UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Cross Domain Multi-View Sentiment Classification

Dinko Lambov

Tese para a obtenção do Grau de Doutor em
Engenharia Informática

Orientador: Prof. Doutor Gaël Harry Dias

Covilhã, Dezembro de 2010

Acknowledgements

I wish to thank my family for the encouragement and patience during this work, and most of all my girlfriend for her loving support. In terms of content and quality of this work, the biggest thanks must go to my supervisor, Dr. Gaël Dias. He has given his time and talent in ways that simply cannot be acknowledged. He guided me in this endeavor with just the right mixture of affirmation and criticism in saying just the right thing at the right time. I also thank Professor Veska Nocheva, Professor Guillaume Cleuziou, João V. Graça and Sebastião Pais for their invaluable help in this work. Finally, I owe a lot to my colleagues and my friends, all the people of Hultig.

The work on this dissertation was supported by Programa Operacional Potencial Humano of QREN Portugal, by grant SFRH/BD/31532/2006 of Portuguese Agency for Research (Fundação para a Ciência e a Tecnologia)

Abstract

With the explosion of user-generated web content in the form of blogs, wikis and discussion forums, the Internet has rapidly become a massive dynamic repository of public opinion on an unbounded range of topics. A key enabler of opinion extraction and summarization is sentiment classification: the task of automatically identifying whether a given piece of text expresses an opinion or a fact. Building high-quality sentiment classifiers using standard text categorization methods is challenging due to the lack of labeled data in a target domain. In this thesis, we consider the problem of cross-domain sentiment analysis: can one, for instance, download rated movie reviews from rottentomatoes.com or IMBD discussion forums, learn linguistic expressions and opinion-bearing features that generally characterize opinionated reviews and then successfully transfer this knowledge to the target domain, thereby building high-quality sentiment models without manual effort? In order to solve such a problem, we present and evaluate a new method based on multi-view learning using both high-level (e.g. level of affective words, level of abstraction of nouns) and low-level features (e.g. unigrams, bigrams). In particular, we show that multi-view learning combining high-level and low-level features with adapted classifiers can lead to improved results over text subjectivity classification. The second task we consider is to automatically produce learning data from Web resources. To do so, we propose to compare Wikipedia and Web Blogs texts to reference objective and subjective corpora and show that Wikipedia texts are representative of objectivity and Web Blogs are representative of subjectivity. As a consequence, learning data will be easy to build for many languages without manual annotation alleviating intensive and time-consuming labor.

Keywords

Sentiment analysis, subjectivity classification, natural language processing, machine learning, co-training, multi-view learning.

Resumo

Com a explosão dos conteúdos da web gerados por utilizadores na forma de blogs, wikis e fóruns de discussão, a Internet tornou-se rapidamente um enorme repositório dinâmico da opinião pública sobre uma gama ilimitada de temas. Um factor essencial de extração de opinião e de sumarização é a classificação de sentimentos: a tarefa de identificar automaticamente se uma determinada parte de um texto expressa uma opinião ou um facto. A construção de classificadores de sentimentos de alta qualidade, utilizando métodos de categorização de texto, é um grande desafio devido á falta de dados etiquetados para um determinado domínio. Nesta tese, consideramos o problema da análise de sentimentos trans-domínios. A questão é a seguinte. Será possível, por exemplo, baixar resenhas de filmes classificadas do rottentomatoes.com ou fóruns de discussão tais como o IMBD, aprender expressões linguísticas e características de opiniões que geralmente caracterizam opiniões e depois com sucesso transferir esse conhecimento para um domínio de destino, construindo assim modelos sobre sentimentos de alta qualidade sem esforço manual? Para o efeito, apresentam-se e avaliam-se novos métodos baseados na aprendizagem multi-vistas utilizando características tanto de alto nível (por exemplo, nível de expressão afectiva, nível de abstração dos substantivos) como de baixo nível (por exemplo uni-gramas ou bi-gramas). Em particular, mostramos que a aprendizagem multi-vistas combinando recursos de alto nível e baixo nível com classificadores adaptados pode levar a melhores resultados sobre a classificação da subjectividade textual. A segunda tarefa que consideramos é a produção automática de dados de aprendizagem a partir de recursos da web. Para isso, propomos comparar textos da Wikipedia e Weblogs com corpora de referência e mostramos que os textos da Wikipedia são representativos da objectividade e os Weblogs são representativos da subjectividade. Como consequência, será mais fácil construir classifcadores para várias línguas, sem anotação manual e assim aliviar uma tarefa trabalhosa e demorada.

