 24: Retrieval performance of the different methods. The top row concerns the IMDb dataset. The middle row is also on the IMDb dataset, but with restricted RLDA lexicon (e.g., no actor names, no movie names). The bottom row concerns the TripAdvisor dataset.

3.6.4 Sentiment-anchors

In this section we aim to understand people's opinions, and opinions influence in named entities sentiment weight. Therefore we are exploiting named entities and sentiment words relations. In Table 10, we observe the sentiment words with highest and lowest sentiment weight in the IMDb dataset. Beyond generic sentiment words such as *amaze* and *waste*, domain-specific sentiment words are also depicted as sentiment words, for instance *oscar* and *stain* (Table 9 and Table 10). Additionally, in these tables we observe sentiment words that go beyond the traditional sentiment words (i.e. the named entities *michael* and *aishwarya*). This is also present in the TripAdvisor data (Table 10) but with different part-of-speech words and in different sentence types. For example, a highly positive review would be less likely to mention the word *carpet*, *toilet* or *management*.

The precision-recall curves for the IMDb-Extracted and TripAdvisor datasets are shown in Figure 24 (a), (c) and (e) graphs. The graphs (a) and (e) present the dictionary-based sentiment lexicons precision-recall curves for the IMDb-Extracted and TripAdvisor dataset, respectively. Figure 24 (c) presents precision-recall curves for the three reductions performed to the RLDA sentiment lexicon (as described in Section 3.5.5). These reductions respond to RLDA-A, RLDA-TA and RLDA-TCA. This graph provides a very clear illustration of the intuition behind the importance of the named entities in sentiment analysis classification. In Figure 24 (b), (d), and (f) graphs we observe the results that correspond to the precision at the top P@5, P@10, P@15, P@20, P@30 and P@100 retrieved reviews. Named entities are frequently discussed in users' reviews and we notice that the model performance improves when the number of named entities increases.

Interestingly, in Figure 24 (b) and (d) the highest precision is attained with RLDA-TCA. Keeping in mind that RLDA-TCA sentiment lexicon results from removing the names of actors, movies and characters, the precision obtained with this sentiment lexicon clearly decays as we look at a larger number of retrieved documents. Therefore, we can reason that for RLDA-TCA is a sentiment lexicon with domain-specific words and presents comparable results to the best sentiment lexicons, however as the number of retrieved reviews increase its noticeable the importance of sentiment bearing named entities for this sentiment retrieval task. While RLDA-TCA lowers its performance with the number of retrieved reviews, RLDA-A (it was only removed the names of actors) presents a precision constant over the different metrics and comparable to the best sentiment lexicon. It is important to notice that RLDA-A contains 45.6% of the sentiment words within the RLDA lexicon. Furthermore, as it is noticeable in Figure 24 (d) at P@20 RLDA-

A performs considerable worse than RLDA and D-RLDA which re-enforces our intuition that named entities (such as actor's names) enclose relevant sentiment weights. In the next chapter the importance of sentiment anchors, hence entities that enclose sentiment, will be further investigated in as sentiment classification and entities' reputation problem.

3.7 Summary

Sentiment words are an essential instrument in sentiment analysis. Positive and negative sentiment words reflect what the sentiment in a review, sentence or a named-entity is about. The importance of detecting such words has led researchers to propose different techniques to compile sentiment lexicons. In this chapter is investigated how to detect such words and domain related idiosyncrasies where specific sentiment words are common.

In a sentiment based method we analyse the dimensionality reduction of the LDA hidden structure for the extraction of a sentiment word lexicon. The lexicon was evaluated in the task of opinion raking and sentiment analysis on datasets spanning two different domains (movie and hotel) which contained 367,691 and 123,678 different words, respectively. We show improvements of the proposed method over the baselines, and notice that the improvements are related to the domain specific words and sentiment word distributions inferred by the Rank-LDA method. It was particularly important to notice that a given sentiment word is not assigned to a fixed value but a probability distribution instead. The analysis of the sentiment word distributions allowed to notice an interesting phenomena: the word *love* presents a mid-range sentiment level distribution, however the domain specific sentiment word *oscar* presents a highly positive sentiment distribution. Furthermore we find important to remember that Rank-LDA does not categorize a word as neutral, positive or negative, instead creates a fine-grain model of how likely this word occurs at different sentiment levels, and this is a critical feature to learn more elaborate sentiment word interactions and to build more effective opinion retrieval systems.

Finally, the work presented in this chapter was published at:

Peleja, F. and Magalhães, J. 2015. "Learning Sentiment Based Ranked-Lexicons for Opinion Retrieval." In *Proceedings of the 37th European Conference on Advances in Information Retrieval (ECIR)*, pages 435-440, Vienna, Austria: Springer. doi: 10.1007/978-3-319-16354-3_47.

Peleja, F. and Magalhães, J. 2015. "Learning Ranked Sentiment Lexicons" In *Proceedimgs 16th International Conference Computational Linguistics and Intelligent Text Processing (CICLing)*, pages 35-48, Cairo, Egypt: Springer. doi: 10.1007/978-3-319-18117-2_3.

4

A Linked-Entities Reputation Model

In this chapter we address the problem of observing users' opinions with the aim of identifying how they influence entities reputation. A related area of study is reputation management which focus on monitoring the reputation of global public opinion about an individual, brand or product. Social Web allows to identify early warnings of reputation shifts and content that influences the reputation of an entity (Petasis et al., 2014). As a result reputation analysis is naturally associated to content that targets or mentions specific entities. Such content is also explored in sentiment analysis problems (chapter 3). The proposed method will take into consideration a sentiment lexicon that includes words that characterize a general sentiment that is commonly used to express an opinion about a given entity. In many cases entities are themselves part of the sentiment lexicon creating a loop from which it is difficult to evaluate their reputation. Additionally, it is not uncommon to find reviews where multiple citations to actors or movies occur. Some entities (e.g., the actors or movie titles) become so important that turn into a synonym of high-quality (or low-quality). As a consequence, these entities represent a domain reference that is vastly cited in the context of an esteemed or disdained example.

The overall sentiment that targets an entity is intrinsically linked to the reputation analysis of the respective entity. In a social media context opinions about different entities are often expressed in an informal manner with the usage of slang words and other language specificities. To deal with this data, formal dictionaries of sentiment words are less appropriate than corpus-based sentiment lexicons (Chen et al., 2012). In this chapter we capture relevant sentiment words and weight them according to their sentiment relevance. The proposed method is able to detect entities' distributions through a generative model that models how user sentences are generated for the same entity at different sentiment levels.

Reputation analysis for entities has been a topic of recent research. Go et al., 2009 work used well-known machine learning algorithms (Naïve Bayes, Maximum Entropy and SVM) to classify the overall sentiment of Twitter messages towards specific keywords, representing various entities preferences, such as movies, famous people, locations and companies. Later, Chen et al., 2012 proposed a constrained optimization problem to extract sentiment polarity from tweets that target movies and people. Chen et al., 2012 lexicon contains both formal and slang words to better accommodate Twitter vocabulary. Their lexicon is built by collecting words from dictionaries such as SentiWordNet (Esuli and Sebastiani, 2006) and Urban Dictionary[18]. Krauss et al., 2008, in turn, used a sentiment analysis approach on IMDb discussions to predict Oscar nominations. In the present thesis we argue that static-lexicons are too coarse-grain and, as a consequence, fail to capture relevant sentiment words (among them entities) that target numerous entities.

In the movie domain, users write reviews with rich information about their preferences. Reviews include a rating and sentences that reflect opinions about the different aspects of the movie, such as characters, actors, or related movies. In such sentences sentiment words are used to express opinions. For example, in the sentence bellow we present a review about the movie *Batman Begins*:

> *"I just came back from a special screening of* ***Batman Begins*** *and I must say this is the* ***best movie*** *I have yet seen this year."*

In this example, the word *best* is a sentiment word that indicates a positive opinion about the movie *Batman Begins*. In users' reviews similar sentences may refer to the same or other entities (i.e. actors). Nonetheless the reputation of an entity is not only influenced by such explicit sentences. Consider, the following sentence from a review about the movie *Iron Man*:

> "Add to that some of the **best** features of **Robocop**, **Batman Begins** and **Terminator II**, and you have one of the more **satisfying** comic-books-turned-blockbuster …."

This sentence illustrates that the cited movies (*Robocop, Batman Begins* and *Terminator II*) and the sentiment words *best* and *satisfying* contribute to the positive reputation of the reviewed movie.

[18] www.urbandictionary.com

In this chapter, we address the following linked task: given user's opinionated reviews, we wish to find named-entities mentions that implicitly affect the entity reputation. To this aim we propose a three-step approach: first, a method that jointly extracts and affects a sentiment weight to entities and domain sentiment words; second, to identify entities associations we exploit entities cross-citations; and third, a graph-based method is introduced to update the entities reputation through an iterative optimization technique.

The contributions of this chapter are two-fold:

- A sentiment graph that represents entities and relations that exist in the corpus. The sentiment graph is represented in a pairwise Markov Network and entities are characterized by the sentiment words used to describe it.
- The sentiment lexicon and the overall sentiment words towards entities is modelled as a ranking problem which is an ideal approach to the problem of reputation analysis.

4.1 Ranking Linked-Entities

In this section, we introduce our entity reputation graph which aims to compute an entity reputation by observing the entities and sentiment words that "link" to an entity. Figure 25 represents the problem at-hand: in this undirected graphical model entities correspond to the grey nodes and the white nodes correspond to sentiment words. In contrast to a graph where the links (or edges) point into a direction (directed graph), in an undirected graph links are bidirectional. Formally an undirected graph is defined as $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is a set of nodes and $\mathcal{E}$ is a set of edges which are unordered pairs of elements of $\mathcal{N}$.

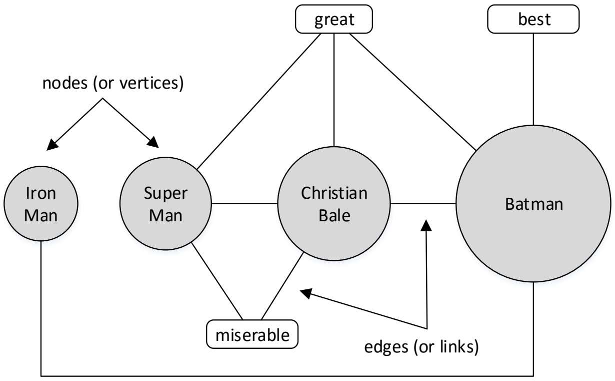


Figure 25: Sentiment words and entities that link to the entity *Batman*.

When attempting to link opinion utterances to a given entity it should be kept in mind that there are two types of opinions: regular opinions and comparative opinions (Jindal and Liu, 2006). A regular opinion expresses an opinion about a particular aspect or entity, e.g., "*Brilliant effects in The Shining.*", where there is a positive sentiment expressed by *brilliant* on the aspect *effects* of the entity *The Shining*. While a comparative opinion compares entities based on their aspects, e.g., "*The Shawshank Redemption and To Kill a Mockingbird are the best movies I have ever seen.*" which compares the movies *Shawshank Redemption* and *To Kill a Mockingbird* based on their overall quality. Comparative opinions produce links between numerous entities (Figure 26).

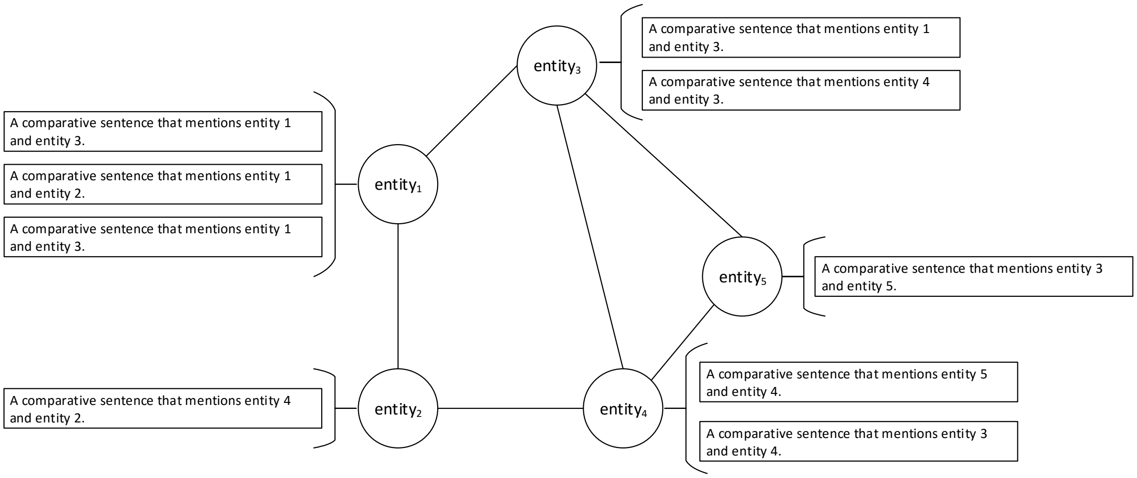

A comparative sentence that mentions entity 1 and entity 3.
A comparative sentence that mentions entity 4 and entity 3.
entity$_3$
A comparative sentence that mentions entity 1 and entity 3.
A comparative sentence that mentions entity 1 and entity 2.
A comparative sentence that mentions entity 1 and entity 3.
entity$_1$
entity$_5$
A comparative sentence that mentions entity 3 and entity 5.
A comparative sentence that mentions entity 4 and entity 2.
entity$_2$
entity$_4$
A comparative sentence that mentions entity 5 and entity 4.
A comparative sentence that mentions entity 3 and entity 4.

Figure 26: Graphical representation of entities in comparative opinions.

A comparative opinion is of the form (E_1, E_2, sw), where E_1 and E_2 are the entities being compared using a sentiment word sw to express their sentiment about the relation of these entities. The example "*The Shawshank Redemption and To Kill a Mockingbird are the best movies I have ever seen.*" will be expanded in the following tuple:

$$(The\ Shawshank\ Redemption, To\ Kill\ a\ Mockingbird, best).$$

In case there are additional sentiment words or entities in the comparative opinion, the number of tuples for that sentence will expand.

In our method we first compute a fine-grain lexicon of sentiment words that best capture the level of user satisfaction, then determine the reputation of an entity for which we observe the domain influence in the entities' reputation. The proposed ranking linked-entities framework is divided into three parts:

- First, compute a ranked sentiment lexicon to determine the sentiment of each individual word, expression and entity in the corpus. This first step builds on the RLDA method proposed on the previous chapter.
- Second, to infer the graph structure we identify the sentiment relations between entities and the respective relevance.
- Finally, a sentiment graph is used to iteratively compute entities reputation. This algorithm explores links between entities, while the second step explored co-occurrence and sentiment associations.

In the following sections the reputation analysis framework is formalized and the computation algorithm of the linked entities sentiment is presented.

4.1.1 Entity Reputation Graph (ERG)

Hatzivassiloglou and McKeown (1997) introduced a method to infer polarities of words. Words are represented in a graph as nodes and links between nodes denote some type of relationship the nodes share. Inspired by Hatzivassiloglou and McKeown (1997) definition, Scheible and Schütze (2012) present an example of a graph using sentiment words in each node (Figure 27). We adopt a similar approach but the links denote a different type of relationship. While Hatzivassiloglou and McKeown (1997) word-to-word links are generated from *and* and *but* connectors ERG links are generated from sentiment relations that are identified by the proposed method. The observed sentiment relations occur between sentiment words and entities or entities and entities.

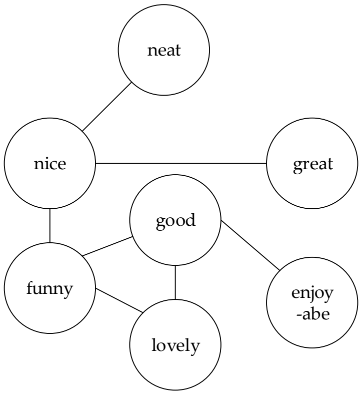


Figure 27: Word graph (Scheible and Schütze, 2012).

ERG is derived from a topic-specific PageRank approach (Page et al., 1998). The sentiment words and entities links are derived from a set of representative sentiment levels which can be interpreted as topics. We aim to capture more accurately the notion of importance of each entity with respect to a particular sentiment level (e.g. a representative sentiment level/topic to evaluate entities reputation). The intuition is that a good authority entity will be pointed by many sentiment words and entities. This mutual reinforcement relationship of sentiment words and entities allows to rank the reputation of each entity. In a similar approach, Zhang and Liu (2011) formulate an algorithm that uses a graph method to extract resource words and phrases that are relevant for a sentiment analysis task. This strengthens the intuition that a method derived from PageRank algorithm can be adapted to a sentiment analysis problem.

To rank entities reputation it is fundamental to capture the entities that are mentioned as esteemed and loathed references. To this end, the nodes in the reputation graph G link to other entities as also sentiment words. PageRank aims to find web pages that are authorities and computes the link-based "authority-strength" of a page by its value in the dominant left eigenvector $\overrightarrow{r}$ of the transition probability matrix M of the graph (Kleinberg, 1998; Scheible and Schütze, 2012). In our reputation graph the rank vector$\overrightarrow{r}$ contains the initial reputation values and its value is updated by the following method,

$$\vec{r} = \vec{r} \times M + (1 - \alpha) \tag{28}$$

where α is a damping factor. For a given entity i the node distribution of the vector t_i is defined as the sum of the links between sentiment words and entities that co-occur with entity i. From this vector we construct the matrix M and apply the abovementioned equation to obtain the final rank reputation vector $\vec{r}$ – the final value for the reputation of each entity.

There are several advantages of opting for a reputation graph based on PageRank algorithm (L. Zhang et al., 2010; Zhang and Liu, 2011). PageRank uses a recursive scheme similar to HITS algorithm. However, unlike HITS algorithm PageRank is independent of a user's query. The original idea beyond HITS is that an important page is pointed by many other pages. For a web page i with an authority score a_i and hub score h_i the scores will be repeatedly update until converge to some k. Here, an authority value represents the sum of the hub values that point to a page and a hub value is the sum of the authority values of the pages it points to. The authorities and hubs vectors are normalized every time the scores get updated, as they depend of each other's equations. Similar to the HITS algorithm PageRank calculation uses the power of iteration to find the dominating eigenvector of the authority matrix where each dimension corresponds to the PageRank of a page. These properties are a main strength for providing more relevant authority nodes and the process of applying the power of iteration entails a smaller computational load.

Another two main advantages of adapting PageRank to our problem is the propagation and attenuation properties (Lee et al., 2011). The propagation property is that the connection (relatedness) of the nodes propagates through the graph links, and the attenuation property is that the propagation strength decreases as we propagate further into graph from the starting node. With the propagation property one can navigate in the graph from a start node and go further from this node also, with the attenuation property measure the relatedness of the current node and start node. The start node is randomly selected and from that node the reputation nodes are updated. Hence, while navigating further in the graph we update the reputation of the nodes according to the relatedness to the starting node (relatedness property) as we also weight the reputation influence according to the node proximity to the starting node (attenuation property).

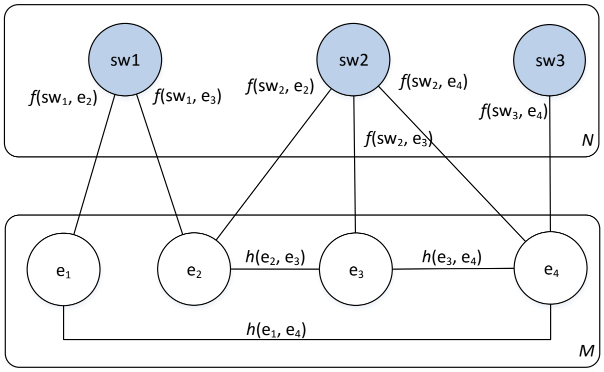

sw1
sw2
sw3
$f(\mathrm{sw}_1, \mathrm{e}_2)$
$f(\mathrm{sw}_1, \mathrm{e}_3)$
$f(\mathrm{sw}_2, \mathrm{e}_2)$
$f(\mathrm{sw}_2, \mathrm{e}_4)$
$f(\mathrm{sw}_3, \mathrm{e}_4)$
$f(\mathrm{sw}_2, \mathrm{e}_3)$
N
e_1
e_2
$h(\mathrm{e}_2, \mathrm{e}_3)$
e_3
$h(\mathrm{e}_3, \mathrm{e}_4)$
e_4
$h(\mathrm{e}_1, \mathrm{e}_4)$
M

Figure 28. Entity reputation graph: the graph factors correspond to the connections entity-sentiment words (label "f") and entity-entity (label "h").

Figure 28 shows the ERG graphical model. The reputation graph incorporates both entities and sentiment words information in a single heterogeneous graph, where nodes correspond to entities or sentiment words. Formally, the graph consists of a set of vertices (nodes) $\mathcal{N}$ corresponding to the extracted entities and sentiment words, and a set of edges $\mathcal{E}$ representing the links between entities and sentiment words. The edge set $\mathcal{E}$ consists of links between entities and sentiment words in which an edge represents a link between the entities e_i and e_j as $h(e_i, e_j)$ or, a link between a sentiment word sw_j and an entity e_i as $f(sw_j, e_i)$. To this end, ERG aims to determine entities reputation and how entities reputation evolve in the reputation graph by weighting these links. Moreover, edges between sentiment words will not be observed since we believe these are misleading for the sentiment weight that targets entities. One must keep in mind that the sentiment weight and polarity for each sentiment word is inferred by the corpus-based sentiment analysis method proposed in chapter 3. And, to iterate the sentiment word value we might lose its sentiment value in the domain. For this reason, sentiment words' links will not be included in the ERG graph.

4.1.2 Reputation Calculation

Given the ERG graph structure and the different graph factors we aim to assign each entity e_i a reputation label $rp_i \in \{pos, neutral, neg\}$. In our graph structure we made the assumption that the entity reputation can only be influenced by its neighbouring entities and sentiment words links. The ERG graph can be seen as a pairwise Markov Network (Taskar et al., 2002; Wang et al., 2011) in which the theory behind Markov Networks is that for any start node the power iteration method applied to the transition matrix M – entities and sentiment words links – will converge to a unique positive stationary vector, which in our model will be the entities' reputation vector. The ERG graph involves a set of entities E where $rp(e_i)$ is the reputation of a given entity e_i. ERG follows the model of the random surfer to compute the reputation of each linked-entity. Formally, the reputation of an entity is computed iteratively as follows,

$$rp(e_i) = f_0(e_i) + \sum_{e_j \in \{N(e_i)\}} \frac{rp(e_j)}{\#\{N(e_j)\}} \cdot \psi(e_i, e_j) \cdot h_{i,j}(e_i, e_j) \tag{29}$$

where the first part of the expression concerns the reputation assigned by the sentiment expressions that target the entity, and the second part concerns the revised reputation assigned by explicit citations, in comparative sentences or as a reference citation. In the next section we will detail the computation of each part of the reputation expression.

Algorithm 2 details how the reputation of the linked entities is computed in an iterative approach. To initiate the iteration, all entities receive a sentiment weight based on the RLDA method – in the next Section the computation of $slda$ will be detailed. In Algorithm 2, the reputation $rp(e_*)$ of each entity is updated according to the reputation of the neighbouring entities and by the sentiment of linked sentiment words. Note that in this formulation the weight of a sentiment word is not affected by the ERG. The variable $\#(e_i, e_j)$ refers to the number of times entities e_i and e_j co-occur and $\#e_*$ to the entity frequency in the corpus. The algorithm stops iterating when the reputation labels for all entities stabilize.