Palavras-chave

Análise de sentimentos, Classificação da subjectividade, Processamento Automático

da Linguagem Natural, Aprendizagem, Co-training, Aprendizagem multi-vistas.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> "*Every man has a right to be wrong in his opinions. But no man has a right to be wrong in his facts.*"
>
> **Bernard M. Baruch**

1.1 Main Objectives

The rapid growth of everyday use of the World Wide Web has changed both the behavior of Internet users and their information needs. A significant number of Web applications allow users to publish and share their experiences and opinions. For example, websites such as Amazon.com[1] and review aggregators such as Yelp.com[2] collect customer reviews on specific products or services, while blogs (short for Weblogs) and services such as Twitter[3] and Facebook[4] allow users to publish their opinions on an infinite array of topics, ranging from the benefits of iPhone to the presidential election. The opinions of lay customers (non-specialist) may serve to balance and complement the authoritative points of view published by online media such as the New York Times newspaper. Therefore, these opinions have attracted an increasing amount of interest from individuals as well as organizations. For example, people are curious about what other people think of certain products or topics, companies want to find out what their target audience likes or dislikes about their products and services, and government officials would like to learn whether people are for or against new policies. In order to facilitate access to this enormous body of user-generated contents, some search engines have expanded their indexing scopes to include blogs, Facebook profiles, Web discussion forums, etc. In fact, gathering reliable information is becoming more and more difficult as more subjective information is produced every day. Within the context of our research, we are especially interested in proposing objective information corresponding to the users' needs, unlike most other proposals, which focus on the evaluation of the public opinion. Indeed, learning subjectivity in text is an exciting research field, also called opinion mining, which offers enormous opportunities for various applications.

[1] http://www.Amazon.com [16th *November*, 2010].
[2] http://www.Yelp.com [16th *November*, 2010].
[3] http://www.twitter.com [16th *November*, 2010].
[4] http://www.facebook.com [16th *November*, 2010].

In particular, it is likely to provide powerful functionalities for competitive analysis and marketing analysis through topic tracking and detection of unfavorable rumors. Although this issue is not to be put apart, it does not fulfill our principal goal. From our point of view, the user may have access to subjective contents but, in this case, he must be alerted that some of the contents he may read can be subjective. Indeed, it is important to guarantee the quality of information when it is ruling most of our opinion.

1.2 Problems of Current Approaches

Most works up-to-date tackle the simpler task of detecting polarity i.e. if a document is treating a given topic in a positive or negative way. Indeed, polarity is an important issue for topic tracking or opinion mining as companies or politicians search for positive recognition from the public opinion. However, our concern is different as we aim at offering reliable information to users rather than focusing on rumors, gossips or malicious contents. Of course, our discourse is purposely tendentious as negative and positive contents can be helpful for many tasks. One of the main difficulties in learning subjectivity in text is that the border between subjectivity and objectivity is fuzzy. Indeed, by definition, a negative or a positive statement is necessarily subjective. Only facts can be objective. In fact, this definition leaves very little space for objective contents. From our point of view, polarity can only express subjectivity if not supported by facts. For example, public opinion may be subjective as it is usually based on misinformation, rumors or even manipulation. In this case, a negative statement can be subjective. So, saying that everything, which is negative is definitely subjective does not represent our point of view and negative and positive statements can be objective. Unfortunately, the definitions of the notions of subjectivity or sentiments in texts have received little attention, perhaps with the exception of Boiy *et al.* (2007). Therefore, we base our research on the most general paradigm, which aims at classifying texts as being either objective or subjective, and let polarity be judged at a second stage of the process of opinion/sentiment learning. It is important to note that fewer works have been proposed within this scope. Another particularity of the studies proposed so far about sentiment analysis is the fact that most of them mainly propose in-domain supervised classifiers. However, within the context of real-world environments, sentiment is expressed differently in different domains and sentiment analysis must deal with this problem. A few researches dealt with cross-domain subjectivity classification and all argued that it is hard to learn a domain-independent classifier. One possible approach is to train the classifier on a domain-mixed set of data instead of training it on one specific domain (Aue & Gamon (2005)). Another possibility is to propose high-level features, which do not depend so much on topics such as part-of-speech statistics (Finn & Kushmerick (2006)). Blitzer *et al.* (2007) proposed another domain adaptation method using pivot features based on domain mu-

tual information to relate training and target domains. Then, the overall approach extended the structural correspondence learning algorithm (SCL) to polarity classification. As a consequence, they identified a measure of domain similarity that correlated well with the potential for adaptation of a classifier from one domain to another.

1.3 Our Proposal

To solve the problems described in the previous section, this thesis introduces a new approach based on multi-view and semi-supervised learning algorithms, using high-level and low-level features to learn subjective language across domains. In particular, we apply different algorithms based on multi-view and semi-supervised learning to obtain maximum performance over two and three views (high-level and low-level features). Experimental results show that our approach outperforms the stochastic agreement regularization (SAR) algorithm, which is the reference algorithm in the domain (Ganchev *et al.* (2008)). In the proposed methods, we use different views to guide the selection of training candidates and to minimize the risk of adding misclassified examples. Within this context, we propose to use more than two views and customize the SAR algorithm to receive a different number of views. Another important contribution (in order to reach language-independency) is the automatic construction of labeled data sets. Indeed, supervised classification techniques require large amounts of labeled training data. However, the acquisition of these data can be time-consuming and expensive. Based on this assumption, we propose to automatically produce learning data from Web resources. In particular, we compare Wikipedia and Weblogs texts to reference objective and subjective corpora and conclude that Wikipedia texts convey objective messages while Weblogs present subjective contents. We also explore a few important opinion-bearing features and we propose a new one, based on the level of abstraction of nouns, which improves accuracy across domains using both support vector machines (SVM) and linear discriminant analysis (LDA) classifiers. In order to evaluate to what extent the given features are discriminative and allow representing distinctively the datasets in the given space of characteristics we propose to do feature selection by applying the Wilcoxon rank-sum test and visualize the datasets using multidimensional scaling.

1.4 Main Contributions

Our work has been evaluated and assessed by different papers presented at international conferences and published in the proceedings thereof. A brief synthesis of the set of publications is given here together with a brief description and explanation of each item.

Lambov *et al*. (2009a) - In this paper, we presented a methodology, which aims at classifying texts at the subjectivity level (i.e. subjective vs. objective) based on high-level semantic features, which can apply to different domains. In particular, we proposed a new feature based on the level of abstraction of nouns, which improved accuracy across domains using both support vector machines (SVM) and linear discriminant analysis (LDA) classifiers.

Lambov *et al*. (2009b) - In this paper, we evaluated the extent to which the given high-level features are discriminative and allow distinctive presentation of the datasets in the given space of high-level characteristics. Therefore, we proposed feature selection by applying the Wilcoxon rank-sum test and visualized the datasets using multidimensional scaling.

Lambov *et al*. (2010) - In this paper, we proposed to combine high-level features (e.g. level of affective words, level of abstraction of nouns) and low-level features (e.g. unigrams and bigram) to learn models of subjectivity, which may apply to different domains. Therefore, we proposed a new scheme based on the classical co-training algorithm (Blum & Mitchell (1998)) over two views and joined two different classifiers LDA and SVM to optimize the approach.

Lambov *et al*. (2011) - In this paper, we propose to compare the SAR algorithm (Ganchev *et al*. (2008)) with the multi-view learning strategy constrained by agreement. Within this overall context, we propose to take into account agreement confidence in the learning process. The basic idea is that at each iteration of the learning process new examples are added only if both classifiers agree on predicted labels. Within this context, we propose three multi-view learning algorithms, which present the best results overall with accuracy levels across domains of 80% compared to 77.1% for the SAR algorithm.

As an extension of these works, we present novel approaches based on multi-view learning, which use the best classifier to guide the selection of the data that are added to the training set in the self-training process. Based on this hypothesis, we propose to use different methodologies to combine high-level and low-level features. We also propose to use more than two views by dividing high-level features into different sets. Within this context, the SAR (Ganchev *et al*. (2008)) algorithm is only defined for two views. So, we mathematically define its formalism for 3 views and experiments will soon be carried out to verify its behavior. This work is being carried out in collaboration with Guillaume Cleuziou and Lionel Martin from the University of Orleans (France).

1.5 Plan of the Thesis

The rest of the thesis is organized as follows: Chapter 2 defines the problem of sentiment analysis, lists major challenges and refers to some work done in this area describing some already developed and used methodologies. Chapter 3 contains a review of related works. Chapter 4 presents some existing resources and a corpus based on Web resources and automatically annotated. Chapter 5 describes in details the opinion-bearing features we used in our experiments. The developed methods are described in Chapter 6. Chapter 7 discusses the evaluation results and finally chapter 8 concludes the thesis and outlines future research perspectives.

Chapter 2

Challenges in Sentiment Analysis

> "*Opinion is that exercise of the human will which helps us to make a decision without information.*"
>
> **John Erskine**

The challenges of opinion detection lie in the subtle nature of opinions, which makes them more complex and heavily dependent on the context in which they are used. Moreover, the noise that is characteristic of the World Wide Web adds an extra challenge for opinion detection in Web documents. It is well-known that people express their feelings and opinions in indirect ways. Word-based models succeed to a surprising extent, but are insufficient in predictable ways when attempting to measure favorability toward entities. Pragmatic considerations, sarcasm, comparisons, rhetorical reversals ("I was expecting to love it"), and other rhetorical structures tend to undermine much of the direct relationship between the words used and the opinion expressed. Any task, which seeks to extract human opinions and feelings from texts will have to deal with these challenges (Mullen & Malouf (2006)). Many applications try to understand sentiment using keywords or clusters of keywords. One of the main difficulties here, is dealing with the considerable problem of frequent spelling errors and slang words in informal texts. This problem is compounded when the work is in a domain where people express their opinions using jargon and other non-dictionary words like in Weblogs (Boughanem *et al.* (2009), Mullen & Malouf (2006)). The difficulties of analysis at the word level percolate to the level of part-of-speech tagging and upwards, making any linguistic analysis challenging. Second, beyond the issues of ambiguity, being able to pull out the tone and meaning in a statement or set of statements is hard because people express things in different ways. As such, finding the sentiment in a sentence is hard when using statistical approaches. For example, let's take the statement: "The hotel room is on the ground floor right by the reception". Is it neutral, positive or negative? Well, different people would probably give different answers. If you ask for a high-floor room with a view away from the noise or reception, the review is clearly negative. If you have mobility issues and ask for a room with easy access, it is positive. And for many people it would just be information and so neutral. The fuzzy boundaries of opinion also exist for sentiment classification. A common phenomenon known as “thwarted expectations” can