Algorithm 2: Entities Reputation

```
Input:
  Graph ERG
  $RLDA(sw_*) \leftarrow$ The $RLDA$ values of each sentiment word
  $RLDA(e_*) \leftarrow$ The $RLDA$ value of each entity
Output: Reputation label for each entity $e$
begin
  foreach $e_i \in E$do
    foreach $e_j \in N(e_i)$do
      $h(e_i, e_j) \leftarrow slda(e_i, e_j) + \left(RLDA(e_i) + RLDA(e_j)\right)/2$
      $\psi(e_i, e_j) \leftarrow \#(e_i, e_j)/\left(\#(e_i) + \#(e_j)\right)$
  repeat
    foreach $e_i \in E$do
      $rp(e_i) \leftarrow f_0(e_i) = \sum_{sw_n \in E \cup SW} f_n(e_i, sw_n)$
      foreach $e_j \in N(e_i)$do
        $rp(e_i) \leftarrow rp(e_j) \cdot \psi(e_i, e_j) \cdot h(e_i, e_j)$
  until all $rp_{i \to j}(e_j)$ and $rp_{i \to n}(e_n)$ stop changing;
return $rp$
```

4.1.3 Graph Structure: Entities and Sentiment Words Links

To infer the graph structure (i.e., the nodes and links) where nodes are entities or sentiment words and links represent the connection between entity-entity and entity-sentiment word. An important step is to determine how to perform entities extraction. One possibility is to extract movie metadata from IMDb – title, actors, characters and directors. An alternative, which does not rely on static metadata, is to automatically extract relevant named entities by using tools such as NLTK Named-Entities and Relation extractor[19]. This tool captures entities that are referred by their alias or by jargon that is used to refer to that entity. For example: *"lotr"* for the movie *Lord of the Rings* and *"spidey"* for the *Spider Man* movies. In the present approach we have used NLTK tool to capture named entities as unigrams and bigrams (e.g. *Alfred Hitchcock* and *Hitchcock*). Furthermore, users' opinions influence entity's reputation (Li et al., 2012) and for this reason the proposed method leverages on a sentiment lexicon that includes general sentiment words and entities that characterize the overall sentiment towards the targeted entity.

The initial step is to capture for each entity (actors, characters or movie titles) the other entities and sentiment words that co-occur most frequently with that specific entity. In a formal definition: for the task of generating the entity-reputation graph, there is a set of entities $E =$

[19] http://www.nltk.org

$\{e_1, e_2, \dots, e_m\}$ where each entity e_i is associated with a set of entities $N = \{e_1, e_2, \dots, e_n\}$, $e_i \notin N$, and a set of sentiment words $SW = \{\, sw_1,\ sw_2, \dots,\ sw_p\}$. The edges between two entities (e_i, e_j) are expressed as follows,

$$h(e_i, e_j) = slda(e_i, e_j) + \frac{RLDA(e_i) + RLDA(e_j)}{2}, \tag{30}$$

where $slda(e_i, e_j)$ is the semantic association between two entities (we will return to this function later) and $RLDA(e_i)$ is the sentiment weight given by the ranked sentiment lexicon (chapter 3). This weight is exclusively affected by the topic modelling algorithm that embeds the rating information. The edges between an entity and a sentiment word (e_i, sw_n) are represented as follows,

$$f(e_i, sw_n) = \frac{RLDA(e_i) + RLDA(sw_n)}{2}. \tag{31}$$

Entities and sentiment words that do not co-occur will have a link weight $f(e_i, sw_n)$ equal to zero. Additionally, this expression is cumulative meaning that an entity is affected by a full set of sentiment words in the many sentences where e_i is mentioned:

$$f_0(e_i) = \sum_{sw_n \in E \cup SW} f_n(e_i, sw_n) \tag{32}$$

where $f_0(e_i)$ quantifies the overall reputation that an entity receives from the linked sentiment words.

The associations between entities are identified by analysing the sentences where entities co-occur, hence, sentences that contain citations about another entity. The explicit co-occurrences are formalized as

$$\psi(e_i, e_j) = \frac{\#(e_i, e_j)}{\#(e_i) + \#(e_j)}, \tag{33}$$

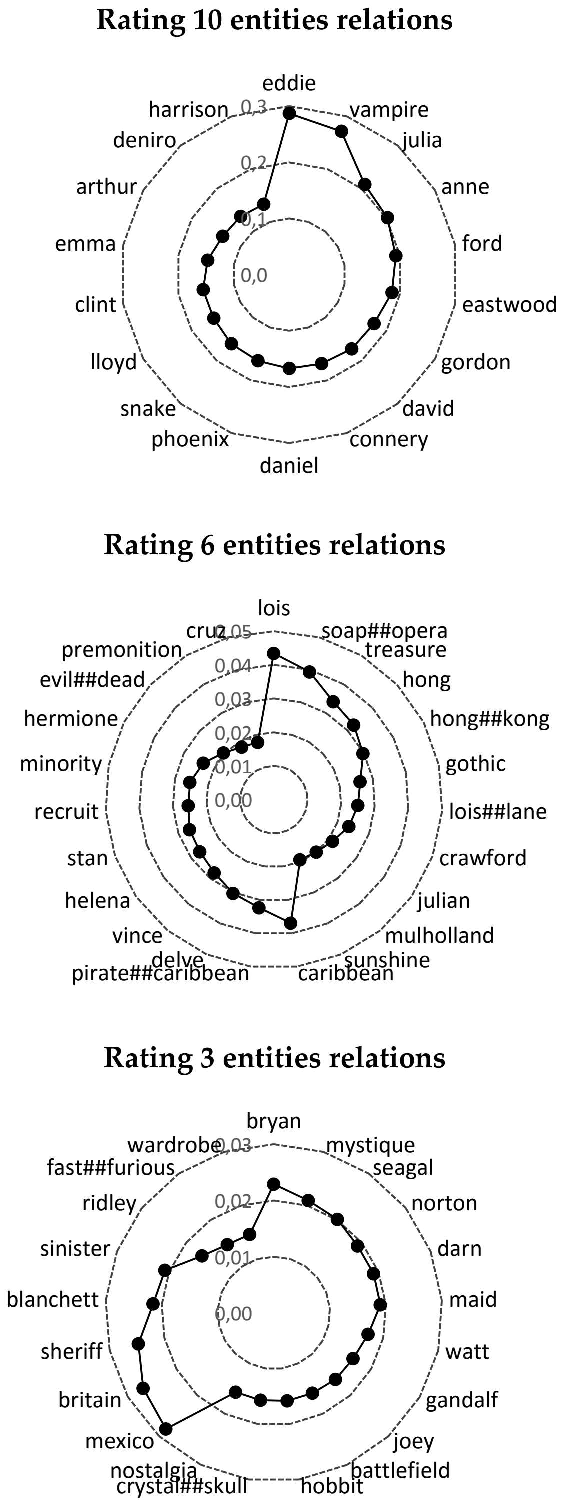
where $\#(e_i, e_j)$ is the number of times the entities e_i and e_j co-occur together and $\#(e_i)$ is the number of times an entity occurs individually.

In chapter 3 we notice the presence of sentiment relations between sentiment words and entities in users' reviews which, consequently, is reflected in our Rank-LDA sentiment lexicon. Following these observations, in the present chapter, we analyse the shallow relations between entities in users' reviews. To this end, a new Rank-LDA model is computed with users' reviews which are solely represented by its entities. The RLDA model observes entities that take place in reviews from the same and different sentiment levels. This process models entities that are semantically related by sentiment level. Figure 29 illustrates the output of this process: the relations among entities at three sentiment levels. For each sentiment level – rating level – the entities are not all connected to each other, hence, the entities network is not fully meshed. These plots show that specific entities reveal a higher relevance according to the sentiment level, e.g. "lois" for Rating 6 or "mexico" for Rating 3. In particular, according to the sentiment level, it gives an insight on how entities are semantically related to each other. For example, in Rating 6 the entities "lois", "caribbean", "lois lane" and "sunshine" share a stronger semantic relation. Note that some entities might not even be considered by Rank-LDA as their presence is residual. To this end, for each sentiment level we observe the different entities semantic associations. This analysis is taken at different sentiment levels and allows to detect entities that share a stronger semantic relation (e.g. "lois" and "caribbean" in Rating 6), which will have an impact on the initial weight that is given in ERG reputation graph sentiment links.

To estimate the weight given to the observed semantic relation between entities e_i and e_j we compute the probability of a sequence of words and its hidden topics. Given the LDA model $p(w,z) = \int p(\theta) \cdot \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n)\, d\theta$ where θ is the random parameter of a multinomial over topics, the semantic relation of two entities is given by:

$$slda(e_i, e_j) = \sum_{r \in R} \sum_{z \in Z} \left(p(e_i, z) + p(e_j, z) \right), \exists e_i, e_j \in \mathrm{z} \tag{34}$$

The two proposed formulations for quantifying entities relations, $\psi(e_i, e_j)$ and $slda(e_i, e_j)$, are the basis to construct the entities graph with sentiment level information embedded on it (captured by the Rank-LDA hidden topics).



Rating 10 entities relations

Rating 6 entities relations

Rating 3 entities relations

Figure 29. Entities semantic associations for different sentiment levels captured by RLDA methodology. Each graph shows a set of entities that are probabilistically linked.

4.1.4 Relation to Previous Work

Now that we have presented our method in detail, we will briefly discuss how the proposed entities reputation graph differs from similar existing methods.

The analysis of an entity reputation starts by identifying sentiment associations between words, which can also be other entities, and the respective entity. Hu and Liu (2004a) applies NLP techniques to define a set of association rules that aim to extract products aspects (i.e., product characteristics). Here, it is introduced the intuition that sentiment words and products' aspects are linked. With a feature-based summary a potential customer observes customers influence on a specific products' aspect reputation. Table 11 shows the summary of feature (aspect) *picture* and the product (entity) *digital camera*. In Hu and Liu (2004a) individual aspects are identified to evaluate their role in the improvement or deterioration of the products' reputation. Unlike Hu and Liu (2004a) ERG inspects sentiment words and entities contributions to the overall reputation of an entity (product). To this end, ERG does not intend to identify individual contributions of different aspects but the full contribution of different sentiment words and entities.

ERG implements an approach that identifies a domain specific sentiment word lexicon. This lexicon encloses products characteristics that are semantically related to the product or to the sentiment words expressed by the users. Hu and Liu (2004a) applies associating mining techniques to identify products' aspects. However, we argue that these techniques fail to capture semantically associated words that are inherent to the latent topics layers and not visible in an association mining algorithm such as CBA (Liu et al., 1998).

The intuition that entities are sentiment related is also mentioned in Kim and Hovy (2004). Here, the authors define opinion holder as "*…people who hold opinions about that topic…*" To this end, Kim and Hovy (2004) make the assumption that opinion holders occur in the vicinity of opinion phrases. Hence, Kim and Hovy (2004) used a window size ruler to observe the sentiment expressed within the sentence, and a named entity tagger tool was used to identify potential opinion holders. However, we argue that this assumption does not hold for other domains (e.g. movie domain) in which it would most likely misclassify opinion targets as opinion holders. To identify opinion holders should be used a parser to identify syntactic relationships and in opinionated text users are more likely to explicitly refer to opinion targets than to opinion holders. While in users' reviews the opinion holder is frequently omitted under the assumption that the holder is the review writer, in news texts holders and targets become more diverse (Lu, 2010). Popescu and Etzioni (2005) have also noticed the sentiment relation between entities and sentiment words, where the authors employed the PMI (pointwise mutual information) metric to associate an opinion phrase (i.e., a single or n-gram sentiment word) with an entity. Previous work (Kim and Hovy, 2004; Hu and Liu, 2004a) has evaluated the sentiment that targets an entity however did not have the objective of evaluating the overall sentiment that influences entities' reputation in a specific domain.

Table 11: Summary for the feature picture of the product digital camera (Hu and Liu, 2004a).

Feature: picture
Positive: 12 • Overall this is a good camera with a really good picture quality. • The pictures are absolutely amazing – the camera captures the minutest of details. • After nearly 800 pictures I have found that this camera takes incredible pictures.
Negative: 2 • The pictures come out hazy if your hands shake even for a moment during the entire process of taking a picture. • Focusing on a display rack about 20 feet away in a brightly lit room during day time, pictures produced by this camera were blurry and in a shade of orange.

In the proposed work entities are linked in a graph which relates to PageRank algorithm (Page et al., 1998). Scheible and Schütze, 2012 have recently proposed a method for polarity sentiment analysis. The authors introduce Polarity PageRank (PPR), a method that integrates lexicon induction and lexicon application in one unified formalism. PPR links document nodes to word nodes and document nodes do not link to other document nodes. This way, Scheible and Schütze, 2012 guarantee that relationships between documents are defined by the relationships of their word links. Our method differs from PPR in two main aspects: PPR uses bag-of-words representation and, as a consequence, our method applies a more fine-grained representation of user reviews (sentiment words and entities). In PPR semantic information is significantly lost when all positional information is discarded. In addition, ERG links are identified through a sentiment analysis methodology while in PPR uses the normalized term-frequency from Salton and McGill (1986).

Efforts to explore graph analysis in sentiment analysis have been studied. For instance, Tan et al. (2011) proposed a directed heterogeneous graph (Figure 30) to improve user-level sentiment analysis in different topics (i.e. "*Obama*", "*Gleenn Beck*", "*Fox News*" and "*Lakers*"). Figure 30 illustrates Tan et al., 2011 factor graph, which is a bipartite graph that represents the factorization of a function. To this end, the factor graph is used to represent the probability distribution function of the sentiment expressed by the user' tweets about different topics. In Tan et al. (2011) graph users are represented in the nodes where both textual and social network information are incorporated. Unlike Tan et al. (2011) heterogeneous graph that adds to the graph a user-tweet link based on the sentiment label of tweets (which contains a set of words that can go up to 140 characters), the proposed reputation graph (ERG) implementation uses sentiment word level sentiment analysis which has a finer granularity. In addition in Tan et al. (2011) experiments the dataset (tweets) do not exhibit strong opinions, and to overcome this problem of labeled data the authors took the assumption that the information about Twitter users' biography would be an indication of their opinion about a given entity (i.e. "*social engineer, karma dealer, & obama lover*"). This can be misleading since in the graph structure it is assumed that tweets from these users will always be positive about a given entity (i.e. *Obama*). To resolve this limitation Tan et al., 2011 propose a conservative strategy: users' name and biography are manually annotated. Entities in ERG reputation graph does not benefit from this type of ground truth data, however we believe the method tackles a problem that is closer to users' opinions about the same or different entities.

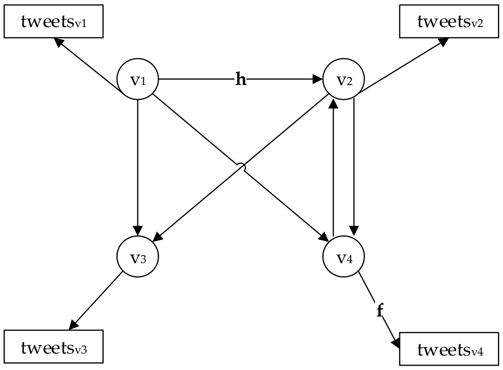


Figure 30: Example of directed heterogeneous graph. The corresponding factor graph has factors corresponding to user-tweet dependencies (label "f") and user-user dependencies (label "h") (Tan et al., 2011).

4.2 Evaluation

For evaluation purposes reviews are split at sentence level, words reduced to the same stem (to a common form) and stop words were removed. In addition, we also computed bigrams with a maximum distance of 3 words between each word-pair.

4.2.1 Datasets

IMDb-Extracted: This dataset contains 1,007,926 million movie reviews, corresponding to a total of 7,102,592 million sentences. In Section 2.4.4 is described in more detail how these reviews were extracted. Reviews are rated in a scale of 1 to 10. For evaluation purposes, the dataset is evenly split into three disjoint splits (A, B and C). Table 12 presents the detailed information about the IMDb-Extracted.

- **Subjective classification:** Following Pang and Lee, 2005 methodology, the split A is used to model a subjective classifier. The online gradient descent method from Vowpal Wabbit library was chosen for this task. To this aim, sentences from movie plots are labeled as objective and sentences from users' reviews as subjective. To build the subjective classifier model with a balanced data, subjective sentences were held out from the training phase. For the subsets B and C, 1,424,503 and 693,349 sentences were classified as objective respectively.
- **RLDA:** The sentiment lexicon RLDA is modeled using split B subjective sentences.
- **Evaluation:** Split C subjective sentences are used for evaluation purposes.

Table 12: Detailed information of IMDb-Extracted.

Split	#reviews	#sentences	#subjective sentences	#entities
A	335,975	167,074	82,537	273,081
B	335,950	2,981,996	2,288,647	950,237
C	335,976	3,953,522	2,503,976	1,348,994

4.2.2 Methodology

Sentiment classification is commonly used to evaluate sentiment lexicons (Liu, 2012). Observing the sentiment associations between sentiment and reputation, we performed a sentiment classification task to evaluate the obtained entities' reputation. To this end, we will evaluate the contribution of reputation scores in the task of detecting the polarity of subjective sentences.

K-Nearest Neighbour

K-Nearest Neighbour (KNN) algorithm is a non-parametric method that will be used for a sentiment classification task. KNN is a non-parametric method because it does not make any assumptions on the underlying data distribution. In KNN a new sample is classified according to the majority vote of its neighbours which are part of the known samples. Among machine learning algorithms KNN is part of the lazy learning classifiers, where a function is approximated locally and all computation is delayed until classification. Hence, it does not use training samples to generalize a class (no explicit training).

KNN is useful to weight the contributions of its neighbours and to measure the proximity between neighbours a common practice is to use the Euclidean distance. Alternatively, Manhattan distance is able to identify alternative routes other than diagonal distance (Euclidean distance). As the observed clusters (sentiment levels) do not tend to form hyper-spherical of equal size – which is the Euclidean distance assumption – we believe Manhattan distance fits best to the problem.

4.2.3 Baselines

To perform a comparative evaluation of the Entity Reputation Graph (ERG) with previous work, he following sentiment lexicons were selected as benchmarks:

- SentiWordNet (Esuli and Sebastiani, 2006): this lexicon was built with a semi-automatic method where manual effort was used to curate some of the output. Is used the top 2,290 positive words and the bottom 4,800 negative words corresponding to a sentiment weight greater than 0.6 (on a scale of 0.0 to 1.0).
- MPQA (Wilson et al., 2005): this lexicon provides a list of words that have been annotated for intensity (weak or strong) in the respective polarity – positive, negative or neutral. The lexicon was obtained manually and an automatic strategy is employed afterwards. Contains 2,718 positive and 4,912 negative words.
- Hu-Liu (Hu and Liu, 2004a): this lexicon contains no numerical scores. Based on the premise that misspelled words frequently occur in users' reviews these words are deliberately included in the lexicon. The lexicon contains 2,006 positive and 4,683 negative words.

4.2.4 Evaluation metrics

Sentiment classification algorithm is evaluated according to its performance in classifying reviews' sentences per each sentiment level. To evaluate this task we compute accuracy as follows,

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{35}$$

where TP is the number of true positives, TN is the number of true negatives, FP the number of false positives and FN is the number of false negatives. Hence, accuracy measures the proportion of true results among the total number of observed samples.

4.2.5 Human Evaluation

Humans judgements are obtained in two ways: first, human assessors judge if a specified sentiment word is relevant to characterize the entity reputation (Table 13 presents examples of sentences from the survey). And, second, human assessors judge if the entity described by different sentiment words entails a positive, negative or neutral influence in the entity reputation. The judgments were collected through the crowdsource platform CrowdFlower[20]. Each participant was asked to judge up to 300 from 3,000 sentences and 3,000 sentiment words-entities pairs. The average response for each annotation was calculated as the coherence score for the gold-standard. Furthermore, to ensure reliability and avoid random answers it is included a number of golden questions with manually predefined answers. Annotations from participants that failed to answer these questions correctly were removed.

Table 13: Sentence examples used in the crowdsource evaluation.

Sentence	Entity	RLDA SW
Having seen a few Hitchcock movies in my day,I cannot believe Zemeckis thought this script qualified.	Hitchcock	cannot believe
Seagal is the only man standing between blah and blah and blah de blah blah.	Seagal	blah blah
If there was an excellent Batman, this is the real deal.	Batman	excellent
Anthony Hopkins did a great job as Diego de la Vega/Zorro.	Anthony Hopkins	great

To ensure a high-quality of the obtained labels target workers were limited to countries where English is the main language and test questions were used to filter unreliable workers. Snow et al. (2008) show that an average of 4 non-expert workers are able to match the quality of expert annotators it was collected judgments from 5 different workers for each sentence (in order to best emulate expert labelling). From the obtained results is selected the sentences with an agreement

[20] http://crowdflower.com

of at least 70% and all the sentences were labelled as very positive or very negative, as it is believed that these express a stronger sentiment value. The inter-annotator agreement is measured as the average of the Spearman correlation between the set of scores of each survey annotation and the average of the other annotators' scores.

4.3 Experiments

In this section is detailed two crowdsource tasks: entity reputation and sentiment analysis. More specifically, crowdsource techniques will evaluate if sentiment words that target entities have the ability to influence their reputation. And, if the proposed method has the ability to correctly identify and weight sentiment words. Finally, ERG graph is evaluated in a sentiment analysis task.

4.3.1 Quality of Sentiment Lexicon for Entity Reputation

Crowdsourcing was used to ask online annotators to label each sentence according to the expressed sentiment towards the named entity as either very positive, positive, negative or very negative. Hence, to evaluate the quality of the ranked sentiment lexicon for entity reputation these tasks are described as follows: first, given a sentence annotators were asked to judge if a specified sentiment word is relevant to characterize the entity reputation. Second, given 5 sentiment words the annotator is again asked to judge if the entity described by those words has a positive, negative or neutral reputation. The first task (REL) evaluates if the captured sentiment words are relevant to measure the entity reputation while the second task (POL) evaluates the method ability to correctly weight sentiment words polarity.

For the REL task it is used 3,000 sentences where each sentence was randomly obtained from the subset C (Table 12). For the POL task it was used 3,000 combinations of sentiment words in which roughly one third were bigrams. For both experiments, it was created a gold standard by selecting the units where workers had an agreement of 75% or more, resulting in 2036 gold units for the first task and 943 gold units for the second task. The task POL-UNI and POL-BI refers to sentiment words obtained from unigrams and bigrams, respectively. The obtained results for the relevance task suggest that a very high percentage of the sentiment words captured by the ranked sentiment lexicon are relevant for entities reputation analysis. In parallel, results for the polarity task show that the associated weights for the sentiment words perform well on standard binary polarity evaluation.

Table 14. Crowdsourcing for Entities Reputation measured with RLDA.