be found in movie reviews, where the author of a movie review sets up a “deliberate contrast to earlier discussion” (Pang *et al.* (2002)). For example, one might first say that a movie is shallow but then go on to say that he enjoyed it nevertheless. This fuzziness of the sentiment boundary may be easy for humans to detect, but it can be challenging for machine learning approaches. Since there is no universal agreement on drawing the boundaries of opinions, it is up to researchers to decide whether or not to include ambiguous opinions in their training sets. Some researchers only use sentences with strong traces of subjectivity in training data in order to reduce fuzziness (Pang *et al.* (2002); Wiebe *et al.* (1999)). This makes the task easier and likely to yield better results. Others conduct classification on different training sets to test and examine the robustness of their system to deal with clear and fuzzy boundary classes (Aue & Gamon (2005); Yu & Hatzivassiloglou (2003)). As a consequence, the findings based on different strategies to distinguish opinions from facts, or positive opinions from negative opinions, are hardly comparable with each other.

2.1 Definition of Subjectivity

One of the challenges of Sentiment Analysis is defining the objects of the study – opinions and subjectivity. Wikipedia[1] defines subjectivity/objectivity identification as classifying a given text (usually a sentence) into one of two classes: objective or subjective. According to Wiebe *et al.* (1999) subjective sentences are used to communicate the speaker's evaluations, opinions, emotions and speculations, while objective sentences are used to convey objective and factual information. So, the difference between these two important ideas is the difference between facts and opinions. Objective statements are facts that can be verified by third parties, while subjective statements may or may not be entirely true as they are colored by the opinions of the speaker. Facts and opinions are really different in the sense that a fact is something that is true or proven and an opinion is only a belief (Table 2.1). According to the Webster’s Dictionary[2], a fact is "anything that is done or happened; anything actually existent; any statement strictly true; truth; reality" and an opinion is something that "indicates a belief, view, sentiment, or conception". This general definitions show that opinions are subjective and that they are always about something. In the context of opinion mining, however, there does not exist a widely-accepted definition of an opinion beyond the general agreement that an opinion is something that is not a fact. Kim & Hovy (2004) define an opinion as a quadruple (Topic, Holder, Claim, Sentiment), in which the Holder believes a Claim about the Topic, and in many cases associates a Sentiment, such as "good" or "bad", with the belief. The authors distinguish among opinions with sentiment and opinions without sentiment and between directly and indirectly expressed opinions with sentiment.

[1] http://www.wikipedia.com [16th *November*, 2010].
[2] http://www.merriam-webster.com [16th *November*, 2010].

Table 2.1: Examples for facts and opinions.

Facts	Opinions
Pilgrims sailed on the ship "Mayflower"	Everyone had an enjoyable voyage
G. Washington was the first U.S. President	G. Washington didn't smile because of his false teeth
There are nine planets in our Solar System	Earth has the best atmosphere
All mammals have live births	Mammals are really cuddly

2.2 Definition of Opinion Mining

The automatic processing of texts to detect an opinion expressed therein, as a unitary body of research, has been denominated opinion mining or sentiment analysis. Opinion mining is the task of detecting opinionated documents or opinionated portions of a document, normally based on the presence or absence of opinion indicator(s). Opinion indicators are sometimes known as opinion cues, opinion markers, or opinion features. In some cases, opinion detection is treated as a binary classification problem, with the two categories opinion and fact, and is evaluated by classification accuracy. There are also cases, especially in opinion retrieval, where opinion detection involves assigning scores to fragments of texts to indicate how opinionated they are. Several definitions of opinion mining have been proposed in the recent studies. Esuli & Sebastiani (2006b) define opinion mining as a recent discipline at the crossroads of information retrieval and computational linguistics, which is concerned not with the topic a document is about, but with the opinion it expresses. This is a very broad definition, that targets opinions expressed at document level. Dave *et al.* (2003) define an opinion mining system as one that is able to "process a set of search results for a given item, generating a list of product attributes (quality, features, etc.) and aggregating opinions about each of them (poor, mixed, good)". Opinion mining, in this context, aims therefore at extracting and analyzing judgments on various aspects of given products. And, in the recent survey about opinion mining and sentiment analysis (Pang & Lee (2008)), opinion mining is defined as the "computational treatment of opinion, sentiment and subjectivity in text".

2.3 Domain Dependency

Domain dependency may seem less problematic for opinion detection than topical classification since generic opinionated words such as “good” and “bad” are not limited to any particular domain. However, there are very few such generic opinion words and it is therefore necessary to extract opinion-bearing features from domain corpora. These features are domain dependent and may not be reusable in another domain for several reasons:
(1) there are specific opinionated words associated to different domains (e.g., "cheap" and "com-

fortable" are frequently used in product reviews, but not in movie reviews),
(2) different domains have different stylistic expectations for language use (e.g., news articles are less likely than blogs to use words such as "blaah" or "sooooooooo"),
(3) some words can be either subjective or objective depending on the domain (e.g., "actor" could be objective in a movie review but subjective in a political blog).
Since sentiment in different domains can be expressed in different ways, supervised classification techniques have specifically dealt with in-domain sentiment classifications. However, sentiment classifiers need to be customizable to new domains in order to be useful in real-word environments such as the Web. But, because language varies so widely, acquisition of labeled data for each different domain can be time-consuming and expensive. At the same time, differences in vocabulary and writing style across domains can cause supervised models to dramatically increase in error. Therefore, we need to turn all domain-dependent features into domain-independent features.

2.4 Context Dependency

Sentiment analysis is a complex field. As it involves the processing and interpretation of natural language, it must deal with natural languages inherently ambiguous in nature. To give an idea of the challenge involved, some of the issues that need to be dealt with are outlined below. The context in which a piece of text is found plays a large part in determining the sentiment of the text. A sentence such as "Go read the book" demonstrates how important context can be. If mentioned regarding a book, this could be considered as a recommendation, while if it is in reference to a film adaption of a book, it would seem to suggest that the film is not worth watching. It is a challenge to decipher and put to use the context of a piece of text, and this is still an open problem today. The subtlety of opinion expression is reflected in its high sensitivity to context. The same expression may be opinionated in one context and not opinionated in another. For example, "Schwarzenegger is an actor" could be a pure objective statement in a review of a movie as shown in Figure 2.1. However, it could be a negative opinion if used in the context of a presidential election. Context dependency is more noticeable in the case of polarity detection. The word "great" is normally associated to positive opinions, but, in the sentence "It is just great", it may actually express a negative opinion. Thus, not only can we not easily identify simple keywords for subjectivity, but we also find that patterns like “the fact that” do not necessarily guarantee the objective truth of what follows them — and bigrams like “no sentiment” apparently do not guarantee the absence of opinions, either. Therefore, when judging an opinion expression, where and how an expression is delivered may be as important as, if not more important than, what is delivered. To date, there is no effective solution to process context information and sophisticated linguistic analyses are usually required.

Figure 2.1: "Schwarzenegger is an actor" - could be objective or negative statement

2.5 Studying Subjectivity

Research in sentiment analysis has emerged to address the research questions: what is a subjective text? Which features express sentiment? How can these features be detected and measured automatically? Most researchers understand subjectivity as a pragmatic, sentence-level feature (Hatzivassiloglou & Wiebe (2000); Wiebe *et al.* (2004)) and focus most of their efforts on semi-automatic or fully automatic identification, extraction, and evaluation of various subjectivity cues. Boiy *et al.* (2007) defined sentiments as emotions, judgments or ideas prompted or coloured by emotions. They investigated how linguistic elements are used in texts to express the emotion of the author, as they comprised the majority of clues to infer emotions from text. They also discussed dimensions of Osgood (Osgood *et al.* (1971)) as indicators of sentiment in text:

(1) evaluation (positive or negative) - This dimension contains all choices of words, part-of-speech, word organization patterns, conversational techniques, and discourse strategies that express the orientation of the writer to the current topic (e.g. "It was an amazing show").

(2) potency (powerful or unpowerful) - This dimension contains all elements that generally express whether the writer identifies and commits himself towards the meaning of the sentence or whether he dissociates himself. It consists of 3 sub-dimensions: proximity, specificity and certainty.

(2.1) Proximity (near or far) - This category contains all linguistic elements that indicate the "distance" between the writer and the topic. The proximity from the writer to the current topic expresses whether the writer identifies himself with the topic or distances himself from it (e.g. "I'd like you to meet John" versus "I'd like you to meet Mr. Adams" (social proximity)).

(2.2) Specificity (clear or vague) - It is the extent to which a conceptualized object is referred to by name in a direct, clear way; or is only implied, suggested, alluded to, generalized, or

otherwise hinted at (e.g. "I left my / a book in your office" (particular vs general reference)).
(2.3) Certainty (confident or doubtful) - This dimension expresses the certainty of the writer towards the expressed content. A stronger certainty indicates that the writer is entirely convinced of the truth of his writings and possibly indicates a stronger emotion (e.g. "It supposedly is a great movie" versus "It definitely is a great movie").
(3) Intensifiers (more or less) - A lot of the emotional words used do not express an emotion, but modify the strength of the expressed emotion. These words, the intensifiers, can be used to strengthen or weaken both positive and negative emotions (e.g. "This is simply the best movie").