Task	Precision	Recall	F-1
REL	84.5%	94.0%	89.0%
POL-UNI	80.2%	85.2%	82.6%
POL-BI	81.4%	82.0%	81.7%

To evaluate the reputation analysis algorithm it is generated a ground-truth dataset containing sentences from subset C (Table 12), where named entities were identified and labelled according to the sentiment polarity expressed towards them. First, 20 popular named entities are manually selected and approximately 4,000 sentences are crawled from subset C. The 4,000 sentences are split in two different sets: roughly 2,500 containing one of the selected entities and at least one sentiment word; and roughly 1,500 containing one of the selected entities and any other named entity (relations *f* and *h* on Section 4.1.1 and 4.1.3). After manually filtering the obtained results (to exclude noisy labels) we extract a ground-truth dataset composed of 729 sentences: 411 for the *f* graph relation and 318 for the *h* graph relation. Approximately 77.36% contain positive sentiment and 22.64% contain negative sentiment.

4.3.2 Sentiment Analysis for Entity Reputation

The number of domain entities pair citations is presented in Figure 31. In this figure allows to observe entities citations with other entities, for example it is possible to observe that the entities *batman* and *indiana* are related to 1,557 and 821 entities, respectively. Figure 32 presents the entities association to domain related sentiment words (e.g. *drama*, *trailer* and *oscar*). Moreover, Figure 33 presents the top positive and negative sentiment words. This illustrates how our method captures both general and domain specific sentiment words. Also, characters and actor names are frequently used as positive, or negative, reference (Figure 25 in Section 4.1). These observations motivate the intuition that sentiment words and domain entities tend to be used to influence entity reputation.

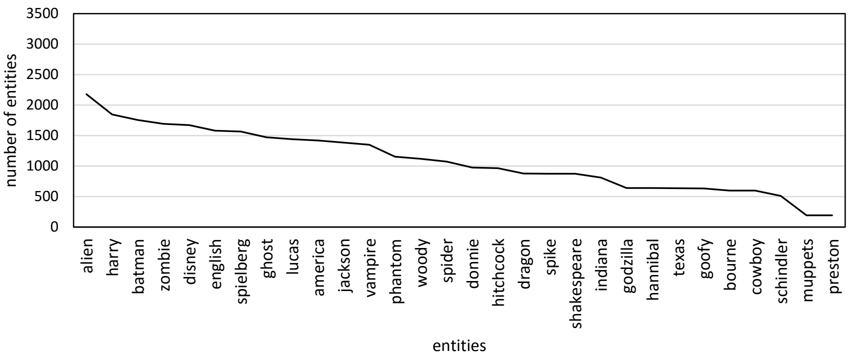


Figure 31: Entities citations.

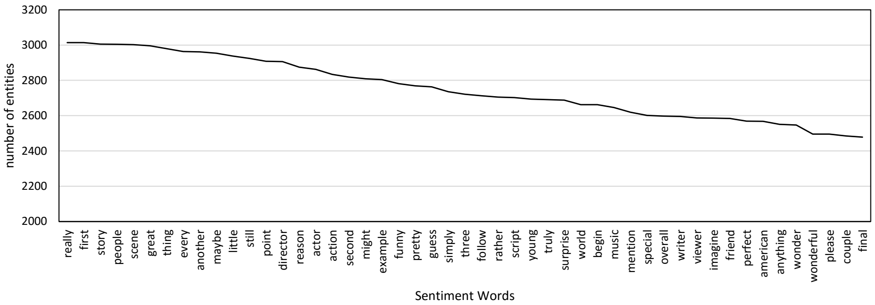


Figure 32: Sentiment words used as entities reputation qualifiers.

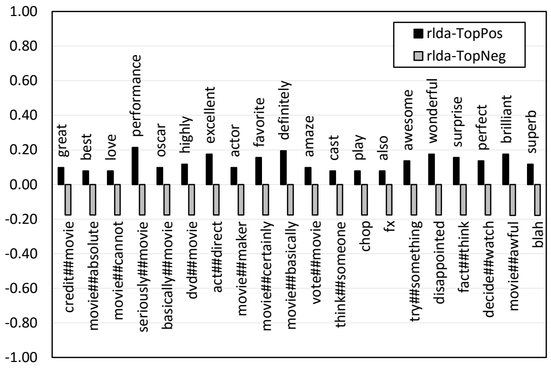


Figure 33: Top positive and negative RLDA sentiment words.

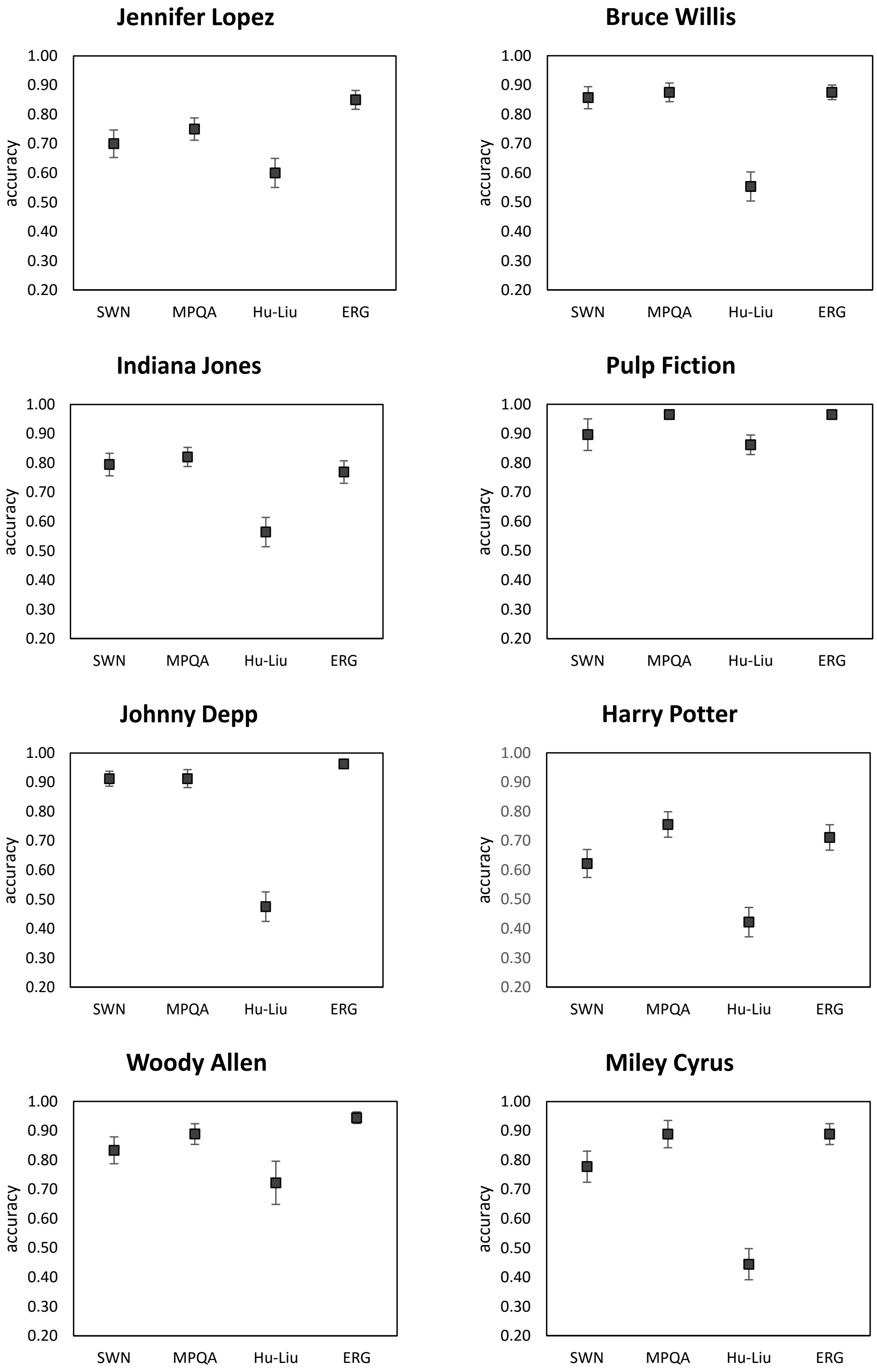


Figure 34. Accuracy and standard deviation of entities reputation analysis results. The results are shown for ERG reputation graph and the baselines SWN, MPQA and Hu-Liu.

Entities reputation graph (ERG) is built using the subset C (Table 12). The reputation graph encloses 12,687 vertices of which 3,177 are entities and 9,510 are sentiment words. ERG entities reputation is evaluated in a sentiment classification task where sentences that contain entities are evaluated. For the KNN classifier a balanced number of sentences were randomly selected from the ground-truth dataset (see previous subsection). Hence, containing a total of 200 sentences, the training split has a balanced number of positive/negative viewpoints targeting each entity reputation. The remaining ground-truth sentences (529) are used for test purposes.

Figure 34 presents the performance results using the K-Nearest Neighbour (KNN) classifier. In these experiments, KNN used the Manhattan distance to measure the nearest neighbour proximity. In Figure 34 is observed the accuracy obtained for each entity, and the standard deviation for the lexicons SWN (SentiWordNet), MPQA, Hu-Liu and ERG. In comparison to the other lexicons performance ERG shows a tendency to deviate towards a higher accuracy. Also, observing the standard deviation for the entities *Jennifer Lopez*, *Woody Allen* and *Pulp Fiction*, we observe a tendency to outperform the mean results obtained with the generic sentiment lexicons SWN, MPQA and Hu-Liu.

Table 15 results demonstrate that ERG reputation graph presents a good performance as outperforms MPQA, SWN and Hu-Liu sentiment lexicons. The MPQA lexicon is the closest competitor. For this reason, the following important aspects regarding the MPQA lexicon should be kept in mind: first, MPQA provides a list of words that were annotated as positive, negative or neutral and, as a consequence, the words have no polarity intensity – *excellent* and *good* are labelled as positive; second, this lexicon is context independent and is limited to approximately 6,886 words. With no context information sentiment words that influence entity reputation might present the incorrect sentiment polarity. For instance MPQA labels *charisma* as a negative word, however in the movie domain this sentiment word has a higher probability to have a positive influence when associated to domain entities. Other than incorrect sentiment weights, context dependent sentiment words may be unknown to generic sentiment lexicons (e.g. *oscar*). The ERG graph had no sentences without weighted sentiment words and only one sentence was represented by a single sentiment word. However, MPQA lexicon was unable to weight a minimum of one word in 39 test sentences, and 150 test sentences were represented solely one sentiment word.

Table 15. Reputation analysis results for the top 12 most cited entities (accuracy).

Entity	SWN	MPQA	Hu-Liu	ERG
Bruce Willis	76.79%	82.14%	58.93%	87.50%
Colin Firth	81.58%	73.68%	52.63%	84.21%
Fight Club	68.97%	93.10%	58.62%	82.76%
Johnny Depp	86.25%	82.50%	50.00%	96.25%
Miley Cyrus	66.67%	77.78%	44.44%	88.89%
Peter Jackson	68.29%	63.41%	58.54%	87.80%
Phantom Menace	75.86%	96.55%	82.76%	96.55%
Pulp Fiction	75.86%	96.55%	82.76%	96.55%
Shia Labeouf	78.57%	71.43%	42.86%	78.57%
Stanley Kubrick	77.78%	83.33%	50.00%	94.44%
Star Trek	83.33%	61.11%	55.56%	61.11%
Woody Allen	72.22%	83.33%	61.11%	94.44%
Total average	76.01%	80.41%	58.18%	87.42%

4.4 Sentiment Graph Visualization

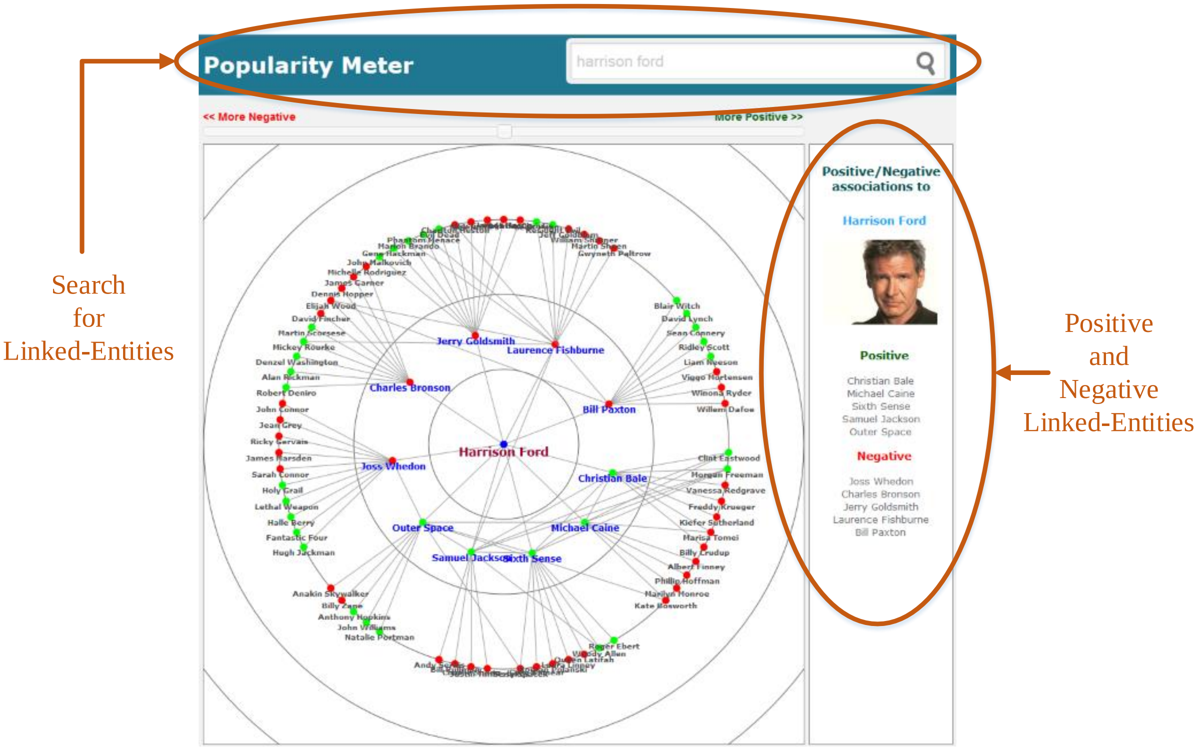
Online reviews consist of plain-text opinions where people share their views about multiple products, services, celebrities and others. This content is quite valuable in terms of reputation and feedback, however the way entities are connected to a positive or negative opinion and how this influences its' reputation is an ongoing research question. To solve this problem one needs to identify the associations of each entity to each sentiment word, thus, reputation analysis is naturally related to sentiment analysis. To address this problem, we introduced PopMeter[21] – a sentiment-graph visualization tool designed to inspect and explore the sentiment of linked-entities.

PopMeter is a sentiment-graph visualization tool that incorporates both entities and sentiment words information in a single heterogeneous graph, where nodes can correspond to entities or sentiment words. The sentiment graph aims at incorporating semantically related entities and entities sentiment weight. The sentiment weight is obtained from a sentiment lexicon that is created from user sentences without human supervision in a generative model that ties words to different sentiment levels (Section 3). In Figure 35 we present the PopMeter web interface, it

[21] Available at: http://popmeter.novasearch.org/

shows the sentiment graph with the actor "Harrison Ford" as its central node. PopMeter enables the user to explore the sentiment graph from a specific central node. The user can get an overview of the sentiment connections, limit the number of negative and positive connections, navigate in the sentiment graph edges, select other central nodes, and search for different entities or sentiment words.

The usage of PopMeter enables the user to observe how entities and sentiment words influence positively or negatively the reputation of other entities. With PopMeter the user can observe how the same entity can have an opposite reputation influence. As illustrated in Figure 36, the character "Hanna Montana" reputation is positively influenced by the "Walt Disney" industry, however, the "Walt Disney" industry is negatively influenced by the character "Hanna Montana".



Search for Linked-Entities

Positive/Negative associations to

Harrison Ford

Positive

Christian Bale
Michael Caine
Sixth Sense
Samuel Jackson
Outer Space

Negative

Joss Whedon
Charles Bronson
Jerry Goldsmith
Laurence Fishburne
Bill Paxton

Positive and Negative Linked-Entities

Figure 35. The PopMeter visualization of linked-entities in a sentiment graph.

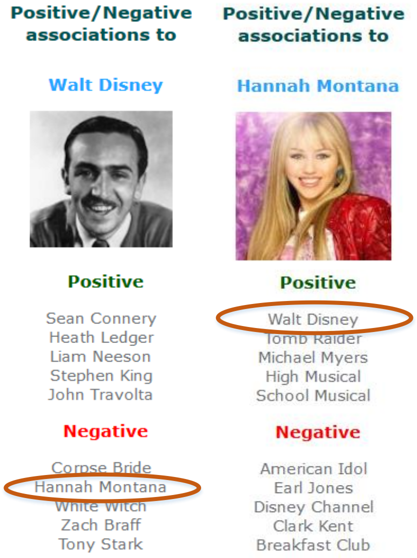


Figure 36. Opposite reputation influence.

4.5 Discussion

PopMeter[22] sentiment-graph is populated by entities and sentiment words. PopMeter presents a visualization of each entity and the respective sentiment connections – entities and/or sentiment words – sorted by the lowest and highest reputation levels. These connections correspond to the sentiment level that link the named entities between each other or with a sentiment word. For example, the sentiment word "happy" links positively and negatively with the named entities "Morgan Freeman" and "David Duchovny" respectively. The named entity "David Duchovny" links positively to the named entity "Sandra Bullock". As seen in Figure 28 (Section 4.1) the link between the named entities i and j are represented by $h(e_i, e_j)$ and the link between the sentiment word k and the named entity i by $f(sw_k, e_i)$. Moreover, to evaluate these links it was generated a ground-truth dataset where entities were identified and labelled according to their sentiment link. Results in Section 4.3 showed that there is a higher tendency to have a positive link (77.6% versus 22.64%). This reflects that there is a higher tendency to mention a popular entity as an esteemed reference. Yet to perform an unbiased evaluation it was chosen in Section 4.3.2 a balanced number of positive/negative viewpoints linking with each entity.

[22] Available at http://popmeter.novasearch.org.

It is important to recall from ERG discussion that generic sentiment lexicons make a higher effort to have a sentiment weight representation for a given sentence. For example, the MPQA lexicon was unable to detect a sentiment weight in 32% of the test sentences. Generic lexicons have a limited number of sentiment words and without context information the reputation of a given entity may be incorrectly identified. PopMeter sentiment graph contains entities that are referred by its regular name and also by its' alias or slang. These references are highly domain specific and with a generic sentiment lexicon would not be possible to observe these entities. We find this aspect important to mention as the usage of alias or slang is quite common in the movie domain. For example, "lort" is a reference to the movie *Lord of the Rings* and "spidey" is a nickname for *Spider Man* movies. Furthermore, with PopMeter the user can visualize these entities connections.

In the ERG reputation graph each entity has a high volume of sentiment links. To have a visual presentation it is vital to choose a subset of these sentiment links, hence, we selected the top 20 sentiment links. The top 20 sentiment links are obtained by the respective reputation values: the 10 named entities or sentiment words with the higher and lower reputation values. Reputation values are highly co-related to the named-entity popularity level, as a consequence popular entities tend to have a very high or low reputation values. Additionally, popular named entities link to a high number of named entities, and since their reputation values are high they tend to populate the top reputation influencers for many different entities. Moreover, to bring a more diverse number of named entities reputation influencers we must go beyond the top 20.

4.6 Summary

A sentiment lexicon includes sentiment words that characterize the general sentiment towards a named entity, however target named entities are themselves part of the sentiment lexicon. To this end, we investigated a reputation method that aims to compute an entity reputation based on the analysis of the sentiment expressed about that entity. In doing so, it was observed that popular entities become a reference in the domain and are, commonly, vastly cited as an example of highly reputable entities.

To evaluate the reputation of the target named entities we proposed a method that extracted a domain sentiment lexicon (chapter 1), and then computed a reputation graph that analyses cross-citations in subjective sentences. Each entity reputation was updated through an iterative optimization method that exploited the graph of the linked-entities. We presented two evaluations: one to evaluate the quality of the ranked sentiment lexicon for entity reputation, and other to evaluate the reputation analysis algorithm. Our results showed that a high percentage of the sentiment words captured by the ranked sentiment lexicon were relevant for entities reputation analysis, and that sentiment words associated to sentiment weights perform well on standard binary polarity evaluation. In the performed experiences our method outperformed three sentiment lexicons baselines. Therefore, in this chapter we have successfully shown that entities reputation can be measured through context dependent sentiment lexicons in which entities are used as part of the sentiment lexicon.

The work presented in this chapter was published at:

Peleja, F., Santos, J. and Magalhães, J. 2014. "Reputation Analysis with a Ranked Sentiment-Lexicon." In *Proceedings of the 37th International ACM SIGIR Conference on Research Development in Information Retrieval*, 1207–10. SIGIR '14. Gold Coast, Australia: ACM. doi:10.1145/2600428.2609546.

Peleja, F., Santos, J. and Magalhães, J. 2014. "Ranking Linked-Entities in a Sentiment Graph." In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 118–25. Warsaw, Poland: IEEE Computer Society. doi:10.1109/WI-IAT.2014.88.

5

Sentiment-based Recommendation

The popularity of the information exchanged by media consumers has been increasing at an enormous rate. While some Web applications allow users to rate or comment a movie, others only allow one of the possibilities. For example, blogs and online forums only support comments and personal media players only support ratings. Some authors such as Takama and Muto (2007) have explored sentiment analysis techniques to build profiles of TV viewers based on the analysis of viewers' comments. In contrast, we bypass the analysis of user profiles and directly compute new recommendations. In this chapter, we investigate how to apply sentiment analysis techniques in recommendation systems.

The principal objective of a recommender systems (RS) is to identify products that users may be interested but are not aware of. In general, a RS suggests unknown items (e.g. movies) by considering information exchanged by users when interacting with the system. Online merchants started to incorporate some efforts in RS in the early 90's (Resnick et al., 1994). As a consequence recommendations have become highly popular in e-commerce services such as Amazon[23] and Netflix[24] (Koren et al., 2009). Before the evolution of RS algorithms users would more likely ask for a recommendation from their own circle of known friends or family than online users. Nonetheless, recommendations demand a certain level of trustworthy knowledge and not everyone is eligible to provide a skilled recommendation. Hence, a RS should be related to a trustworthy service (i.e. associated to a popular e-commerce service), and observe the interactions

[23] http://www.amazon.com

[24] http://www.netflix.com

of a large amount of users to provide a more reliable and insightful recommendation which an average person could not provide.

In general, two families of algorithms inspire recommendation systems: content-based filtering which analyses the correlation between users' personal information and items metadata, and collaborative filtering (CF) which analyses the patterns of user-item ratings that are modelled over time to predict items that might be of interest to particular users. The main difference between these two strategies relies on the nature of the information used to build the RS. Content-based approaches are more likely to use information related to users and items which many times is obtained manually, as a consequence becomes a very expensive approach. Also, the recommendations are limited to like-minded users. In contrast, collaborative-filtering approaches automatically identify future preferences by observing users' interaction, i.e., user-item explicit ratings and users' reviews.