2.6 Summary

In this chapter we have discussed some challenges that make sentiment classification much more difficult. In general, sentiment and subjectivity are quite context-sensitive and domain dependent, which complicates the classification process. In this thesis, we propose a solution to sentiment classification problem when we do not have any labeled data from target domain but have some labeled data in a different domain, regarded as source domain.
In the next chapter we give an overview of the most important works, that consider the same matter, published so far and describe some already developed and used methodologies.

Chapter 3

Related Work

"*In every fat book there is a thin book trying to get out.*"

Anonymous

Recent years have seen a growing interest in the detection of subjective content in text, which has led to a number of approaches. Many features have been used to characterize opinionated texts at different levels: words (Hatzivassiloglou & McKeown (1997), Esuli & Sebastiani (2005)), sentences (Wiebe *et al*. (1999)) and texts (Pang *et al*. (2002), Wiebe *et al*. (2004)). Other research in the sentiment classification field regards cross-domain classification (Aue & Gamon (2005), Blitzer *et al*. (2007)). Finally, over the past few years, semi-supervised and multi-view learning proposals have emerged (Ganchev *et al*. (2008), Blum & Mitchell (1998)). So, in this chapter we review in some depth several works, which have influenced or are otherwise theoretically relevant to the work we present in this thesis. The first section surveys a range of different sentiment analysis tasks and opinion-bearing features, within which the work described in this thesis was developed. Then, in section 3.2, we discuss other research works, which focus on cross-domain sentiment classification. Finally, in section 3.3, we look at some recent works, which explore semi-supervised and multi-view learning proposals.

3.1 In-Domain Single-View Sentiment Classification

In this section, we review some related works, which presented sentiment classification or studied subjectivity in a single domain and proposed some important opinion-bearing features. In particular, they recognized opinions from various granularities such as word, sentence or text, each of which is essential. The level of granularity is determined mainly by the purpose of the application. For example, detecting opinions at the document level is usually enough for opinion retrieval, but not for opinion summarization, which requires identification of opinions at the level of sentences or paragraphs. So, this section contains an overview of the key areas of related work regarding each of these levels of opinion detection.

3.1.1 Word Level Sentiment Classification

Opinion detection at the word level is used to determine whether a single word or phrase is opinionated within a certain context, such as the sentence or document in which the term appears. However, in practice, term-level opinion detection is rarely the ultimate goal. Instead, it provides a set of words that can serve as opinion-bearing features for higher level opinion detection.

At word level, Hatzivassiloglou & McKeown (1997) were the first to distinguish between positive and negative adjectives using conjunctional relations. They proposed that, if two adjectives are joined together with a conjunctive word such as "and", there is a good chance that they share the same opinion polarity (e.g., "nice and warm"). In contrast, if two adjectives are connected with a conjunctive word such as "but", they are likely to have opposite opinion polarities (e.g., "nice but expensive"). This is shown in the following three sentences (where the first two are perceived as correct and the third is perceived as incorrect taken from Hatzivassiloglou & McKeown (1997)).

1. The tax proposal was simple and well received by the public.

2. The tax proposal was simplistic but well received by the public.

3. (*) The tax proposal was simplistic and well received by the public.

* The third sentence was incorrect, because "and" is usually used with adjectives that have the same semantic orientation ("simple" and "well-received" are both positive), but "but" is used with adjectives that have different semantic orientations ("simplistic" is negative).

Combining the constraints across many adjectives, a clustering algorithm separated the adjectives into groups of different orientations, and finally, adjectives were labeled positive or negative. For that purpose, Hatzivassiloglou & McKeown (1997) used a four-step supervised learning algorithm to infer the semantic orientation of adjectives from constraints on these conjunctions (Algorithm 1).

For their experiments, they used the 21 million word Wall Street Journal corpus[1], automatically annotated with a part-of-speech tagger. Their algorithm classified adjectives with accuracies ranging from 78% to 92% depending on the amount of available training data. Best classification accuracy was achieved for identifying polar adjectives when only those conjunctions that occurred five or more times were retained in the first step. In particular, they based their study on the hypothesis that positive adjectives are more frequent than negative ones. One drawback of the simple conjunction hypothesis is that it is not realistic to assume that all the words in a conjunction demonstrate polarity.

[1] Available form the ACL Data Collection Initiative as CD ROM 1.

Algorithm 1 Algorithm for adjective classification.

1: All conjunctions of adjectives are extracted from the corpus along with relevant morphological relations.
2: A log-linear regression model combines information from different conjunctions to determine if each two conjoined adjectives are of same or different orientation. The result is a graph that links adjectives using dissimilarity scores between 0 and 1 based on the type of conjunction (i.e. "same orientation," "reverse orientation" or "no conjunction") between each pair of adjectives.
3: A clustering algorithm separates the adjectives into two subsets of different orientation. It places as many words of same orientation as possible into the same subset.
4: Assignment of the label "positive" to the subset containing adjectives with high average frequency in the corpus, keeping with their observation that the corpus contained a large proportion of positive adjectives.

Bruce & Wiebe (1999) performed a statistical analysis of assigned sentence classifications, finding that adjectives are statistically significantly and positively correlated with subjective sentences in corpora on the basis of the log-likelihood ratio test statistic. Indeed, they found that the probability of a sentence being subjective was 56% by simply knowing that the sentence contained at least one adjective even though there were more objective than subjective sentences in the corpus. In addition, they identified that dynamic adjectives (Quirk *et al.* (1985)) are indicative of subjective sentences. Dynamic adjectives denote attributes, which are to some extent at least, under the control of the one who possesses them. For instance, "brave" denotes an attribute, which may not always be in evidence (unlike "red", for example). So, they manually applied syntactic patterns to identify a list of 124 dynamic adjectives from about 500 sentences in the Wall Street Journal Treebank Corpus. A preliminary examination indicated that these dynamic adjectives were more subjective than the rest of the adjectives in the corpus.
In the same domain but using a different corpus, Hatzivassiloglou & Wiebe (2000) confirmed the strong correlation between dynamic adjectives and subjectivity with more than 30% improvement in precision over adjectives as a whole. They suggested that another type of adjectives, gradable adjectives, were also useful opinion indicators that had 13-21% higher precision than the adjectives as a whole. Gradable adjectives are those that can participate in comparative constructs (e.g., "That website is reasonably popular. But this one is more popular.") and accept modifying expressions that act as intensifiers (e.g., "My teacher was very happy with my homework.", where "very" is an intensive modifier). Gradable adjectives can be identified manually or automatically using a statistical model, but the latter is computationally exhaustive because of the requirement for syntactic parsing and morphological analysis. They also examined another type of adjectives, which is more intuitive as opinion evidence. Unlike dynamic and gradable adjectives, which are judged by the context of usage, semantic-oriented adjectives

can be identified more easily. Semantic-oriented adjectives are polar words that are either positive or negative. Adjectives with polarity, such as "good", "bad", or "beautiful" are inherently connected to opinions as opposed to adjectives, such as "black" or "Portuguese", which have no polarity. In particular, they noted that all sets involving dynamic adjectives, gradable adjectives and adjectives with positive or negative polarity were better predictors of subjective sentences than the class of adjectives as a whole. As a consequence, within the context of our research we used the proportion of dynamic, gradable and semantically oriented adjectives in texts to characterize their level of subjectivity.

Later, Turney (2002) proposed an unsupervised algorithm to classify reviews as positive or negative. Three steps are conducted. First, phrases that contain adjectives or adverbs are identified by using a part-of-speech tagger. Second, each extracted phrase is labeled with a semantic orientation. And third, a given review is classified as positive or negative (which Turney calls "recommended" or "not recommended") based on the average semantic orientation of phrases in that review. The reason to extract phrases containing adjectives and adverbs is that they can indicate subjective and evaluative sentences. Using a phrase instead of a single word can also help to identify a context and hence to improve semantic orientation. For instance, "unpredictable" might be negative in certain domains, but "unpredictable plots" should be positive in the movie domain. The method of extracting phrases is based on utilizing part-of-speech patterns. For example, two adjacent words are extracted if the first word is an adverb and the second word is an adjective, while the third word should not be a noun. The most interesting procedure is the second step where the semantic orientation of phrases is labeled. Turney calculated the semantic orientation of a phrase based on the Semantic Oriented Pointwise Mutual Information ($SO-PMI$) measure. For that purpose, Turney evaluated the Pointwise Mutual Information of the given phrase and the word "excellent" minus the PMI with the word "poor". Sentiment orientation of the phrase ph is estimated as in Equation 3.1, where pos_words represents pre-defined positive words such as, "excellent, good", and neg_words represents pre-defined negative words such as "poor, bad".

$$SO-PMI(ph) = PMI(ph, pos\_words) - PMI(ph, neg\_words). \tag{3.1}$$

The pointwise mutual information (Church & Hanks (1989)) between ph and $words$ is defined as in Equation 3.2, where $P(ph, words)$ is the probability that ph and $words$ co-occur.

$$PMI(ph, words) = \log_2 \frac{P(ph, words)}{P(ph).P(words)}. \tag{3.2}$$

If the words are statistically independent, the probability that they co-occur is given by the product $P(ph).P(words)$. The ratio between $P(ph, words)$ and $P(ph).P(words)$ is a measure of the degree of statistical dependence between ph and $words$.

In particular, Turney calculated the PMI based on the number of Web pages returned by search engines, when the pair of the phrase and the word were queried. The selection of words is not totally random but is motivated by the domain properties of the unlabeled target words. For example, Turney chose "excellent" and "poor" because they are commonly used in the target domain of movie reviews. Finally, a review polarity is predicted from the average semantic orientation (positive or negative) of the phrases it contains. A positive $SO - PMI$ score implies that the target word has a stronger association with positive seeds than with negative ones and thus should be labeled as "positive". Accordingly, a negative $SO - PMI$ score implies the "negative" class label. The method achieved an average accuracy of 74% when evaluated on 410 reviews from Epinions[1], sampled from four different domains (reviews of automobiles, banks, movies and travel destinations). In particular, the accuracy ranged from 84% for automobile reviews to 66% for movie reviews. According to him, the difficulty with movie reviews is that there are two aspects to note a movie, (1) the events and actors in the movie (the elements of the movie), and (2) the style and art of the movie (the movie as a gestalt; a unified whole).