The central hypothesis of this chapter is that in the context of RS, people interactions are not limited to user profiles or explicit ratings. In social media platforms such as online forums users leave valuable feedback about products, organizations (entities) and products' aspects that are later appreciated by other users. In this chapter we propose to improve product recommendations by exploring the output of sentiment analysis in two different approaches: (1) capturing user textual opinions as sentiment-ratings and (2) exploring the reputation of entities to address cold-start recommendations.

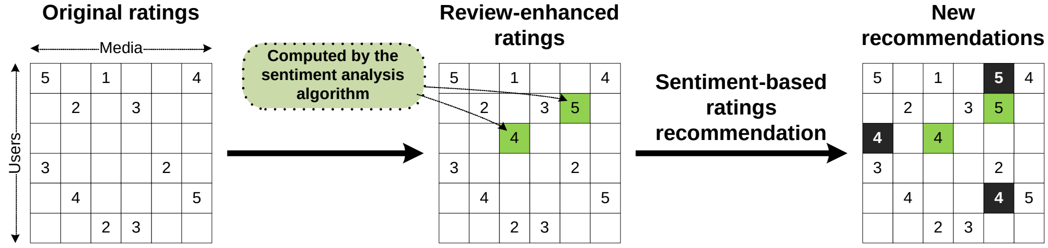

Original ratings
Media
Users
Computed by the sentiment analysis algorithm
Review-enhanced ratings
Sentiment-based ratings recommendation
New recommendations

Figure 37. Overview of the sentiment-based ratings framework.

In the first approach we propose to use sentiment-based ratings in a collaborative recommendation system. The standard collaborative filtering approach assumes the existence of a ratings matrix containing all users-products ratings. Figure 37 illustrates the process described in this chapter. The ratings matrix is by nature highly incomplete: there is a large number of products and each user only rates a limited number of products. In the movie domain (i.e. IMDb) a user is likely to rate an average of 30 movies from a set of 2 million movies where the remaining user ratings are unknown. Therefore, the ratings matrix can be made more complete by adding ratings inferred by a sentiment analysis approach of user reviews.

In the second approach, we turn our attention to automatic systems that mine trends and reputations across multiple social media services (chapter 4), which we believe can be of great value in many recommendation scenarios. In a general recommendation scenario user-item ratings are predicted for old items, i.e., movies that have already been rated by users. A different, and more challenging, scenario for recommender systems is the cold-start scenario: given a new item that has not been rated or commented by any user, how can we relate this item to other items or potential consumers? To this end, we tackle the cold-start problem with an analysis of social-media services in a content-based recommendation system approach.

This chapter concludes with the description of the SentiMovie[25] demonstrator which was developed to allow the user to visualize the influence of sentiment analysis techniques on a recommendation system framework. This integrated approach grants that no information is lost with extra processing steps such as creating user profiles. SentiMovie presents a visualization of the algorithm ability to provide movie recommendations, where the algorithm observes both explicit ratings and inferred ratings obtained from a sentiment analysis classification of free-text comments with no rating associated.

The contributions of this chapter are as follows:

- Using inferred ratings obtained from user- reviews we are able to improve a recommendation algorithm. To this aim, we provide experiments that corroborate our intuition: sentiment information should not be disregarded in recommendation systems.
- A collaborative filtering recommendation algorithm is improved using probabilistic sentiment ratings. Experiments show that probabilistic sentiment ratings that include a broader scale are able to provide a model more accurate than positive vs. negative.
- A content based recommendation algorithm is improved by using actors and directors' reputation (chapter 4) to better characterize upcoming (new) movies.
- SentiMovie interface that allows the user to have an insight of the advantages of using sentiment analysis algorithms in a recommendation system framework.

[25] http://popmeter.novasearch.org/sentimovie-web/

5.1 Collaborative-Sentiment based Recommendation

To develop a novel recommendation system the proposed framework uses the information obtained from a sentiment analysis model, and the explicit ratings given to the products by each user. The algorithm behind the recommendation framework analyses user comments and represents these together with user explicit ratings in a collaborative matrix integrating the interactions of all users. The framework is divided in two parts: an algorithm that studies users' comments and computes ratings from this analysis, and a collaborative filtering recommendation algorithm that merges all data in a single matrix.

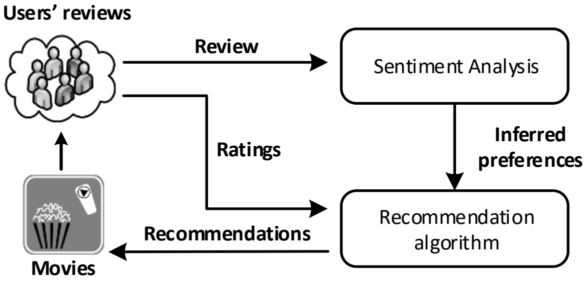


Figure 38. Recommendations based on ratings and reviews.

Figure 38 depicts this approach: as in a typical social-media scenario users comment and rate movies to express their preferences. The ratings are received by the recommender system and user ratings are inferred from the text comments. Finally, both explicit and inferred ratings are merged in the recommendation algorithm. Since the inferred ratings are the result of a text sentiment analysis algorithm and are not explicitly provided by users, this approach is denoted as a weakly-supervised recommendation algorithm.

5.1.1 Collaborative-Filtering Matrix Factorization

Recommendation algorithms have proven their ability to influence users' future purchases by observing the available user explicit ratings, product characteristics and/or users' past behaviour. However, the amount of products rated by users is a small percentage of the total number of available products: the user-product matrix used by recommendation algorithms is sparsely filled with ratings explicitly given by the users. Among existing RS techniques, collaborative filtering (CF) techniques are widely used, where latent factor models are quite popular (Koren et al., 2009). These statistical models establish a relationship between a set of variables and a set of latent variables, such tools are very useful for high-dimensional data. A well-known alternative to latent factor models are the neighbourhood methods. However, neighbourhood methods make the assumption that like-minded users should share their neighbourhood, as a consequence do not offer diverse recommendations. To discover a wider range of recommendations we compute a RS with latent factor models. The intuition behind latent factor models is to be able to map both users and products onto the same latent factor space. Latent factor models represent users and products as vectors with k dimensions:

$$p_u = (u_1, u_2, \dots, u_k) \quad q_i = (i_1, i_2, \dots, i_k), \tag{36}$$

where p_u is the user u factors vector, q_i is the product i factors vector, and k is the number of latent factors (dimensions) where each user u and movie i are represented. With this latent factor representation of users and products we intend to achieve a rating prediction rule to assess user preferences for each product. For an unknown rating[26] $r_{ui} = \emptyset$ we wish to predict its value by calculating the dot product of their respective latent factors, as follows:

$$\hat{r}_{ui} = p_u \cdot {q_i}^T, \tag{37}$$

where $\hat{r}_{ui}$ is the predicted rating of user u for product i. Also, in this scenario we wish to consider a set of data $R = \{R_{ra}, R_{rev}\}$ composed of the R_{ra} ratings matrix provided by users and the R_{rev} set of text reviews written by users. More specifically, the ratings matrix assumes the form,

$$R_{ra} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,m} \\ \vdots & \ddots & \vdots \\ r_{n,1} & \cdots & r_{n,m} \end{bmatrix}, \quad r_{ij} = \{\emptyset, 1,2, \dots, 10\} \tag{38}$$

where each element r_{ij} corresponds to rating assigned by user i to product j. This matrix is highly incomplete since most elements are empty, i.e., $r_{ij} = \emptyset$. The reviews set

$$R_{rev} = \{(re_1, r_1, p_1, u_1), \dots, (re_k, r_k, p_k, u_k)\} \tag{39}$$

contains k elements, where each element is represented by a text review re_i and the corresponding rating r_i. Variables p_i and u_j are indicator variables holding the product and user index respectively.

For reviews re_i, the rating r_i is unknown (or withheld in the training phase). Thus, text reviews do not need to be accompanied by ratings since our proposal is to infer probabilistic ratings from reviews. A set of reviews with unknown ratings is described as follows,

$$\hat{R}_{rev} = \{(re_1, \hat{c}_1, \theta_1, p_1, u_1), \dots, (re_k, \hat{c}_k, \theta_k, p_k, u_k)\} \tag{40}$$

[26] Rating from a user u to a product i that we have no information about.

where for each text review re_i we wish to infer a probabilistic rating $\hat{c}_i = \{1, \dots, 10\}$ and the corresponding probability θ_i (this can be seen as a confidence level).

Finally, the proposed recommender system considers both explicit ratings R_{ra} and review's inferred ratings $\hat{R}_{rev}$ to better predict future recommendations,

$$\hat{R}_{ra} = \begin{bmatrix} \hat{r}_{1,1} & \cdots & \hat{r}_{1,m} \\ \vdots & \ddots & \vdots \\ \hat{r}_{n,1} & \cdots & \hat{r}_{n,m} \end{bmatrix}, \quad \hat{r}_{ij} = \{1,2, \dots, 10\}. \tag{41}$$

The combination of ratings and reviews is not as straightforward as one might initially suppose. Reviews are quite biased and writing skills differ greatly according to the users. To this end, we propose a sentiment analysis algorithm to explore such sheer volume of valuable information. Also, to compute the ratings predictions we followed a singular value decomposition (SVD) approach. SVD provides a convenient way of breaking a matrix into a computationally simpler and meaningful problem. Following Koren et al. (2009) we compute a low-rank SVD decomposition of the R_{ra} matrix, $R_{ra} = U\Sigma V^T = P \cdot Q^T$:

$$R_{ra} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,n} \\ \vdots & \ddots & \vdots \\ r_{m,1} & \cdots & r_{m,n} \end{bmatrix} = \begin{bmatrix} u_{1,1} & \cdots & u_{1,k} \\ \vdots & \ddots & \vdots \\ u_{m,1} & \cdots & u_{m,k} \end{bmatrix} \cdot \begin{bmatrix} p_{1,1} & \cdots & p_{1,k} \\ \vdots & \ddots & \vdots \\ p_{n,1} & \cdots & p_{n,k} \end{bmatrix}^T = U\Sigma V^T = P \cdot Q^T. \tag{42}$$

where $P = U \cdot \sqrt{\Sigma}$, $Q = \sqrt{\Sigma} \cdot V$, the matrix U contains the left singular vectors, Σ contains the singular vales and V contains the right singular vectors of the original users-products matrix R_{ra}. Here we consider a k-rank approximation of the full matrix.

In the decomposed users-products matrix R each element r_{ij} corresponds to a rating assigned by user i to product j. Each vector (row) p_u of P represents a user u and each vector (row) q_i of Q represents a product i. Therefore SVD enables a low-rank approximation by zeroing out the less relevant (lower) singular values and preserves only the k most relevant ones, contained in the matrix Σ. However, SVD is originally designed to be used over a complete matrix and matrix R is a sparse matrix. Hence, SVD technique must undergo some modifications to deal with sparsely filled matrices. In that sense, Simon Funk[27] suggested an efficient solution to learn the factorization model which has been widely adopted by other researchers (Koren et al., 2009). The method consists in decomposing the ratings matrix into a product of a user-factor matrix with a product-factor matrix, by taking into account the set of known ratings only. Hence, matrices P and Q are given by:

[27] http://sifter.org/~simon/journal/20061211.html

$$[P, Q] = \underset{p_u, q_i}{argmin} \sum_{r_{ui} \in R_{ra}} (r_{ui} - p_u \cdot {q_i}^T)^2 + \lambda \cdot (\|p_u\|^2 + \|q_i\|^2). \tag{43}$$

This expression accomplishes two goals: matrix factorization by minimization and the corresponding regularization. The first part of the equation pursues the minimization of the difference (henceforth referred to as error) between the known ratings (r_{ui}) present on the original R_{ra} ratings matrix and their decomposed representation P and Q. The second part controls generality by avoiding overfitting during the learning process, where λ is a constant defining the extent of regularization, usually chosen by cross-validation.

Even though latent factors have the ability to capture rating tendencies, some improvements can be made to the model by defining baseline predictors. A straightforward choice for a baseline predictor is the global average of the observed ratings. Moreover, some users can be more demanding than others, similarly the ratings associated with the products will differ according to the user. Based on this premise, we should capture these trends: user-related and product-related deviations from the average rating. Therefore, prediction rule will be modified into:

$$\hat{r}_{ui} = p_u \cdot {q_i}^T + \mu + b_u + b_i, \tag{44}$$

where global rating average and biases are observed. The parameters μ, b_u and b_i represent the global rating average, user bias and product bias, respectively. To this end, user-product predictions are computed according to the expression:

$$[P, Q] = \underset{p_u, q_i}{argmin} \sum_{r_{ui} \in R} (r_{ui} - \hat{r}_{ui})^2 + \lambda \cdot \left(\|p_u\|^2 + \|q_i\|^2 + {b_u}^2 + {b_i}^2\right). \tag{45}$$

which considers users and products biases in the ratings matrix decomposition. For a matter of simplicity we use the $\hat{r}_{ui}$ instead of the full equation expression.

5.1.2 Matrix Factorization with Sentiment-based Regularization

In this section we introduce the proposed novel matrix factorization framework that aims to improve the recommendation algorithm by uncovering new ratings from user reviews. These sentiment-driven ratings are included along with the explicit ratings in the recommendation model. Other authors (Leung et al., 2006; Jakob et al., 2009) have explored the idea of a concept or keyword driven approach to interlink different ratings by taking textual data into the heart of the recommendation algorithm. In contrast, we propose to quantify the uncertainty of user opinions and include this uncertainty in a matrix factorization system.

The first step is to make a probabilistic prediction (rating $\hat{c}_{ui}$) from the analysis of a user review re_{ui}. To consider this additional information we add a new parcel to the equation of the user-products predictions,

$$\hat{R}_{ra} = \underset{p_u,q_i}{argmin} \sum_{r_{ui} \in R} (r_{ui} - \hat{r}_{ui})^2 + \sum_{\hat{c}_{ui} \in \hat{R}_{rev}} (\hat{c}_{ui} - \hat{r}_{ui})^2 + \lambda \cdot \left(\|p_u\|^2 + \|q_i\|^2 + {b_u}^2 + {b_i}^2 \right) \tag{46}$$

where $\hat{c}_{ui}$ is the rating obtained by a sentiment analysis algorithm. In this expression we materialize the uncertainty of a user opinion in a rating. However, this is the result of a decision supported by some probability. For example, the sentiment analysis decided for rating 2 with a probability of 0.65. Hence, we argue that a inferred rating $\hat{c}_{ui}$ should not be treated in the same way as an explicit rating r_{ui} because it is subject to algorithm's consistencies and stability.

To account for the sentiment analysis uncertainty, the factorization of the recommendation matrix is re-written as follows,

$$\begin{aligned} \hat{R}_{ra} = \underset{p_u,q_i}{argmin} \sum_{r_{ui} \in R} (r_{ui} - \hat{r}_{ui})^2 + \sum_{\hat{c}_{ui} \in \hat{R}_{rev}} \theta_{ui} \cdot (\hat{c}_{ui} - \hat{r}_{ui})^2 + \lambda \\ \cdot \left(\|p_u\|^2 + \|q_i\|^2 + {b_u}^2 + {b_i}^2 \right) \end{aligned} \tag{47}$$

where θ_{ui} corresponds to the confidence factor associated with a rating $\hat{c}_{ui}$. The inferred rating

$$\hat{c}_{ui} = \underset{r \in \{1,\ldots,10\}}{\operatorname{argmax}} p(ra_{ui} = r | re_{ui}) \tag{48}$$

corresponds to the rating value that maximizes the probability of rating $ra_{ui} = r \in \{1, \ldots, 10\}$ given the text review re_{ui}. The confidence level,

$$\theta_{ui} = p(ra_{ui} = \hat{c}_{ui} | re_{ui}) \tag{49}$$

is the probability of a rating given the text review re_{ui}. We consider the pair of values θ_{ui} and $\hat{c}_{ui}$ to be essential for the proposed framework. Together, they rise to a regularized decomposition of a full ratings matrix in which this matrix is composed of a user entered ratings and probabilistic ratings.

5.1.3 Probabilistic Sentiment-Ratings Inference

The task of the probabilistic sentiment-ratings classifier is to infer the sentiment-ratings $\hat{c} = \{\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_m\}$ of a given set of reviews $R = \{re_1, re_2, \ldots, re_n\}$. Every review re_i contains a set of opinion words $re_i = (ow_{i1}, ow_{i2}, \ldots, ow_{im})$ extracted according to the sentiment analysis techniques described in chapter 3 (Section 3.5 – Sentiment Classification). For a matter of simplicity in this discussion we shall ignore the user and product indices. The goal is to learn a classifier to infer the rating of a given review re_i. Following a machine learning approach, this classifier is learnt as a probabilistic model $p(ra_i|re_j)$ estimated from a training set $\Theta = \{(re_1, ra_1), (re_2, ra_2), \dots, (re_j, ra_j)\}$.

Traditional binary classifiers for sentiment analysis such as Pang et al. (2002) do not provide enough broadness to cover all ratings. Moreover, a linear regression scattered across all ratings scale would not provide the required confidence level. To this end, we propose a solution that reaches beyond the simple "thumbs up vs. thumbs down" approach to reviews ratings.

The need for a finer-grain sentiment analysis approach pushed us towards a multiple-Bernoulli classifier (chapter 3 – Section 3.5) implemented in a one-versus-all setting. For each rating value (e.g. 10 if the rating range is 1 to 10) a Bernoulli classifier is learned. This renders the model,

$$p(ra_i = r|re_i) = \frac{f^r(ra_i, re_i)}{\sum_{L=1}^{10} f^L(ra_i, re_i)} \tag{50}$$

where each function $f^r(ra_i, re_i)$ corresponds to the r^{th} rating. In the multiple-Bernoulli classifier model the rating prediction is normalized according to the predictions of all ratings. For each review i the rating that maximizes the expression,

$$\underset{r}{\operatorname{argmax}}\, p(ra_i = r|re_i) = \frac{f^r(ra_i, re_i)}{\sum_{L=1}^{10} f^L(ra_i, re_i)} \tag{51}$$

is assigned to the review. The prediction probability will correspond to the confidence factor θ_i of the factorization recommendation matrix prediction. Functions $f^r(ra_i, re_i)$ are learned with an online gradient descent and a squared error loss (Vowpal Wabbit).

5.2 Linked-Entities Reputation and the Cold-Start Problem

In recommender systems the cold-start problem is a well-known problem. When a new item has no ratings, it becomes difficult to relate it to other items or users. In this section, we address the cold-start problem and propose to leverage on social-media trends and reputations to improve the recommendation of new items. The proposed framework models the long-term reputation (chapter 4) of actors and directors to better characterize new movies. Also, the proposed framework aims to model the long-term reputation of actors and directors to better characterize new movies (cold-start problem).

To handle the cold-start problem, we explore the reputation of new movies, directors and actors in social-media services, namely on Twitter and IMDb. Y Moshfeghi et al. (2011) showed that recommendation of movies that are unknown to the system, performs best when considering both the movie metadata and the sentiment expressed in movie reviews. Building on this idea, we represent a movie as the vector

$$m_j = \left(D_j, A_j, G_j, R_j, S_j\right), \tag{52}$$

where D_j is the set of directors, A_j is the set of participating actors, G_j is the set of corresponding genres, R_j is the set of associated user ratings and S_j is the social-media feedback inferred by the monitoring process described previously. The S_j variable is composed of the Twitter posts (or tweets) about the movie m_j as well as the reputation of its directors and actors, obtained from IMDb. As we will see, S_j will be fundamental for improving cases of cold-start recommendations where $R_j = \emptyset$.

In this scenario, users rate the movies they have watched and from this data we compute their profiles in terms of preferences towards directors, actors and genres. Formally, a user u_i is then represented as the vector

$$u_i = \left(D^i, A^i, G^i\right), \tag{53}$$

where D^i is the set of directors, A^i is the set of actors and G^i is the set of genres. These three sets follow the same structure and are represented as

$$D^i = \left\{\left({d_i}^1, {dr_i}^1, {df_i}^1\right), \dots, ({d_i}^n, {dr_i}^n, {df_i}^n), \dots\right\}, \tag{54}$$

$$A^i = \left\{\left({a_i}^1, {ar_i}^1, {af_i}^1\right), \dots, ({a_i}^n, {ar_i}^n, {af_i}^n), \dots\right\}, \tag{55}$$

$$G^i = \left\{\left({g_i}^1, {gr_i}^1, {gf_i}^1\right), \dots, ({g_i}^n, {gr_i}^n, {gf_i}^n), \dots\right\}, \tag{56}$$

where the first element ${d_i}^n$ identifies the director, ${dr_i}^n$ is the average rating given by the user to the movies directed by that director and ${df_i}^n$ is the number of movies directed by ${d_i}^n$ that are rated by the user. The same rationale applies to A^i and G^i.

Recommendations are computed by predicting user-movie ratings for the target user and the candidate new movies. A new movie is recommended if the predicted rating is above the user-specific threshold T_i, formally obtained by the expression:

$$T_i = \frac{\sum_{r_i^k \in ur_i} r_i^k}{|ur_i|}, \tag{57}$$

where $ur_i = \left\{r_i^1, \dots, r_i^k, \dots, r_i^K\right\}$ are the user u_i past ratings and $|ur_i|$ is the total number of past ratings given by u_i.

5.2.1 Formal Model

We start by exploring the similarity of the movie profile and user profile. This similarity is obtained by quantifying how much a user likes each aspect of the movie separately, i.e., the values $\hat{d}_{ij}$, $\hat{a}_{ij}$ and $\hat{g}_{ij}$, and later combining them into a final score.

To infer the user u_i preference towards the directors of the movie m_j, we compute the weighted average of how much the user likes each director of the movie, i.e., the weighted average of the values dr_i for each director on D_j. The weight that represents the contribution of each director rating to the average is calculated according to the number of movies that the user rated where the director participated, i.e., each director's corresponding value df_i on the user profile D^i. The reasoning is that a user formulates a more refined and accurate opinion about a director if he/she watches more movies from that director. Hence, we consider that directors that have been watched more times by the user should have a stronger weight on the prediction. Let $D_{ij} = D^i \cap D_j$ be the set of the directors of movie m_j that are on the user profile D^i. The weight $w_{d_{ij}^n}$ of the n^{th} director $d_{ij} \in D_{ij}$ is then obtained by the expression

$$w_{d_{ij}^n} = \frac{df_i}{\sum_{p \in D_{ij}} df_p}, \tag{58}$$

such that $\sum_n w_{d_{ij}^n} = 1$. Considering this, the preference of user u_i towards the team of directors of the movie m_j is obtained by the expression

$$\hat{d}_{ij} = \frac{\sum_{n \in D_{ij}} dr_i^n \cdot w_{d_{ij}^n}}{|D_{ij}|}, \tag{59}$$

where $|D_{ij}|$ is the number of directors on D_{ij}. Since all director ratings dr_{ij} are values between 1 and 10, the resulting average $\hat{d}_{ij}$ will also be a value between 1 and 10. Note that when none of the directors of movie m_j are on the user's directors set D^i, $\hat{d}_{ij} = 0$.

The likeliness of user u_i appreciating the actors of a given movie m_j can be obtained in similar method as shown in the aforementioned equation for the team of directors. Let $A_{ij} = A^i \cap A_j$ be the set of actors of movie m_j that are on the user profile A^i. Thus, the user u_i preference towards likes the actors of the movie m_j is obtained by the expression

$$\hat{a}_{ij} = \frac{\sum_{n \in A_{ij}} ar_i^n \cdot w_{a_{ij}^n}}{|A_{ij}|}, \tag{60}$$

where $|A_{ij}|$ is the number of actors on A_{ij} and $w_{a_{ij}^n}$ is the weight of the actor a_{ij}^n. Similarly to $\hat{d}_{ij}$, when none of the actors of movie m_j are on the user actors A^i, $\hat{a}_{ij} = 0$. In turn, let $G_{ij} = G^i \cap G_j$ be the set of the genres of movie m_j that are on the user profile G_i. How much the user u_i likes the genres of the movie m_j is obtained by the expression

$$\hat{g}_{ij} = \frac{\sum_{n \in G_{ij}} gr_i^n \cdot w_{g_{ij}^n}}{|G_{ij}|}, \tag{61}$$

where $|G_{ij}|$ is the number of genres on G_{ij} and $w_{g_{ij}^n}$ is the weight of the genre g_{ij}^n. Like $\hat{d}_{ij}$, and $\hat{a}_{ij}$, $0 \leq \hat{g}_{ij} \leq 10$, with 0 occurring when none of movie genres are on the user genres G^i.

The predicted rating $\widehat{pr}_{ij}$ for user u_i and the cold-start movie m_j is obtained by the expression:

$$\widehat{pr}_{ij} = \frac{1}{T}\left(\theta_a \cdot \hat{a}_{ij} + \theta_d \cdot \hat{d}_{ij} + \theta_g \cdot \hat{g}_{ij}\right). \tag{62}$$

where θ_a, θ_d and θ_g are constants controlling the contributions of directors, actors and genres to the rating predictions. Their values are estimated from a set of training data by finding the values that minimize Mean Average Error. Additionally, let T be the number of feature set ratings $\hat{d}_{ij}$, $\hat{a}_{ij}$ and $\hat{g}_{ij}$ that are different from 0.

5.2.2 Social-Media Trends and Reputations

In this section we argue that a recommended movie should not only match the user preferences, but also be a high quality movie: note that $\widehat{pr}_{ij}$ lacks this second component. Hence, we will extend the $\widehat{pr}_{ij}$ computation to include social-media feedback. Next we will formalize how social-media trends and entities reputation (chapter 4) is incorporated in this recommendation algorithm framework. The social-media feedback is given by,

$$S_j = \{T(m_j), reps(m_j)\}, \tag{63}$$

as the set of tweets $T(m_j)$ where the movie m_j is mentioned, and the reputation of all actors and directors participating on movie m_j.

New-movies popularity on Twitter

The social-media feedback about new movies is obtained from Twitter: tweets where the movie title is mentioned are stored and labelled according to the movies' titles. The captured tweets are then classified by a sentiment classifier such that, for each tweet, it's inferred if it's a positive or negative reference to the movie. A tweets index is then constructed to allow fast look-ups for the cold-start recommendation. Formally, the resulting tweets for a certain movie m_j are represented as the set

$$T(m_j) = \{(t_{j1}, s_{j1}), \dots, (t_{jl}, s_{jl}), \dots, (t_{jM}, s_{jM})\}, \tag{64}$$

where t_{jl} is the tweet (talking about m_j) and s_{jl} is the sentiment of the tweet such that $s_{jl} \in \{pos, neg\}$. We used a KNN classifier and a domain-specific sentiment lexicon for extracting the tweets features.

Actors and directors reputation on IMDb

The social-media feedback on directors and actors is obtained from IMDb: movie reviews are crawled and used to build a sentiment graph linking the named entities, from which the reputations of directors and actors are computed. The sentiment graph building process corresponds to the method presented in chapter 4. Ultimately, this process allows us to obtain the reputation of the directors and actors of the new movies we want to recommend. Hence, the crawled reviews correspond to the old movies where those directors and actors have participated. Formally, the reputation of all the directors and actors participating on a movie m_j is represented by the expression

$$\text{reps}(m_j) = \{\text{rep}(e_1), \dots, \text{rep}(e_k), \dots\}, \tag{65}$$

where the reputation of each entity e_k is $\text{rep}(e_k) \in [0.0, 1.0]$, with 0.0 being the worst reputation and 1.0 being the best reputation.

5.2.3 Recommendations with Social-Media Signals

Yashar Moshfeghi et al. (2011) and Krauss et al (2008) used hidden latent factors to correlate movies through sentiment analysis techniques. Here, however, new movies do not have reviews and tweets about new movies, hence this information is too scarce to infer relevant latent topics. Therefore, to explore emotion as a qualitative measure we obtain and consider the inherent quality of new movies, directors and actors. The rating prediction $\hat{r}_{ij}$ is obtained by considering both how popular the movie is $\text{pop}(m_j)$, and how much a user might enjoy the movie m_j, given the reputations $\text{reps}(m_j)$ of its participants. The proposed approach is formalized as

$$\hat{r}_{ij} = \alpha_t \cdot (\text{pop}(m_j) + \ bias_i) + (1 - \alpha_t) \cdot \widehat{pr}_{ij|\text{reps}(m_j)}, \tag{66}$$

where α_t is a constant reflecting the importance of the movies popularity to the final user-movie rating. Note that $\text{pop}(m_j)$ represents the general opinion towards the movie m_j. The user bias

$bias_i$ is used to adjust this value to the user personal standards. Formally, the user u_i bias accounts for the deviation of the user past ratings from the general average rating of the corresponding movies:

$$bias_i = \frac{\sum_{r_i^k \in ur_i}(r_i^k - avg_{<k>})}{|ur_i|}. \quad (67)$$

Let $ur_i = \{r_i^1, \ldots, r_i^k, \ldots, r_i^K\}$ be the user u_i past ratings, $avg_{<k>}$ be the average rating of the movie $m_{<k>}$, and $|ur_i|$ is the number of past ratings given by user u_i. Rewriting the predicted rating with $\widehat{pr}_{ij}$ the new reputations information we have:

$$\widehat{pr}_{ij|\mathrm{reps}(m_j)} = \frac{1}{T}\left(\theta_a \cdot \hat{a}_{ij|\mathrm{reps}(m_j)} + \theta_d \cdot \hat{d}_{ij|\mathrm{reps}(m_j)} + \theta_g \cdot \hat{g}_{ij}\right). \quad (68)$$

Modelling user preferences $\hat{a}_{ij|\mathrm{reps}(m_j)}$ and $\hat{d}_{ij|\mathrm{reps}(m_j)}$ with social-media signals

Up until this point, when predicting the values $\hat{d}_{ij}$ and $\hat{a}_{ij}$ (i.e., how much a user likes or dislikes the directors and actors of a movie) the entities that the user does not know were not considered. Hence, we propose to enhance the calculation of $\hat{d}_{ij}$ and $\hat{a}_{ij}$ by observing the reputations of directors and actors available in $\mathrm{reps}(m_j)$. To this end, two new variables, $\widehat{ud}_{ij}$ and $\widehat{ua}_{ij}$, are introduced to express the reputation of the unknown directors and actors:

$$\widehat{ud}_{ij} = \frac{\sum_{d \in D_j - \mathrm{D}^{\mathrm{i}}} \mathrm{rep}(d)}{|D_j - \mathrm{D}^{\mathrm{i}}|}, \qquad \widehat{ua}_{ij} = \frac{\sum_{a \in A_j - \mathrm{A}^{\mathrm{i}}} \mathrm{rep}(a)}{|A_j - \mathrm{A}^{\mathrm{i}}|}, \quad (69)$$

where $D_j - D^i$ and $A_j - A^i$ are the sets of directors and actors on movie m_j that the user does not know.

To consider $\widehat{ud}_{ij}$ and $\widehat{ua}_{ij}$, in the calculation of $\widehat{pr}_{ij|\mathrm{reps}(m_j)}$, one ought to note that $\hat{d}_{ij}$ and $\hat{a}_{ij}$ represent user preferences towards their known directors and actors. Thus, $\hat{d}_{ij|\mathrm{reps}(m_j)} = \hat{d}_{ij}$ and $\hat{a}_{ij|\mathrm{reps}(m_j)} = \hat{a}_{ij}$, when all the directors or actors of m_j are known by the user, and $\hat{d}_{ij|\mathrm{reps}(m_j)} = \widehat{ud}_{ij}$ and $\hat{a}_{ij|\mathrm{reps}(m_j)} = \widehat{ua}_{ij}$, when the user does not know any directors or actors of the movie. The general case is when the user knows some of the directors and actors of the movie. Formally, the final directors and actors scores $\hat{d}_{ij|\mathrm{reps}(m_j)}$ and $\hat{a}_{ij|\mathrm{reps}(m_j)}$ are calculated by considering both the user preferences and the public opinion, i.e., a weighted average between the scores of the known entities and the unknown entities:

$$\hat{d}_{ij|\mathrm{reps}(m_j)} = \delta_{ud} \cdot (\widehat{ud}_{ij} + bias_i) + (1 - \delta_{ud}) \cdot \hat{d}_{ij}, \quad (70)$$

$$\hat{a}_{ij|\mathrm{reps}(m_j)} = \delta_{ua} \cdot \left(\widehat{ua}_{ij} + bias_i\right) + (1 - \delta_{ua}) \cdot \hat{a}_{ij}, \quad (71)$$

where the constants δ_{ud} and δ_{ua}, represent the contribution of the unknown directors and actors to the computation of $\hat{d}_{ij|\text{reps}(m_j)}$ and $\hat{a}_{ij|\text{reps}(m_j)}$ respectively. They are computed as:

$$\delta_{ud} = \frac{|D_j - D^i|}{|D_j|}, \qquad \delta_{ua} = \frac{|A_j - A^i|}{|A_j|}, \tag{72}$$

where $|D_j - D^i|$ is the number of directors on movie m_j that the user u_i does not know and $|A_j - A^i|$ is the number of actors on movie m_j that the user does not know. Once again, the user bias $bias_i$ is accounted in order to adjust the public opinion on directors and actors to the user personal standards.

Modelling a movie popularity $\text{pop}(m_j)$ with social-media trends

So far, the predicted rating $\widehat{pr}_{ij}$ captures an incomplete set of indicators about the movie, missing a key indicator which is the trendiness of that movie. Krauss et al. (2008) has showed that movie trendiness is projected in Oscar nominations, which are generally associated with highly rated movies. The set $T(m_j)$, containing tweets targeting movie m_j, can be used to predict its reputation. Oghina et al. (2012) have shown that the fraction of likes/dislikes is the strongest feature for predicting IMDb movie ratings from social-media. Following this remarks, we consider the popularity of a movie m_j to be measured as

$$\text{pop}(m_j) = \frac{|pos_{m_j}|}{|tweets_{m_j}|}, \tag{73}$$

where $|pos_{m_j}|$ is the number of positive tweets referring the movie m_j and $|tweets_{m_j}|$ is the total number of tweets referring m_j.

5.3 Evaluation

For evaluation purposes reviews are split at sentence level[28], words are reduced to the same stem and words are labelled according to its word family: adjectives, adverbs, verbs and nouns[29].

5.3.1 Datasets

For the proposed RS framework we are interest in performing a rating inference on users' reviews into a RS algorithm. To this end, it's required that the dataset contains information about the user, product, review content and the respective rating. Here, one of the challenges is the lack of

[28] Reviews are split at sentence level with the tool NLTK (http://nltk.org).

[29] Part of Speech Tagging with the tool Freeling 3.0 (http://nlp.lsi.upc.edu/freeling/).

adequate dataset. Most well-known available sentiment analysis datasets contain no information regarding the rating of the review, similarly the RS datasets only provide ratings with no associate users' reviews (Pang and Lee, 2004; Turney, 2002). To resolve this constrain, we extracted an IMDb dataset that follows the necessary requirements for the proposed model. The data is available for research purposes at http://novasearch.org/datasets/.

IMDb contains a high amount of data to which numerous users only review a few number of movies. Additionally, not every movie contains helpful reviews. To this end, it was implemented an extractor that obtains reviews according to the top rated movies and users – users with higher helpfulness (or usefulness[30]) level are chosen. The implementation is described in Algorithm 1 (chapter 2 – Section 2.4.4. Datasets and Pre-processing Steps). The IMDb-Extracted dataset is described in Table 16[31].

To evaluate the proposed recommendation algorithm we selected two state-of-the-art datasets that will be used in our experiments: 698,210 movie and music reviews from the Amazon electronic commerce website[32] and, for comparison purposes, 53,112 movie reviews from Jakob et al. (2009) experiments (Table 17). More specifically,

- Amazon (Jindal and Liu, 2008): The dataset includes 698,210 reviews from 3,700 users. The dataset covers 8,018 products (movies and music). Reviews are rated in a scale from 1 to 5.
- IMDb-TSA09 (Jakob et al., 2009): This data covers 2,731 movies and 509 users. The reviews are rated in a scale from 1 to 10.

Table 16: Detailed information of IMDb-Extracted dataset split.

Split	#Reviews	Description
A	335,975	Train sentiment analysis algorithm.
B	335,975	Test sentiment analysis algorithm / Train recommendation system.
C	417,147	Train recommendation system (no explicit ratings).
D	335,976	Train recommendation system.
E	201,586	Test recommendation system.
F	102,634	Validate recommendation system.

[30] Given a user review other users may evaluate its' usefulness.

[31] Notice that splits A, B and D correspond to splits A, B and C described in chapter 4 – Section 4.4.1. Datasets.

[32] http://amazon.com.

Table 17: Detailed information of Amazon and IMDb-TSA09 dataset splits.

Split	Amazon	IMDb-TSA09	Description
A	184,996	23,599	Train sentiment analysis algorithm.
B	182,651	23,601	Test sentiment analysis algorithm / Train recommendation system.
C	236,450	Split A	Train recommendation system combined with split B
D	94,113	5,912	Test recommendation system.

To evaluate the cold-start recommendation algorithm for the linked-entities recommendation it was selected the following IMDb movie reviews:

- We focused the extraction process on users who have rated at least one of a selection of 60 new movies, finalists on 5 popular movie awards ceremonies: the 2014 editions of *The Golden Globes*, *The Critic's Choice Awards, The BAFTA Film Awards*, *The Independent Spirit Awards* and *The Oscars*.
- We selected such movies so we could additionally to IMDb reviews capture a great number of tweets in that small time period. Hence, between January 2014 and March 2014 it was crawled 52,236 tweets that mention the new movies.

In total, to evaluate the Linked-Entities Recommendation model we obtained a dataset with 1,064,766 ratings, given by 2,909 users to 60 new movies and 46,843 old movies. The computation of the actors and directors reputation for the new movies used a total of 124,236 IMDb reviews: we considered a total of 225 actors and 169 directors corresponding to the 60 new movies.

5.3.2 Baselines

To determine the importance of having ratings inferred by a sentiment analysis algorithm we evaluate three matrix factorization approaches:

- Baseline: recommendation with users' explicit ratings (only).
- Sentiment-ratings: recommendations with inferred ratings from a sentiment analysis algorithm.
- Probabilistic sentiment-ratings: inferred ratings in which each rating is associated to a confidence level. This corresponds to the full regularized matrix factorization with probabilistic sentiment-ratings.

- IMDb-TSA09: Jakob et al. (2009) proposed a framework that extracts opinions from free-text reviews to improve the accuracy of movie recommendations.

As to the, Linked-Entities Recommendation model the following baseline methods are applied:

- KNN: K-Nearest Neighbour algorithm is known to be successful in hybrid recommendations (Amatriain et al., 2009).
- FM1 and FM2: Recommendations with no social-media feedback where it is used the formal recommendation model – user-movie ratings are predicted only by exploring user preferences. We distinguish the case where θ_d, θ_a and θ_g are all equal to 1.0, also these weights were estimated with 10-fold cross validation resulting in $\theta_d = 0.35$, $\theta_a = 0.2$ and $\theta_g = 0.45$ constants that control the contributions of directors, actors and genre respectively (section 5.2.1).

5.3.3 Evaluation Metrics

The evaluation of the sentiment analysis algorithm is given by the standard deviation measures precision (P), recall (R) and $Fscore$ which the latest is the harmonic mean between P and R,

$$Fscore = \frac{2 \cdot P \cdot R}{(P + R)}. \tag{74}$$

To evaluate the recommender system algorithm the statistical measure root mean square error (RMSE) and mean average error (MAE) is applied,

$$MAE = \frac{\sum_{\hat{r}_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}|}{\#\hat{R}}, \tag{75}$$

$$RMSE = \sqrt{\frac{\sum_{\hat{r}_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2}{\#\hat{R}}}. \tag{76}$$

where $\hat{R}$ represents the set of ratings, r_{ui} represents the rating given by the user u to movie i and $\hat{r}_{ui}$ represents the rating predicted by the recommendation system algorithm. Small values of RMSE indicate a more accurate system. Furthermore, we also use the Mean Average Error (MAE) to assess rating predictions quality between the real ratings and predicted ratings.

5.4 Experiments

In this section it is discussed the following tasks: multiple-Bernoulli sentiment classification for the Amazon and IMDb-Extracted datasets, sentiment-based recommendations for the datasets

Amazon, IMDb-TSA09 and IMDb-Extracted and, finally, a qualitative analysis about probabilistic sentiment-ratings versus explicit ratings.

5.4.1 Multiple-Bernoulli Sentiment Analysis

In order to integrate the inferred ratings in the recommendation system it is required to evaluate the sentiment analysis (SA) framework performance. To evaluate the sentiment algorithm for the datasets Amazon, IMDb-TSA09 and IMDb-Extracted it was selected the split A and B (Table 16 and Table 17) to train the algorithm and test it.

Figure 39 shows the evaluation results for the sentiment analysis framework for the IMDb-extracted. Figure 41 presents the ratings confusion matrix between the predicted ratings and the actual ratings and Table 18 the ratings confusion matrix for the Amazon dataset, where the values in bold in the diagonal correspond to the correctly classified reviews. If the sentiment analysis algorithm would be completely accurate, only the diagonal would be active. The right-most column and the bottom column present the number of ratings for a given level. For example, there are 95,248 ratings of 5 rating level and the algorithm correctly predicted 29,463 of these ratings. We recall that in the performed multi Bernoulli classification we aim to distinguish between similar ratings which entails a more challenging task than a binary (positive vs. negative) (Sparling, 2011). Additionally, in Figure 39 similarly to the observed results in Figure 41 the performance for lower ratings is not as good as for higher ratings which is consistent with Zhang et al. (2010) findings. We believe this is due to the negative reviews being highly sparse in relation to the positive reviews. The IMDb-Extracted shows a mean precision, recall and F-score of 0.72, 0.68 and 0.65 respectively.

Users' reasoning upon providing a rating and providing the associated review tends to frequently differ. On rating a movie some users can prove to be more demanding, or generous, than others. Nonetheless, we aim at inferring a rating to a recommendation algorithm in which ratings are re-adjusted through rating biases. Also in the paradigm of recommendation algorithms it's not critical to infer the exact rating but to correctly identify the patterns in users' likes and dislikes.

A more convenient visualization of the confusion-matrix is presented by Figure 40 and Figure 41. Considering the predicted rating and the actual rating the confusion matrices illustrates the cross-rating interference. It can be observed that incorrectly predicted ratings are usually in the neighbouring ratings. For example, for the IMDb dataset the rating 8 in the diagonal has 0.038

which is followed by 0.039 (rating 7) and 0.034 (rating 9), in which the matrix diagonal contains the correctly predicted ratings. This is also justified by the nature of the data since users might write a review with a rating of 4 while others not as demanding write a similar review with a rating of 5 (Pang and Lee, 2005). Furthermore, it's observable that the diagonal and the surrounding elements hold a higher accuracy, showing a low interference across distant ratings. These properties are fundamental to include the probabilistic sentiment-ratings into the matrix factorization procedure. The relaxed nature of our sentiment analysis approach places most of the predictions in the correct rating or in neighbouring ratings. This preserves the trend that is present in review data.

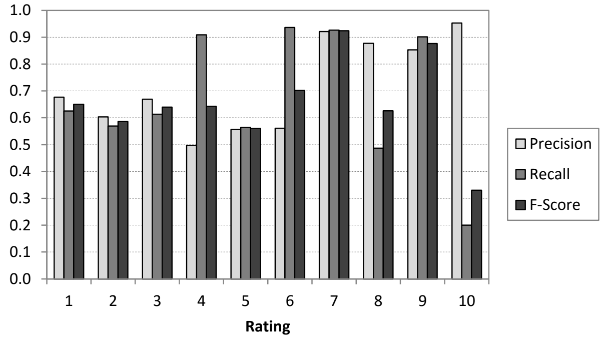
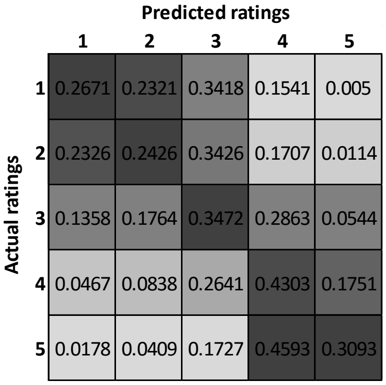


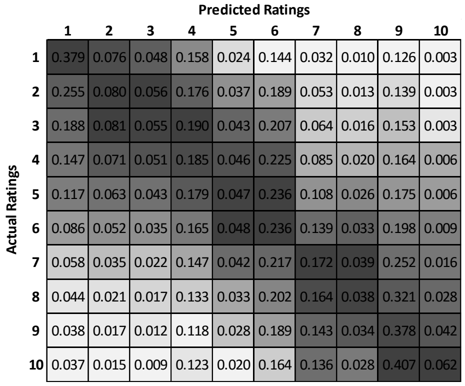
Figure 39: Multiple-Bernoulli Sentiment analysis results (IMDb-Extracted).

Table 18: Ratings confusion-matrix (Amazon).

		Predicted rating Values					
		1	2	3	4	5	
True rating values	1	**1723**	1497	2205	994	32	6451
	2	2000	**2086**	2946	1468	98	8598
	3	2931	3806	**7494**	6178	1173	21582
	4	2369	4253	13410	**21848**	8892	50772
	5	1695	3892	16447	43751	**29463**	95248
		10718	15534	42502	74239	39658	



		Predicted ratings				
		1	2	3	4	5
Actual ratings	1	0.2671	0.2321	0.3418	0.1541	0.005
	2	0.2326	0.2426	0.3426	0.1707	0.0114
	3	0.1358	0.1764	0.3472	0.2863	0.0544
	4	0.0467	0.0838	0.2641	0.4303	0.1751
	5	0.0178	0.0409	0.1727	0.4593	0.3093



		Predicted Ratings									
		1	2	3	4	5	6	7	8	9	10
Actual Ratings	1	0.379	0.076	0.048	0.158	0.024	0.144	0.032	0.010	0.126	0.003
	2	0.255	0.080	0.056	0.176	0.037	0.189	0.053	0.013	0.139	0.003
	3	0.188	0.081	0.055	0.190	0.043	0.207	0.064	0.016	0.153	0.003
	4	0.147	0.071	0.051	0.185	0.046	0.225	0.085	0.020	0.164	0.006
	5	0.117	0.063	0.043	0.179	0.047	0.236	0.108	0.026	0.175	0.006
	6	0.086	0.052	0.035	0.165	0.048	0.236	0.139	0.033	0.198	0.009
	7	0.058	0.035	0.022	0.147	0.042	0.217	0.172	0.039	0.252	0.016
	8	0.044	0.021	0.017	0.133	0.033	0.202	0.164	0.038	0.321	0.028
	9	0.038	0.017	0.012	0.118	0.028	0.189	0.143	0.034	0.378	0.042
	10	0.037	0.015	0.009	0.123	0.020	0.164	0.136	0.028	0.407	0.062

Figure 40: Normalized predicted ratings distribution (Amazon).

Figure 41: Predicted ratings distribution for the Multiple-Bernoulli classification (IMDb-Extracted).

5.4.2 Sentiment-based Recommendations

In the first experiment with the sentiment-based recommendation we will analyse the datasets Amazon and IMDb-TSA09. However, IMDb-Extracted characterizes a larger dataset which can provide more in-depth analysis. Regarding the splits for Amazon dataset we start by defining the following setting:

1. **RS Lower bound (LB)**: the recommendation algorithm is trained on a set of trainings corresponding to split C (Table 17). This establishes the error lower bound.
2. **RS Upper bound (UB)**: the recommendation algorithm is trained on the maximum number of ratings, corresponding to the union of the splits B and C (Table 17). This establishes the error upper bound.
3. **RS + SA (SA)**: the recommendation system is trained on explicit ratings (split C) and ratings inferred from unrated reviews (split B). In this experiment, all ratings of split B are withheld.

The summary in Figure 42 provides a strong message: the sentiment analysis of the textual content of users' reviews, when joined together with ratings explicitly provided by the users, can indeed improve recommendations. In respect to the replacement of explicit ratings by inferred ratings the error observed with SA brings into light an interesting result. When using just explicit ratings for the LB and UB, the RMSE was 1.0092 and 0.9963, respectively. However with the inferred ratings (SA), we obtained a lower RMSE of 0.9845. Hence, it is noticeable that inferred ratings can better accommodate the uncertainty of the explicit rating assigned by users. This is explained by the fact that some ratings are strongly biased by users and, the review textual content provides a more complete opinion. For example, users' reviews that focus on answering other reviews or unrelated information about the movie (actors previous performances).

Figure 43 provides a detailed view of how the threshold value influences the recommendations quality (RMSE). The upper bound (UB) and lower bound (LB) correspond to the RS recommendations where the SA algorithm had no influence. The RS+SA curve includes the ratings from split C (Table 17). When SA inferred ratings are added to the set of LB ratings, we see that the recommendation framework can indeed improve the overall RMSE. In Figure 43 for a threshold $th = 0.0$, all inferred ratings are used by the recommendation system; for a threshold $th = 0.5$, only the inferred ratings with probabilities 0.0 and 1.0 are used. As the threshold th increases, ratings with probabilities near 0.5 are ignored (they are ambiguously positive or negative). Thus, the higher the threshold, the fewer inferred ratings are considered (# of SA ratings curve). The RS+SA curve illustrates how the analysis of unrated reviews can indeed improve the RMSE of the computed recommendations. As we exclude ratings closer to a probability 0.0 and 1.0, the RMSE increases until it reaches its worst value for $th = 0.5$. This corresponds to considering 1 rating star and 5 rating star inferred ratings (in Amazon rating scale corresponds to the lower and highest rating star). We reason for the achieved RMSE value is related to the high amount of 5 rating starts that the Amazon dataset holds, and to the wrath of some users when writing 1 rating star review. To best examine this behaviour of the RS+SA curve, Figure 44 presents an insightful look into the performance of the comments analysis algorithm. Precision is quite high for 5 rating star but is extremely low for the other rating levels – this is critical because the recommendation algorithm needs both low and high rating values. Recall is below 30% for 1 and 2 ratings level and above 30% 30% for 3, 4 and 5 rating level. These recall values generate a small set of 1 and 2 rating levels. Note that precision and recall measure the exact match between the actual ratings and the inferred ratings. However, for the recommendation algorithm what is most important is the average error between the actual rating and the inferred rating. In other words, we need to consider the mean absolute error of each predicted rating. Also, Figure 44 illustrates the MAE curve (mean absolute error) between the predicted ratings and the true ratings. One can see that for 2 and 4 rating level, the average distance between the predicted and the true rating is less than 1. Hence, this graph shows that noisier data is concentrated on ratings with 1 and 5 rating level, which clarifies the RS+SA curve behaviour.

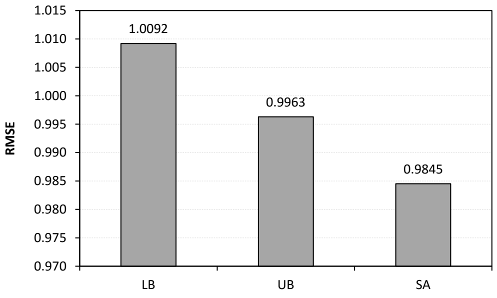


Figure 42: RMSE for LB, UB and SA blend with RS (Amazon).

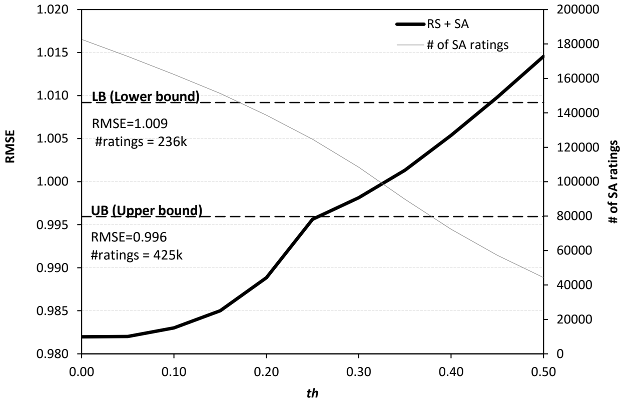


Figure 43: RMSE of the recommendations versus the sentiment analysis output. As the threshold increases, less $\Phi(re_i)$ ratings are included (Amazon).

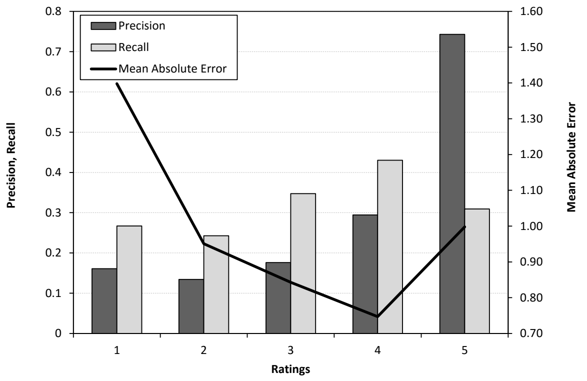


Figure 44: Sentiment analysis precision and recall per rating. The MAE measure indicates the average distance to the true ratings (Amazon).

For comparison purposes we have performed the sentiment-based recommendation experiments with Jakob et al. (2009) IMDb dataset (Table 17). While Jakob et al. (2009) approach explores media related information such as genre and actors, it does not take into account unrated reviews. In this experiment, we trained the sentiment analysis algorithm with the split A (Table 17) and, for the recommendation algorithm the split A was used to train individually, and combined with the inferred ratings from split B (Table 17). Finally, the split D was used to evaluate the recommendation algorithm.

In the first experiment (Figure 45) our baseline performed better (RMSE=1.819) than Jakob et al. (2009) where it was observed the full set of ratings. Also, when it was included the entire set of sentiment ratings (RMSE=1.823) our approach was slightly better than Jakob et al., 2009. In a second experiment (Figure 46), with the goal of evaluating the influence of the sentiment analysis performance, it was used 50% of explicit ratings and 50% of inferred ratings to train the recommendation algorithm. We achieved an error of 1.886 which is slightly better than just using 50% of explicit ratings (RMSE=1.896). Since, IMDb-TSA09 has a small set of training reviews, we believe that a finer grain classifier or more training data can further increase this gap. These experiments show how the proposed approach compares to existing ones: despite being competitive, it can also extract extra information from the text reviews to infer unknown ratings, which makes it applicable to a wider range of scenarios.

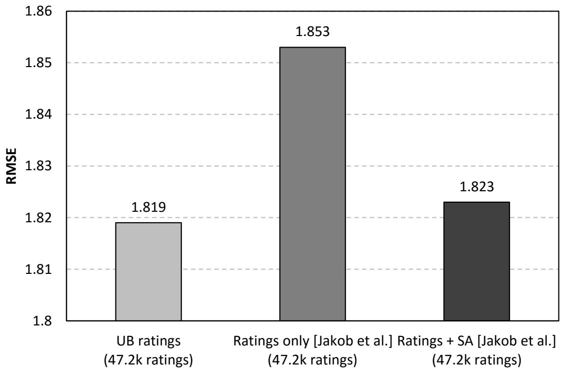


Figure 45: RMSE when only ratings are used (IMD-TSA09).

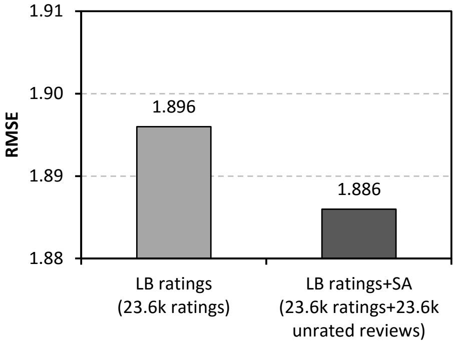


Figure 46: RMSE when sentiment analysis inferred ratings are in included (IMDb-TSA09).

5.4.3 Finer-grain Sentiment-based Recommendation

In this section it will be detailed the sentiment-based recommendation experiments for the large scale dataset IMDb-Extracted (Table 16). This dataset performs a more in-depth analysis of the sentiment-based inferred ratings ability to improve product recommendations. It is analysed a set experiments where the proportion of inferred ratings differs. To this aim, we want to investigate if the use inferred sentiment-based recommendations other than explicit ratings improves the recommender systems performance.

To illustrate the importance of probabilistic ratings when using a large dataset we perform an in-depth analysis. To this end, it was implemented and evaluated three matrix factorization approaches:

- **Baseline**: recommendations with just users' explicit ratings.
- **Sentiment-ratings**: recommendations with inferred ratings from the sentiment analysis framework (as the experiments performed in Section 5.4.2 for datasets Amazon and IMDb-TSA-09).
- **Probabilistic sentiment-ratings**: inferred ratings in which each rating is associated to a confidence level. This corresponds to the full regularized matrix factorization with probabilistic sentiment-ratings.

Sentiment-based Recommendations: $\#\boldsymbol{R_{ra}} > \#\boldsymbol{R_{rev}}$

This first experiment examines the proposed recommendation framework in a setting where the number of explicit ratings $\#R_{ra}$ is higher than the number of text-reviews $\#R_{rev}$. We trained the recommendation system on a data subset of explicit ratings (D and B), and the C subset of

IMDb reviews that have no rating information. The RMSE is measured on the test subset E with 201,586 ratings.

Table 19: IMDb-Exctracted subsets for $\#R_{ra} > \#R_{rev}$.

Dataset	Description	#Ratings	#Reviews
DB	Model trained with explicit ratings from split B and D	671,951	-
DB+Ci	Model trained with explicit ratings from split B and D and inferred SA ratings from split C	671,951	417,147

The baseline model evaluated on the data subset DB with just explicit ratings achieved an RMSE of 2.099, see Figure 47. Ratings were then inferred from the text reviews of the data subset C. When these sentiment-ratings are simply added to the factorization procedure we can observe that the error increases (RMSE=2.122). The explanation for this result is related to the fact that these inferred ratings in the recommendation algorithm are being treated as explicit ratings, i.e., $\theta_{ij} = 1$. This is equivalent to assigning a total confidence to the inferred ratings. However, this is obviously too optimistic and the uncertainty of the sentiment analysis must be taken into account. The third result in Figure 47 adds the sentiment-ratings and the probability that this rating is correct $\theta_{ij} = [0,1]$ as a regularization factor in the matrix factorization procedure. The result RMSE= 2.094 confirms that regularizing the contribution of each sentiment-rating in the optimization procedure can indeed improve the accuracy of the overall recommendations.

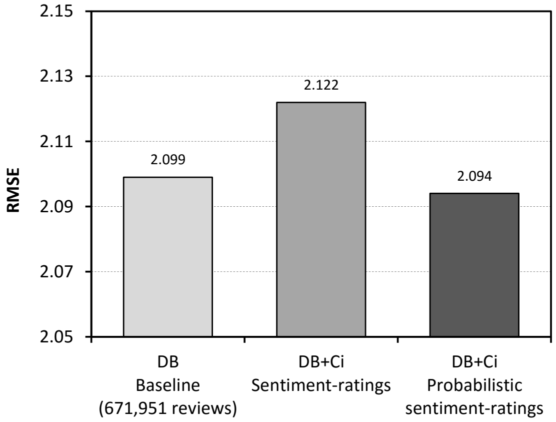


Figure 47: RMSE for baseline and review-based inferred ratings ($\#R_{ra} > \#R_{rev}$).

Sentiment-based Recommendations: $\#\boldsymbol{R_{rev}} > \#\boldsymbol{R_{ra}}$

This recommendation experiment examines the proposed framework in a setting where the number of text-reviews $\#R_{rev}$ increases in relation to the number of explicit ratings $\#R_{ra}$.

Table 20: IMDb-Exctracted subsets for $\#R_{rev} > \#R_{ra}$.

Dataset	Description	#Ratings	#Reviews
D	Model trained with explicit ratings from split D	335,976	-
D+Bi	Model trained with explicit ratings from split D and inferred SA ratings from split B	335,976	335,976
D+Ci	Model trained with explicit ratings from split D and inferred SA ratings from split C	335,976	417,147
D+BCi	Model trained with explicit ratings from split D and inferred SA ratings from split B and C	335,976	753,123

The details concerning the datasets for this experiment are in Table 20. The recommendation system is trained on the subset D of explicit ratings, and an increasing number of ratings inferred from the reviews. In this experiment, the explicit ratings of the B subset were withheld.

The baseline recommendation system (the subset D with just explicit ratings) achieved an RMSE of 2.086, Figure 48. When new review data is added to train the recommendation system, one can observe the same behaviour as in the previous experiment: the error increases when only the sentiment-ratings are included, but the error decreases when the confidence levels are included (probabilistic sentiment-ratings). Adding the review subset C we obtained RMSE=2.082, with the B subset the RMSE dropped to 2.08. Finally, Figure 48 shows that with probabilistic sentiment-ratings we were able to improve the RMSE from 2.086 to 2.079. Hence, this outcome supports the intent of our concept, also the intuition that users' reviews should not be disregarded by recommendation algorithms (Leung et al., 2006; Jakob et al., 2009).

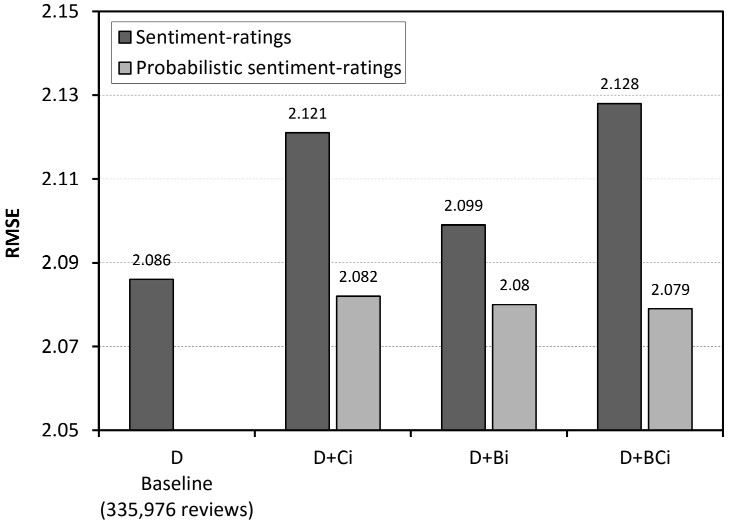


Figure 48: RMSE for baseline and review-based inferred rating ($\#R_{rev} > \#R_{ra}$).

Probabilistic Sentiment-Ratings versus Explicit Ratings

The consolidation of the two previous experiments brings into light a surprising result concerning the replacement of the explicit ratings by probabilistic sentiment-ratings. When using just explicit ratings in the first experiment (DB baseline, Figure 47) the RMSE was 2.099. However, in the second experiment we considered the explicit ratings of the D subset and the probabilistic sentiment-ratings of the subset B (D+Bi, Figure 48) and obtained a lower RMSE of 2.08. Thus, the proposed framework managed to improve recommendation results by replacing the user ratings with the output of a sentiment analysis algorithm that predicts each rating based in users written opinion.

We argue that the proposed sentiment-based regularization of the factorization matrix can better accommodate the uncertainty regarding the explicit rating assigned by users. The uncertainty of a user rating is hidden in the text review to be quantified by sentiment analysis. For instance, often the users' reviews focus on answering other reviews or diverge to non-related information to the movie. A review with a high amount of content non-related to the explicit rating might have a doubtable explicit rating, however, its confidence level is still maximum (θ_{ij}) in the recommendation algorithm. In the following Table 21 we present some examples from split B (Table 16) with content from users' reviews and respective explicit rating where the assertiveness in the user' review is captured with an associated confidence level.

Table 21: Examples of review textual content from IMDb-Extracted (split B).

Rating	Movie	Review Content
10	Star Wars: Episode III	"I personally am more of a fan of the original trilogy than what I have been of the prequels. Although I did enjoy (...) they definitely were not as well done as A New Hope or Empire. I think the general criticisms of the first two prequels was lack of good story, and poor acting…"
8	Hell's Kitchen	"This is not a creative cooking contest. It's not supposed to be. (...)"
2	The Ringer	"I really like Johnny Knoxville, love the series Jackass, some really funny s+;* and I'm so happy to see him doing more movies (...). But this film was AWFUL! ..."

5.4.4 Linked-Entities Recommendation

To evaluate the proposed linked-entities recommendation model we leveraged on data concerning 60 Oscar upcoming movies collected from different sources. Here, the method MRep observes the contribution of the movie reputation $\mathrm{pop}(m_j)$, which is inferred from tweets; ERep observes the contribution of entities reputations $\hat{a}_{ij|\mathrm{reps}(m_j)}$ and $\hat{d}_{ij|\mathrm{reps}(m_j)}$, which were computed from IMDb reviews; and, FRep uses the full spectrum of social-media reputation where both movies popularity and entities reputation are considered.

Monitoring the Popularity of New Movies. We compared the predicted popularities, concerning Equation 66 (rating prediction) with the average IMDb ratings of the target movies, captured several months after the movies' release data. Figure 49 shows the predicted ratings and the IMDb average ratings. From it, we can observe the overall deviation of the predicted ratings. Overall, the MAE is 0.59, which is in the same error range found in literature (Oghina et al., 2012). The prediction errors varied from 0.026 ("Blue is the Warmest Colour") to 2.29 ("Her"). By analysing the overall error, we can observe that movies with lower IMDb ratings are more likely to have a higher prediction error: for instance, while "Blue is the Warmest Colour" has an average IMDb rating of 8.0, examples of high error such as "The Invisible Woman" (MAE=2.01) and "Computer Chess" (MAE=1.73), have an average IMDb rating of 6.3. This leads us to believe that Twitter users are more likely to share positive tweets about movies than negative tweets, which makes our method more precise for highly rated movies.

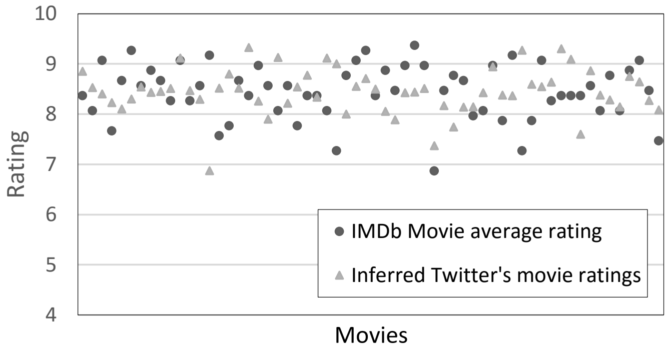


Figure 49: Twitter-based Movie Ratings vs IMDb Movie Ratings.

Figure 50 plots the MAE and Fscore curves for a range of value. Both MRep and FRep present the best results for the importance of movies popularity (α_t) values which are below or equal to 0.40 – after this point both MAE and Fscore start to deteriorate. For MRep, both the best MAE and Fscore values are obtained at $\alpha_t = 0.35$ (MAE of 1.2266 and Fscore of 87.2%). For FRep, the best Fscore is also obtained at 0.35 (87.7%), while the best MAE is obtained at 0.20. These results suggest that the popularity of movies has a significant influence when predicting user-movie cold-start ratings. However, if the general opinion (α_t) is considered too much against the personal preferences, the predicted user-movie rating drops the personalization component, leading to less accurate predictions. For subsequent experiments, we set $\alpha_t = 0.35$.

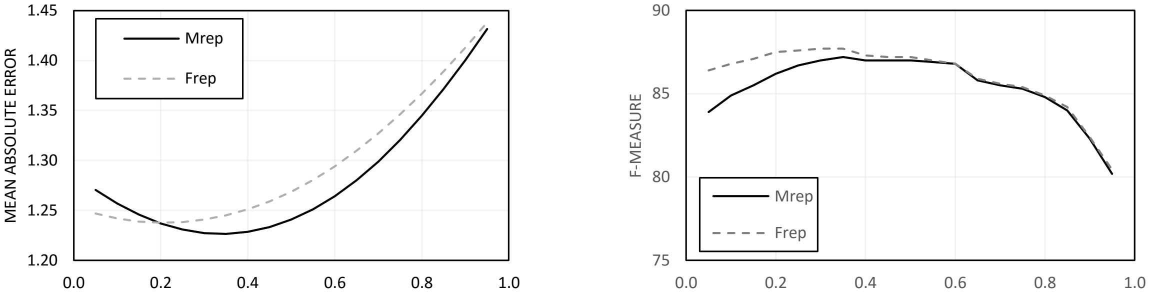


Figure 50: Estimation of the importance of movies popularity (α_t) to the movie rating prediction.

Table 22 shows the methods that consider social-media information (inferred by reputation analysis described in chapter 4) outperform the baselines. In terms of rating prediction, MRep presents the best MAE results. FRep, for instance, presents the best results in all recommendation measures, with a total Fscore of 87.7%. The improvement in recall values for the social-media methods relatively to the baselines shows that the reputation of movies, directors and actors help especially in identifying great movies, which are more usually relevant for users.

Table 22: Linked-Entities comparative results.

	MAE	Precision (%)	Recall (%)	Fscore (%)
KNN	1.3933	70.1	86.5	78.3
FM1	1.3058	70.3	85.2	77.8
FM2	1.2962	71.4	87.7	79.6
MRep	**1.2266**	**76.0**	**98.5**	**87.2**
ERep	1.2536	75.6	96.1	85.9
FRep	1.2450	**76.0**	**99.4**	**87.7**

Cases of extreme user cold-start. When users have not rated many movies, their preferences cannot be well modelled: these users suffer from the cold-start problem. This happens mostly, but not exclusively, for users who are new to the system. We simulate a scenario where all our 2,909 test users suffer from the cold-start problem by not considering their previous ratings and perform experiments with MRep, ERep and FRep. Furthermore, we compute Random recommendations 20 times and average the respective results for comparison, as these are often used as baselines in these scenarios. Table 23 shows both the obtained MAE and Fscore results for each method. Note that user's bias cannot be considered in this scenario since there are no previously given ratings from any user.

Table 23: Extreme user-side cold-start.

	MAE	Fscore
Random	3.21	39.81%
Mrep	1.62	74.75%
Erep	1.77	63.50%
Frep	1.64	69.38%

From the main methods, ERep obtains the worst results (MAE of 1.7741 and Fscore of 63.5%) while MRep obtains the best results (MAE of 1.6236 and Fscore of 74.75%). FRep presents intermediate results. When compared to Random recommendations, even the weakest method (i.e., ERep) outperforms it by reducing the MAE to almost half and improving the Fscore results by 23.7%. Overall, these results show that the popularity of new movies is a good baseline predictor of the quality of a movie and is useful for recommending movies when the user preferences are not known. While the reputation of the new movies directors and actors present weaker results, these still prove to be an average predictor of a movie quality, as a 63.5% recommendation accuracy is very good for a scenario where there is no information on users, improving considerably when compared to random recommendations.

5.5 Sentiment-based Recommendation Visualization

Figure 51 shows the frontal page of the SentiMovie sentiment-based recommendation demonstrator. The left bar presents the sentiment analysis algorithms performance – precision, recall and F1 – and for the recommendation algorithm the obtained RMSE (Root Mean Squared Error) (baseline algorithm and probabilistic sentiment-based recommendations). The user can observe in a pie chart the percentage of the sentiment ratings that were predicted based on the sentiment analysis of users' reviews. It is observed that there is a tendency for a user to provide a review when the user has a strong positive or negative sentiment about the movie (product). In the pie chart the sentiment predictions of ratings 10 and 1 is 25% and 13%, respectively. Additionally, SentiMovie allows the user to search for different movie titles or click on the cover of a suggest movie from the SentiMovie front-page.

Figure 52 presents a review with an inferred sentiment rating of 10 in which the most positive and negative sentiment words are depicted in green (enclosed in rectangles) and red (enclosed in circles) respectively. Once navigating in the SentiMovie visualization interface and selecting a movie the user observes the inferred sentiment ratings at each rating level. In Figure 53 the x axis represents the rating levels and the y axis the volume of inferred sentiment ratings. In addition, taking the assumption that an inferred rating above 6 is positive otherwise negative, Figure 53 is shows in a chart pie the percentage of positive and negative inferred sentiment ratings. Hence, a binary sentiment evaluation visualization.

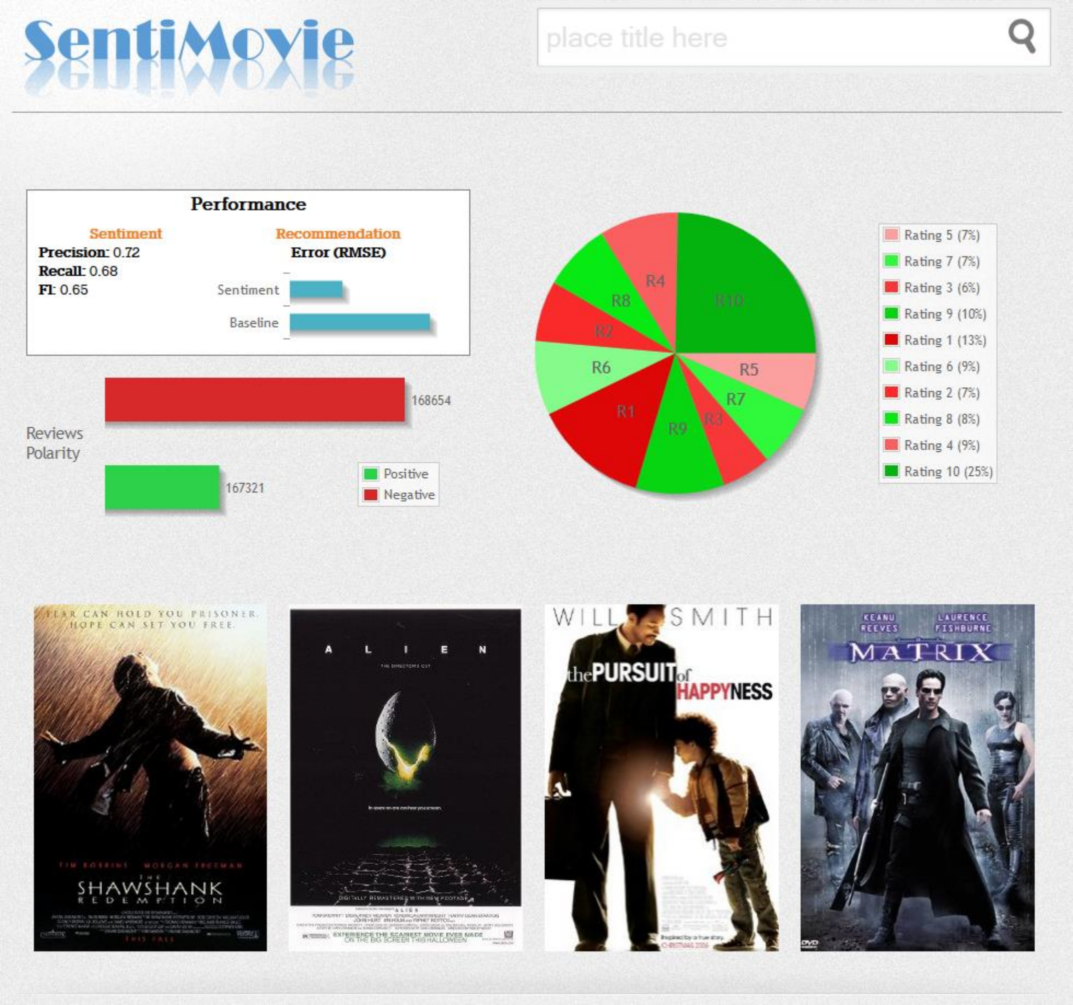
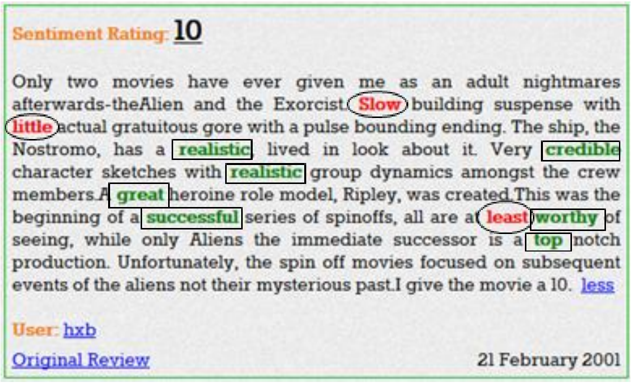


Figure 51. SentiMovie: visualization of users' movies reviews analysis using sentiment analysis algorithms in a sentiment-based recommendation RS framework (Section 5.1.2: Matrix Factorization with Sentiment-based Regularization).



Sentiment Rating: 10

Only two movies have ever given me as an adult nightmares afterwards-theAlien and the Exorcist. Slow building suspense with little actual gratuitous gore with a pulse bounding ending. The ship, the Nostromo, has a realistic, lived in look about it. Very credible character sketches with realistic group dynamics amongst the crew members.A great heroine role model, Ripley, was created.This was the beginning of a successful series of spinoffs, all are at least worthy of seeing, while only Aliens the immediate successor is a top notch production. Unfortunately, the spin off movies focused on subsequent events of the aliens not their mysterious past.I give the movie a 10. less

User: hxb

Original Review 21 February 2001

Figure 52. Sentiment rating prediction. Positive (enclosed in rectangles) and negative (enclosed in circles) sentiment words are highlighted in the review.

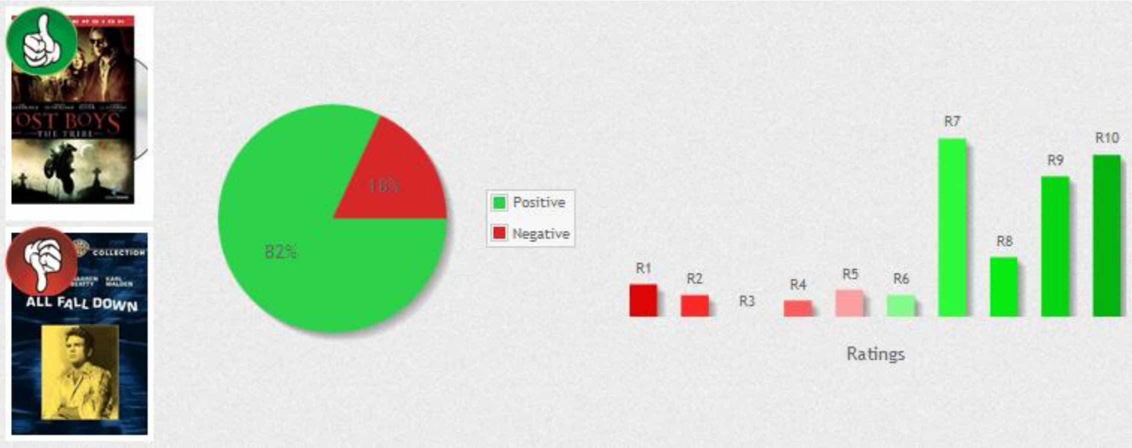


Figure 53. Sentiment-based recommendations.

As seen in this chapter sentiment-based ratings are included in a recommendation system. To this end when a user selects a movie we observe the thumbnails of two other movies (Figure 53). These thumbnails are overlapped with a thumbs up/thumbs down icons. Hence, a positive/negative movie recommendation. This visualization allows the user to receive sentiment-based recommendations through the analysis of other users' viewpoints. For example, when observing the information about the movie "The Pursuit of Happiness" the sentiment-based recommender system suggests "Elisabeth Town" as a positive recommendation and, "Sex and the City" as a negative recommendation. Moreover, a given movie can be in the spectrum of positive suggestions (recommendations) as also in the negative spectrum. This visualization allows us to observe how this type of users' textual content (reviews) leverages in the existing sentiment relation between different movies (products) and, as a consequence in the movies recommendations.

5.6 Summary

In this chapter, we demonstrated how a sentiment analysis algorithm can be integrated with a recommendation system. We proposed to integrate in a single recommendation framework both explicit ratings and free-text comments with no associated rating. In this work, user ratings were inferred from text comments and merged into the recommendation algorithm. Our approach was evaluated in 1,729,293 movie reviews and compared with explicit ratings (only), inferred ratings and explicit ratings, and finally, probabilistic inferred ratings and explicit ratings. To this end, probabilistic inferred ratings materialized the uncertainty of a user opinion in a rating, and the obtained results confirm that sentiment inferred ratings were able to improve recommendation algorithms.

The recommendation algorithm used in the SentiMovie demonstrator was published at Peleja et al. (2012). An early version of the recommendation algorithm was published in a Multimedia Systems journal paper Peleja et al. (2013). In summary, the contributions of this chapter were published at:

Peleja, F., Dias P. and Magalhães J. 2012. "A Regularized Recommendation Algorithm with Probabilistic Sentiment-Ratings." In *Proceddings on the IEEE 12th International Conference on Data Mining Workshops (ICDMW/SENTIRE)*, 701–8. Brussels, Belgium. doi:10.1109/ICDMW.2012.113.

Peleja, F., Dias P., Martins F. and Magalhães J. 2013. "A Recommender System for the TV on the Web: Integrating Unrated Reviews and Movie Ratings." *Journal of Multimedia Systems, Springer-Verlag New York,* 19 (6):543–58. doi:10.1007/s00530-013-0310-8.

Moreover, this research has progressed to incorporate named entities in a cold-start recommendation problem. We show that entities reputation can be a good indicator in the cold-start problem of movies recommendation. The work related to this approach is now under review:

Santos, J., Peleja F. and Magalhães J. "Monitoring Social-Media for Cold-Start Recommendations", *Multimedia Tools and Applications, Special Issue on Immersive TV* (*under review).*

6 Conclusion

This thesis revolved around three research themes: extract sentiment words from online reviews, predict reputation of entities and investigate sentiment-based recommendations. In the context of online commerce textual content that expresses a sentiment is an important piece of information. For detecting sentiment words, we focused on the analysis of online users' comments, and developed a method to automatically detect words, or pairs of words, that express a sentiment. To compute the reputation of entities we analysed how sentiment words are used to influence what is said about a given entity. Finally, we investigate recommendation models having in mind the findings given by our sentiment lexicon and entities reputation. Hence, this thesis makes the following key research contributions:

1. Provides a method that identifies domain specific sentiment words
2. Investigate different types of data (Amazon, IMDb and TripAdvisor)
3. Builds a graph to infer entities reputation depicting,
   a. Sentiment relations between entities and sentiment words
   b. A better understanding on the sentiment link between entities
4. Exploits sentiment relations to incorporate sentiment in a,
   a. Collaborative recommendation system, in particular proposes a method that resolves the problem of unknown users' ratings
   b. Content-based recommendation system that uses users comments about different entities to help improve recommendations about new products

One of main emphasis of this thesis was to use an automatic method to extract sentiment words and entities reputation. We would like to strengthen this contribution by exploring new methods for this part of the thesis. To improve our claims there are some aspects that we have identified which can be addressed in the future. For example, we noticed that our objectives do not always align with the available datasets, and consequently a better pre-processing of the data could improve the quality of our results. We notice that the Amazon dataset contains many reviews that do not target the product instead should have been filtered out (spam). In future we would like to address such issues by introducing a more insightful analysis of the data that is being used.

We also propose to improve the reputation algorithm by tackling the problem of polysemy – many entities have multiple meanings (i.e. entity *Batman* may refer to different *Batman* movies). The challenge is to correctly identify to which meaning the entity refers to. We have also identified that, in the entire thesis, the experiments focused solely on English language. However, we believe that the proposed algorithms can be valuable in areas such as Reputation Management and existing Recommendation Systems, and to this end we find it important to have in the future a coverage in different languages. Hence, we plan to use language independent tools to handle a larger spectrum of communities. Finally, we plan to use different types of content – i.e. blogs, forums, and even broaden our scope to capture the sentiment expressed in news articles and their impact in entities reputation. We believe that such algorithms will help gathering much more complete information, which can prove to be useful for journalists and data analysts.

Below, we revisit and provide answers to the research objectives we raised in chapter 1 (Section 1.2).

6.1 Research Objectives

In chapter 3 we investigated how to detect sentiment words and domain related idiosyncrasies where specific sentiment words are common. We aimed to detect the importance of such words in a sentiment classification and opinion ranking task. To this end we summarized the research questions into the following objective:

> *Objective 1: Apply probabilistic techniques to extract sentiment words from online reviews. Departing from the more traditional positive and negative representation, characterize sentiment words in terms of their sentiment distribution.*

As it was foreseeable, for a specific domain we found that domain related lexicons perform better than generic lexicons. We found that our model goes beyond traditional positive versus negative sentiment words since it does not characterize a word as positive versus negative, and instead creates a fine-grain model of how likely a word occurs at different sentiment levels which proved to be a critical feature to learn more elaborate sentiment word interactions and to improve opinion retrieval systems.

In the process of answering research questions for objective 1, we noticed that beyond generic sentiment words (i.e. *love* or *poor*) our sentiment lexicon (Rank-LDA) captures domain specific sentiment words that prove to be of high importance in sentiment analysis problems. For example, the adjective *stain*, and the nouns *oscar*, *michael* and *aishwarya*. Our experiments showed that as the number of users' reviews increase becomes more noticeable the importance of sentiment bearing entities. To this extent, we formulated the following research objective:

> *Objective 2: Predict the reputation of entities by investigating in a sentiment graph the sentiment words and entities sentiment relations and co-occurrence probability using propagation algorithms.*

We presented two evaluations to the proposed reputation model: one to evaluate the quality of the ranked sentiment lexicon in an entity reputation task and another to evaluate the quality of the obtained reputation values. Our results showed that a high percentage of the sentiment words captured by the ranked sentiment lexicon were relevant for an entity reputation task.

So far, we have proposed two algorithms, one that extracts words or pairs of words that are used to express a sentiment, and another that evaluates how such words influence entities reputation. We notice the possibility to use the aforementioned sentiment lexicon and reputation algorithms in the domain of recommendation algorithms and improve state-of-the-art work. Hence, we formulated our final research objective:

> *Objective 3: Investigate two recommendation system problems: first, techniques that embedded sentiment based ratings (Objective 1) in a recommendation system algorithm; and second, apply entities reputation analysis (Objective 2) in a recommendation system.*

Our experiments demonstrated that it is possible to improve a recommendation system algorithm with sentiment analysis algorithms. For this task we performed two experiments: first, an algorithm that analyses user' comments and users explicit ratings in a collaborative matrix which integrates the interactions of all users; second, a recommender system that aims to take into account trends and reputations across social media services. To this end, we have successfully improved a collaborative recommendation system with sentiment-based recommendations which were computed by using the sentiment analysis techniques discussed in Objective 1, and showed the potential of using the reputation analysis of named entities (Objective 2) in the problem of a recommendation system cold-start scenario.

To conclude, this dissertation three main contributions are (1) a fully generative method to learn domain specific lexicons from a domain corpus, (2) a reputation analysis approach to infer entities reputation and influence, and (3) a recommendation system that uses the algorithms proposed in (1) and (2). The first contribution consists of an automatic method to learn a space model for opinion retrieval and sentiment analysis classification. The proposed generative model learns sentiment word distributions by embedding multi-level relevance judgments in the estimation of the model parameters. In addition to words' sentiment distributions the model captures specific named entities that due to their popularity become a sentiment reference in their domain. The second contribution is a three-step reputation analysis framework: first, the method jointly extracts named entities reputation and a domain specific sentiment lexicon; second, an entities graph is created by analysing cross-citations in subjective sentences; and, third the graph structure results in a pairwise Markov Network where a propagation algorithm computes the reputation of each entity. Finally, we have successfully applied (1) and (2) in a collaborative recommendation system.

6.2 Demonstrators

I also demonstrated the applicability of the proposed research in two use cases: SentiMovie and PopMeter.

- **SentiMovie** illustrates the output of a sentiment-based recommendation system. In this visualization the quality of users' recommendations are improved in a matrix factorization with a new factor to regularize probabilistic sentiment ratings.
- **PopMeter** presents a sentiment graph that is designed to visualize and explore the sentiment of linked-entities. It uses a sentiment graph populated by named entities and sentiment words, and this visualization helps to identify the main entities and/or sentiment words that have an influence in a given entity reputation.

SentiMovie allows the user to navigate through positively and negatively recommended movies. For each movie the application allows the user to observe: (i) overall sentiment ratings predictions for the reviews targeting that movie; (ii) for a given user review a visual representation of the most positive and negative sentiment words and the respective sentiment rating prediction; and, (iii) for each movie are given two movie recommendations, one that the fans of that movie will be pleased and another that the fans will probably dislike. The inferred sentiment ratings obtained from user reviews are used in a recommendation algorithm and in chapter 5 we proved the concept that sentiment ratings can improve recommendation algorithms, which SentiMovie allows to visualize.

Bibliograph

Aciar, S. et al., 2007. Informed recommender: Basing recommendations on consumer product reviews. *Journal of Intelligent Systems, IEEE*, 22(3), pp.39–47.

Adomavicius, G. & Tuzhilin, A., 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), pp.734–749.

Aktolga, E. & Allan, J., 2013. Sentiment diversification with different biases. *Proceedings of the 36th International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp.593–602.

Albornoz, J.C., Chugur, I. & Amigó, E., 2012. Using an Emotion-based Model and Sentiment Analysis Techniques to Classify Polarity for Reputation. P. Forner, J. Karlgren, & C. Womser-Hacker, eds. *Conference and Labs of the Evaluation Forum, Online Working Notes (CLEF)*, 1178.

Amatriain, X. et al., 2009. The wisdom of the few: a collaborative filtering approach based on expert opinions from the web. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. pp. 532–539.

Amigó, E. et al., 2013. Overview of RepLab 2013: Evaluating Online Reputation Monitoring Systems P. Forner et al., eds. *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, 8138, pp.333–352.

Asur, S. & Huberman, B.A., 2010. Predicting the future with social media. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 1, pp.492–499.

Aue, A. & Gamon, M., 2005. Customizing sentiment classifiers to new domains: a case study. *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, 1, pp.207–218.

Baccianella, S., Esuli, A. & Sebastiani, F., 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC)*, 25, pp.2200–2204.

Belkin, N.J. & Croft, W.B., 1992. Information filtering and information retrieval: two sides of the same coin? *Commun. ACM*, 35(12), pp.29–38. Available at: http://doi.acm.org/10.1145/138859.138861.

Bespalov, D. et al., 2011. Sentiment Classification Based on Supervised Latent n-gram Analysis. *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*, pp.375–382.

Bethard, S. et al., 2004. Automatic Extraction of Opinion Propositions and their Holders. *Proceedings of the Association for the Advancement of Artificial Intelligence Spring Symposium on Exploring Attitude and Affect in Text (AAAI)*, pp.22–24.

Blei, D.M. & McAuliffe, J.D., 2007. Supervised Topic Models. *Advances in Neural Information Processing Systems (NIPS), Curran Associates, Inc.*, pp.121–128.

Blei, D.M., Ng, A.Y. & Jordan, M.I., 2003. Latent dirichlet allocation. *Journal of Machine Learning Research, JMLR.org*, 3, pp.993–1022.

Blum, A. et al., 2004. Semi-supervised Learning Using Randomized Mincuts. *Proceedings of the Twenty-first International Conference on Machine Learning*, p.13.

Blum, A. & Chawla, S., 2001. Learning from Labeled and Unlabeled Data Using Graph Mincuts. *Proceedings of the 8th International Conference on Machine Learning (ICML)*, pp.19–26.

Bross, J. & Ehrig, H., 2013. Automatic Construction of Domain and Aspect Specific Sentiment Lexicons for Customer Review Mining. *Proceedings of the 22Nd ACM International Conference on Conference on Information Knowledge Management (CIKM)*, pp.1077–1086.

Burke, R., 2002. Hybrid Recommender Systems: Survey and Experiments. *Journal of User Modeling and User-Adapted Interaction, Kluwer Academic Publishers*, 12(4), pp.331–370.

Burke, R., 2007. Hybrid Web Recommender Systems P. Brusilovsky, A. Kobsa, & W. Nejdl, eds. *The Adaptive Web, Springer Science and Business Media*, 4321, pp.377–408.

Calais Guerra, P.H. et al., 2011. From bias to opinion: a transfer-learning approach to real-time sentiment analysis. *Proceedings of the 17th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp.150–158.

Cambria, E., 2013. An Introduction to Concept-Level Sentiment Analysis. F. Castro, A. F. Gelbukh, & M. González, eds. *Advances in Soft Computing and Its Applications, 12th Mexican International Conference on Artificial Intelligence (MICAI)*, 8266, pp.478–483.

Chen, L. et al., 2012. Extracting Diverse Sentiment Expressions with Target-Dependent Polarity from Twitter. *In Proceedings of the 6th International Association for the Advancement of Artificial Intelligence Conference on Weblogs and Social Media (AAAI/ICWSM).*

Clausen, A., 2003. Online Reputation Systems: The Cost of Attack of PageRank. *Bachelors (with Honours), University of Melbourne*.

Cohen, W.W. & Singer, Y., 1999. Context-sensitive learning methods for text categorization. In

ACM Transactions on Information Systems (TOIS). pp. 141–173.

Das, S. & Chen, M., 2001. Yahoo! for Amazon: Sentiment parsing from small talk on the Web. *Journal Social Science Research Network (SSRN), Electronic Journal, EFA Barcelona Meetings*, pp.1375–1388.

Dave, K., Lawrence, S. & Pennock, D.M., 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. *Proceedings of the 12th International Conference on World Wide Web (WWW)*, pp.519–528.

Denecke, K., 2009. Are SentiWordNet scores suited for multi-domain sentiment classification? In *In Proceedings of ICDIM'2009*. IEEE, pp. 33–38.

Ding, X., Liu, B. & Yu, P.S., 2008. A holistic lexicon-based approach to opinion mining. *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM)*, pp.231–240.

Dumais, S.T. et al., 1988. Using latent semantic analysis to improve access to textual information. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp.281–285.

Esuli, A. & Sebastiani, F., 2005. Determining the semantic orientation of terms through gloss classification. *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM)*, pp.617–624.

Esuli, A. & Sebastiani, F., 2006. Sentiwordnet: A publicly available lexical resource for opinion mining L. R. and Evaluation, ed. *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC)*, 6, pp.417–422.

De Finetti, B., 1990. Theory of Probability. *John Wiley & Sons Ltd., Chichester*, 1-2.

Gerani, S., Carman, M.J. & Crestani, F., 2010. Proximity-based opinion retrieval. *Proceeding of the 33rd international ACM Conference on Research and development in Information Retrieval (SIGIR)*, pp.403–410.

Glance, N. et al., 2005. Deriving Marketing Intelligence from Online Discussion. *Proceedings of the 11th ACM International Conference on Knowledge Discovery in Data Mining (SIGKDD)*, pp.419–428.

Go, A., Bhayani, R. & Huang, L., 2009. Twitter Sentiment Classification using Distant Supervision. *CS224N Project Technical report, Stanford*, pp.1–12.

Guttman, R.H., 1998. Merchant differentiation through integrative negotiation in agent-mediated electronic commerce. *Doctoral dissertation, Massachusetts Institute of Technology, Program in Media Arts and Sciences*.

Harvey, M., Ruthven, I. & Carman, M., 2010. Ranking Social Bookmarks Using Topic Models. *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, pp.1401–1404.

Hatzivassiloglou, V. & McKeown, K.R., 1997. Predicting the semantic orientation of adjectives. *Proceedings of the 35th Annual Meeting on Association for Computational Linguistics (ACL)*, pp.174–181.

Hatzivassiloglou, V. & Wiebe, J.M., 2000. Effects of adjective orientation and gradability on sentence subjectivity. *Proceedings of the 18th Conference on Computational Linguistics (COLING)*, 1, pp.299–305.

Heerschop, B. et al., 2011. Polarity analysis of texts using discourse structure. *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*, pp.1061–1070.

Hofmann, T., 1999. Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM conference on Research and development in information retrieval (SIGIR)*, pp.50–57.

Hu, M. & Liu, B., 2004a. Mining and summarizing customer reviews. *Proceedings of the tenth ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp.168–177.

Hu, M. & Liu, B., 2004b. Mining opinion features in customer reviews. *Proceedings of the Association for the Advancement of Artificial Intelligence 19th International Conference on Artifical Intelligence (AAAI)*, pp.755–760.

Hu, X. et al., 2013. Exploiting social relations for sentiment analysis in microblogging. *Proceedings of the 6th ACM International conference on Web Search and Data Mining (WSDM)*, pp.537–546.

Hu, Y., Koren, Y. & Volinsky, C., 2008. Collaborative Filtering for Implicit Feedback Datasets. *Proceedings of the Eighth IEEE International Conference on Data Mining*, pp.263–272.

Jakob, N. et al., 2009. Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion (CIKM/TSA)*, pp.57–64.

Jansen, B.J. et al., 2009. Twitter Power: Tweets As Electronic Word of Mouth. *Journal of the American Society for Information Science and Technology, John Wiley and Sons, Inc.*, 60(11), pp.2169–2188.

Jiang, L. et al., 2011. Target-dependent Twitter sentiment classification. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics and Human Language Technologies (ACL/HLT)*, 1, pp.151–160.

Jindal, N. & Liu, B., 2006. Mining Comparative Sentences and Relations. *Proceedings of the 21st National Conference on Artificial Intelligence*, 2, pp.1331–1336.

Jindal, N. & Liu, B., 2008. Opinion spam and analysis. In *Proceedings of the international conference on Web search and web data mining (WSDM)*. ACM, pp. 219–230.

Jo, Y. & Oh, A.H., 2011. Aspect and sentiment unification model for online review analysis. *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*, pp.815–824.

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98*. pp. 137–142.

Joshi, M. et al., 2010. Movie Reviews and Revenues: An Experiment in Text Regression. *Proceddings of the Annual Conference of the North American Chapter of the Association of Computational Linguistics (NAACL/HLT)*, pp.293–296.

Kamps, J. et al., 2004. Using wordnet to measure semantic orientation of adjectives. *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC)*, pp.1115–1118.

Kang, J.-H., Lerman, K. & Getoor, L., 2013. LA-LDA: A Limited Attention Topic Model for Social Recommendation. *Proceedings of the 6th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, pp.211–220.

Kim, S.-M. & Hovy, E., 2004. Determining the sentiment of opinions. *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, 4, pp.1367–1373.

Kleinberg, J.M., 1998. Authoritative Sources in a Hyperlinked Environment. *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.668–677.

Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. *Proceeding of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp.426–434.

Koren, Y., Bell, R. & Volinsky, C., 2009. Matrix Factorization Techniques for Recommender Systems. *Journal of Computer, IEEE Computer Society Press Los Alamitos*, 42(8), pp.30–37.

Krauss, J. et al., 2008. Predicting Movie Success and Academy Awards through Sentiment and Social Network Analysis. *The 16th European Conference on Information Systems (ECIS)*, pp.2026–2037.

Kwak, H. et al., 2010. What is Twitter, a Social Network or a News Media? *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pp.591–600.

Landauer, T.K., Foltz, P.W. & Laham, D., 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3), pp.259–284.

Lee, S. et al., 2011. Random Walk Based Entity Ranking on Graph for Multidimensional Recommendation. *Proceedings of the Fifth ACM Conference on Recommender Systems*, pp.93–100.

Leung, C.W.K., Chan, S.C.F. & Chung, F., 2006. Integrating Collaborative Filtering and Sentiment Analysis: A Rating Inference Approach. *Proceedings of the European Conference on Artificial Intelligence, Workshop on Recommender Systems (ECAI)*, pp.62–66.

Li, C. et al., 2012. TwiNER: Named Entity Recognition in Targeted Twitter Stream. *Proceedings of the 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp.721–730.

Lim, K.W. & Buntine, W., 2014. Twitter Opinion Topic Model: Extracting Product Opinions from Tweets by Leveraging Hashtags and Sentiment Lexicon. *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, pp.1319–1328.

Lin, C. & He, Y., 2009. Joint Sentiment/Topic Model for Sentiment Analysis. *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pp.375–384.

Liu, B., 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies, Morgan and Claypool Publishers*, pp.1–167.

Liu, B., 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing, CRC Press, Taylor and Francis Group*.

Liu, B., Hsu, W. & Ma, Y., 1998. Integrating classification and association rule mining. *Proceedings of the 4th International Conference on Knowledge Discovery and Data mining (KDD)*.

Lourenco Jr., R. et al., 2014. Economically-efficient Sentiment Stream Analysis. *Proceedings of the 37th International ACM Conference on Research Development in Information Retrieval (SIGIR)*, pp.637–646.

Lu, B., 2010. Identifying Opinion Holders and Targets with Dependency Parser in Chinese News Texts. *Proceedings of the NAACL HLT 2010 Student Research Workshop*, pp.46–51.

Martineau, J. & Finin, T., 2009. Delta TFIDF: An Improved Feature Space for Sentiment Analysis. *Proceedings of the 3th Association for the Advancement of Artificial Intelligence Internatonal Conference on Weblogs and Social Media (AAAI/ICWSM)*, pp.258–261.

Martín-Wanton, T. et al., 2012. UNED at RepLab 2012: Monitoring Task. In *Conference and Labs of the Evaluation Forum, Online Working Notes (CLEF)*.

Martín-Wanton, T., Gonzalo, J. & Amigó, E., 2013. An unsupervised transfer learning approach to discover topics for online reputation management. *Proceedings of the 22nd ACM International Conference on Information Knowledge Management (CIKM)*, pp.1565–1568.

Moghaddam, S. & Ester, M., 2011. ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. *Proceedings of the 34th international ACM conference on Research and development in Information Retrieval (SIGIR)*, pp.665–674.

Moghaddam, S. & Ester, M., 2012. On the Design of LDA Models for Aspect-based Opinion Mining. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, pp.803–812.

Moshfeghi, Y., Piwowarski, B. & Jose, J.M., 2011. Handling data sparsity in collaborative filtering using emotion and semantic based features. *Proceedings of the 34th international ACM conference on Research and development in Information Retrieval (SIGIR)*, pp.625–634.

Mui, L., Mohtashemi, M. & Halberstadt, A., 2002. A Computational Model of Trust and Reputation. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS)*, 7, pp.2431–2439.

Mullen, T. & Collier, N., 2004. Sentiment analysis using support vector machines with diverse information sources. In *In Proceedings of Conference on Empirical Methods in Natural Language Processing*. pp. 412–418.

Oghina, A. et al., 2012. Predicting IMDB Movie Ratings Using Social Media. *Proceedings of the 34th European Conference on Advances in Information Retrieval (ECIR)*, pp.503–507.

Ohana, B. & Tierney, B., 2009. Sentiment classification of reviews using SentiWordNet. In *9th. IT & T Conference*. p. 13.

Page, L. et al., 1998. The PageRank citation ranking: Bringing order to the Web. *Proceedings of the 7th International World Wide Web Conference (WWW)*, pp.161–172.

Pang, B. & Lee, L., 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *Proceedings of the Association of Computational Linguistics (ACL)*, pp.271–278.

Pang, B. & Lee, L., 2008. Opinion mining and sentiment analysis. *Journal of Foundations and Trends in Information Retrieval, Now Publishers Inc.*, 2(1-2), pp.1–135.

Pang, B. & Lee, L., 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 43(1), pp.115–124.

Pang, B., Lee, L. & Vaithyanathan, S., 2002. Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 10, pp.79–86.

Pazzani, M.J., 1999. A Framework for Collaborative, Content-Based and Demographic Filtering. *Journal of Artificial Intelligence Review, Kluwer Academic Publishers Norwell*, 13(5-6), pp.393–408.

Peleja, F. et al., 2013. A recommender system for the TV on the web: integrating unrated reviews and movie ratings. *Journal of Multimedia Systems, Springer-Verlag New York*, 19(6), pp.543–558.

Peleja, F., 2015. PopMeter: Linked-Entities in a Sentiment Graph N. Hanbury, Allan and Kazai, Gabriella and Rauber, Andreas and Fuhr, ed. *Proceedings of the 37th European Conference on Advances in Information Retrieval (ECIR)*, pp.785–788.

Peleja, F., Dias, P. & Magalhães, J., 2012. A Regularized Recommendation Algorithm with Probabilistic Sentiment-Ratings. *Proceedings on the IEEE 12th International Conference on Data Mining Workshops (ICDMW/SENTIRE)*, pp.701–708.

Peleja, F. & Magalhães, J., 2015. Learning Sentiment Based Ranked-Lexicons for Opinion Retrieval N. Hanbury, Allan and Kazai, Gabriella and Rauber, Andreas and Fuhr, ed. *Proceedings of the 37th European Conference on Advances in Information Retrieval (ECIR)*, pp.435–440.

Peleja, F. & Magalhães, J., 2013. Opinions in User Reviews: An Evaluation of Sentiment Analysis Techniques. *Local Proceedings of the 16th Portuguese Conference on Artificial Intelligence (EPIA)*, pp.468–479.

Peleja, F., Santos, J. & Magalhães, J., 2014a. Ranking Linked-Entities in a Sentiment Graph. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pp.118–125.

Peleja, F., Santos, J. & Magalhães, J., 2014b. Reputation Analysis with a Ranked Sentiment-lexicon. *Proceedings of the 37th International ACM Conference on Research Development in Information Retrieval (SIGIR)*, pp.1207–1210.

Peng, W. & Park, D.H., 2011. Generate Adjective Sentiment Dictionary for Social Media Sentiment Analysis Using Constrained Nonnegative Matrix Factorization. In *Proceedings of the 6th International Association for the Advancement of Artificial Intelligence Conference on Weblogs and Social Media (AAAI/ICWSM)*.

Petasis, G. et al., 2014. Sentiment Analysis for Reputation Management: Mining the Greek Web. *Artificial Intelligence: Methods and Applications*, 8445, pp.327–340.

Popescu, A.-M. & Etzioni, O., 2005. Extracting product features and opinions from reviews. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pp.339–346.

Prabowo, R. & Thelwall, M., 2009. Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2), pp.143–157.

Qiu, G. et al., 2009. Expanding domain sentiment lexicon through double propagation. *Proceedings of the 21st international jont conference on Artifical intelligence*, pp.1199–1204.

Qu, L., Ifrim, G. & Weikum, G., 2010. The bag-of-opinions method for review rating prediction from sparse text patterns. *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pp.913–921.

Quirk, R. et al., 1985. A Comprehensive Grammar of the English Language. *General Grammar Series, Longman*.

Ramage, D. et al., 2009. Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1, pp.248–256.

Rao, D. & Ravichandran, D., 2009. Semi-supervised Polarity Lexicon Induction. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (COLING)*, pp.675–682.

Resnick, P. et al., 1994. GroupLens: an open architecture for collaborative filtering of netnews. *Proceedings of the ACM conference on Computer supported cooperative work*, pp.175–186.

Resnick, P. et al., 2000. Reputation Systems. *Magazine Communications of the ACM, ACM New York*, 43(12), pp.45–48.

Resnick, P. & Varian, H.R., 1997. Recommender Systems. *Magazine Communications of the ACM, ACM New York*, 40(3), pp.56–58.

Rietjens, B., 2006. Trust and reputation on eBay: Towards a legal framework for feedback intermediaries. *Journal of Information and Communications Technology Law, Taylor and Francis Group*, 15(2), pp.55–78.

Riloff, E., Patwardhan, S. & Wiebe, J., 2006. Feature subsumption for opinion analysis. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.440–448.

Riloff, E. & Wiebe, J., 2003. Learning extraction patterns for subjective expressions. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.105–112.

Sabater, J. & Sierra, C., 2001. REGRET: Reputation in Gregarious Societies. *Proceedings of the 5th International Conference on Autonomous Agents*, pp.194–195.

Salton, G. & McGill, M.J., 1986. Introduction to modern information retrieval.

Salton, G., Wong, A. & Yang, C.-S., 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11), pp.613–620.

Schafer, J. Ben et al., 2007. Methods and Strategies of Web Personalization P. Brusilovsky, A. Kobsa, & W. Nejdl, eds. *The Adaptive Web, Springer Berlin Heidelberg*, 4321, pp.291–324.

Scheible, C. & Schütze, H., 2012. Bootstrapping Sentiment Labels For Unannotated Documents With Polarity PageRank. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pp.1230–1234.

Snow, R. et al., 2008. Cheap and fast---but is it good?: evaluating non-expert annotations for natural language tasks. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.254–263.

Song, Y. et al., 2009. Topic and keyword re-ranking for LDA-based topic modeling. *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pp.1757–1760.

Sparling, E.I., 2011. Rating : How Difficult is It ? In *RecSys '11 Proceedings of the fifth ACM conference on Recommender systems*. ACM Press, pp. 149–156. Available at: http://dl.acm.org/citation.cfm?id=2043932.2043961.

Spina, D. et al., 2013. UNED Online Reputation Monitoring Team at RepLab 2013. In *Conference and Labs of the Evaluation Forum, Online Working Notes (CLEF)*.

Spina, D., Gonzalo, J. & Amigó, E., 2014. Learning Similarity Functions for Topic Detection in Online Reputation Monitoring. *Proceedings of the 37th International ACM Conference on Research Development in Information Retrieval (SIGIR)*, pp.527–536.

Standifird, S.S., 2001. Reputation and e-commerce: eBay auctions and the asymmetrical impact of positive and negative ratings. *Journal of Management, Southern Management Association*, 27(3), pp.279–295.

Stevens, K. et al., 2012. Exploring Topic Coherence over Many Models and Many Topics. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*, pp.952–961.

Stone, P.J. et al., 1966. The General Inquirer: A Computer Approach to Content Analysis. *MIT Press, Cambridge, MA*.

Stone, P.J. & Hunt, E.B., 1963. A Computer Approach to Content Analysis: Studies Using the General Inquirer System. *Proceedings of the Spring Joint Computer Conference*, pp.241–256.

Takama, Y. & Muto, Y., 2007. Profile Generation from TV Watching Behavior Using Sentiment Analysis. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops (WI-IAT)*, pp.191–194.

Takamura, H., Inui, T. & Okumura, M., 2005. Extracting Semantic Orientations of Words using Spin Model. *Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 47, pp.133–140.

Tan, C. et al., 2011. User-level Sentiment Analysis Incorporating Social Networks. *Proceedings of the 17th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp.1397–1405.

Tang, H., Tan, S. & Cheng, X., 2009. A survey on sentiment detection of reviews. *Journal of Expert Systems with Applications, Pergamon Press*, 36(7), pp.10760–10773.

Taskar, B., Abbeel, P. & Koller, D., 2002. Discriminative Probabilistic Models for Relational Data.

Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, pp.485–492.

Titov, I. & McDonald, R., 2008. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. *Proceedings of Annual Meeting of the Association for Computational Linguistics and Human Language Technology Conference (ACL/HLT)*, pp.308–316.

Turney, P., 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pp.417–424.

Turney, P. & Littman, M., 2002. Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus. *Tecnical report ERC-1094, National Research Council of Canada*, (ERB-1094), p.11.

Turney, P.D., 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL L. De Raedt & P. Flach, eds. *Proceedings of the 12th European Conference on Machine Learning (EMCL)*, 2167, pp.491–502.

Turney, P.D. & Littman, M.L., 2003. Measuring praise and criticism: Inference of semantic orientation from association. *Journal of ACM Transactions on Information Systems (TOIS), ACM New York*, 21(4), p.37.

Valitutti, R., 2004. WordNet-Affect: an Affective Extension of WordNet. *In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pp.1083–1086.

Velikovich, L. et al., 2010. The Viability of Web-derived Polarity Lexicons. *Proceddings of the Annual Conference of the North American Chapter of the Association of Computational Linguistics (NAACL/HLT)*, pp.777–785.

Villena-Román, J. et al., 2012. DAEDALUS at RepLab 2012: Polarity Classification and Filtering on Twitter Data. P. Forner, J. Karlgren, & C. Womser-Hacker, eds. *Conference and Labs of the Evaluation Forum, Online Working Notes (CLEF)*.

Wang, H., Lu, Y. & Zhai, C., 2010. Latent aspect rating analysis on review text data. *Proceedings of the 16th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp.783–792.

Wang, X. et al., 2011. Topic Sentiment Analysis in Twitter: A Graph-based Hashtag Sentiment Classification Approach. *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*, pp.1031–1040.

Wang, Y., Liu, Y. & Yu, X., 2012. Collaborative Filtering with Aspect-Based Opinion Mining: A Tensor Factorization Approach. *Proceedings on the IEEE 12th International Conference on Data Mining (ICDM)*, pp.1152–1157.

Weichselbraun, A., Gindl, S. & Scharl, A., 2013. Extracting and Grounding Contextualized Sentiment Lexicons. *Journal of Intelligent Systems, IEEE*, 28(2), pp.39–46.

Wiebe, J., Bruce, R. & O'Hara, T., 1999. Development and use of a gold-standard data set for subjectivity classifications. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (ACL)*, pp.246–253.

Wiebe, J. & Cardie, C., 2005. Annotating expressions of opinions and emotions in language. Language Resources and Evaluation. *Proceddings of the International Conference on Linguistic Resources and Evaluation (LREC)*, 2(39), pp.165–210.

Wiebe, J. & Riloff, E., 2005. Creating subjective and objective sentence classifiers from unannotated texts. *Proceedimgs International Conference Computational Linguistics and Intelligent Text Processing (CICLing)*, pp.486–497.

Wiebe, J.M., 1990. Identifying subjective characters in narrative. *Proceedings of the 13th Conference on Computational Linguistics (COLING)*, 2, pp.401–406.

Wiebe, J.M., 1994. Tracking point of view in narrative. *Journal of Computational Linguistics, MIT Press Cambridge*, 20(2), pp.233–287.

Wilson, T. & Wiebe, J., 2005. Annotating Attributions and Private States. *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky (CorpusAnno)*, pp.53–60.

Wilson, T., Wiebe, J. & Hoffmann, P., 2005. Recognizing contextual polarity in phrase-level sentiment analysis. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (EMNLP)*, pp.347–354.

Wilson, T., Wiebe, J. & Hwa, R., 2004. Just How Mad Are You? Finding Strong and Weak Opinion Clauses. *Proceedings of the 19th Association for the Advancement of Artificial Intelligence Conference on Artifical Intelligence (AAAI)*, pp.761–767.

Yang, B. & Cardie, C., 2012. Extracting opinion expressions with semi-Markov conditional random fields. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*, pp.1335–1345.

Yu, H. & Hatzivassiloglou, V., 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences M. Collins & M. Steedman, eds. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 10(3), pp.129–136.

Yuan, G., Ho, C. & Lin, C., 2011. Recent Advances of Large-scale Linear Classification. *Computer*, (3), pp.1–15. Available at: http://www.csie.ntu.edu.tw/~cjlin/papers/survey-linear.pdf.

Zhang, H. et al., 2007. An LDA-based community structure discovery approach for large-scale social networks. *Journal of Intelligence and Security Informatics, IEEE*, pp.200–207.

Zhang, L. et al., 2010. Extracting and Ranking Product Features in Opinion Documents. *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pp.1462–1470.

Zhang, L. & Liu, B., 2011. Extracting Resource Terms for Sentiment Analysis. *Proceddings of the International Joint Conference on Natural Language Processing (IJCNLP)*, pp.1171–1179.

Zhang, M. & Ye, X., 2008. A generation model to unify topic relevance and lexicon-based sentiment for opinion retrieval. *Proceedings of the 31st annual international ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp.411–418.

Zhang, W. et al., 2010. Augmenting Online Video Recommendations by Fusing Review Sentiment

Classification. *Proceedimgs on IEEE International Conference on Data Mining Workshops (ICDMW)*, pp.1143–1150.

Zhao, W.X. et al., 2010. Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.56–65.

Zhu, X. & Ghahramani, Z., 2002. Learning from Labeled and Unlabeled Data with Label Propagation. *Technical Report CMU-CALD-02-107, Carnegie Mellon University*.