Yi *et al.* (2003), proposed to extract positive and negative opinions about specific features of a topic. By feature terms they mean terms that have either a part-of or attribute-of relationship with the given topic or with a known feature term of the topic. They developed a system called Sentiment Analyzer that first extracts what are the relevant subject terms, and then assigns an opinion polarity to statements about them. This is done by using a sentiment dictionary of words and phrases that have been labeled with a known sentiment value. Their method first determines candidate feature terms based on structural heuristics and then narrows the selection using a mixture language model and the log-likelihood ratio. A pattern-dependent comparison is then made to a sentiment lexicon gathered from a variety of linguistic resources. They identified evaluation expressions using a dictionary, which they built using external resources such as WordNet (Miller (1990)) and the General Inquirer[2]. The size of the dictionary is approximately 3000 words (2500 adjectives and less than 500 nouns). In particular, they automatically extracted these expressions using rules and scores based on the log likelihood. To identify relations between aspects and evaluations, they used manually-created patterns. The patterns have the following two types: (target, verb, source) and (adjective, target). For example, ("the digital zoom", "be", "too grainy") matches the first pattern and ("good quality", "photo") matches the second one. The method was evaluated on two domains, digital camera and 9 music review articles, using topic relevance judgments performed by the authors, and achieved precision of 87% and recall of 56%.

Finally, Esuli & Sebastiani (2005) and Esuli & Sebastiani (2006a) presented a semi-supervised methodology to identify the semantic orientation of words using their gloss definitions from

[1] http://www.epinions.com/ [16th $November, 2010$].
[2] http://www.wjh.harvard.edu/inquirer/ [16th $November, 2010$].

online dictionaries. In particular, they provided a manually-composed set of words with positive and negative connotation and expanded it with the synonyms of the polar words. Finally, they used this expanded data set to predict the polarity of words on the basis of their glosses. The assumption here is that terms with similar orientation tend to have "similar" glosses. For instance, the glosses of "honest" and "intrepid" will both contain appreciative expressions, while the glosses of "disturbing" and "superfluous" will both contain derogative expressions. Their process is composed of the following steps as shown in Algorithm 2.

Algorithm 2 Algorithm for identifying semantic orientation of words using their glosses.

1: A seed set (Sp, Sn), representative of the two categories Positive and Negative, is provided as input.

2: Lexical relations (e.g. synonymy) from a thesaurus, or online dictionary, are used in order to find new terms that will also be considered representative of the two categories because of their relation with the terms contained in Sp and Sn. This process can be iterated. The new terms, once added to the original ones, yield two new, richer sets $S'p$ and $S'n$ of terms. Together, they form the training set for the learning phase of Step 4.

3: For each term t_i in $S'p \bigcup S'n$ or in the test set (i.e. the set of terms to be classified), a textual representation of t_i is generated by collating all the glosses of t_i as found in a machine-readable dictionary. Each such representation is converted into a vectorial form by standard text indexing techniques.

4: A binary text classifier is trained on the terms in $S'p \bigcup S'n$ and then applied to the terms in the test set.

In particular, they have classified terms by learning a classifier from the vectorial representations of the terms in $S'p \bigcup S'n$, and then applying the resulting binary classifier (Positive vs. Negative) to the test terms. It is important to note that they obtained vectorial representations for the terms from their textual representations by performing stopword removal and weighting by the cosine similarity measure and the well-known normalized $tfidf$ (Salton *et al.* (1975)). When tested on publicly available corpora, this method outperformed the published methods of (Hatzivassiloglou & Wiebe (2000), Kamps *et al.* (2004), Turney & Littman (2003)) although the best-performing known method (Turney & Littman (2003)) is beaten only by a small margin. Besides accumulating glosses for every possible sense associated with a target word, they also used glosses at the level of word sense (i.e. the gloss for each sense serves as a distinct representation of the target word). Therefore, if a word has several senses, it will have multiple glosses and thus be classified several times. As such, Esuli & Sebastiani (2006b) were able to classify any WordNet synset, which represents a unique sense defined by a unique gloss, into positive, negative and objective classes. They constructed a publicly available lexical resource called Senti-Wordnet to visually represent opinion-related properties for each word sense in WordNet. In Senti-Wordnet, each synset in WordNet is associated to three numerical scores $Obj(s)$, $Pos(s)$

and $Neg(s)$ to describe how objective, positive, and negative are the terms contained in the synset. Their method to construct the Senti-WordNet is based on the quantitative analysis of the glosses associated with the synsets, and on the use of the resulting vectorial term representations for semi-supervised synset classification.

In the related works listed above, we focused on word level opinion detection, i.e., finding words or phrases that carry a positive or negative sentiment (opinion-bearing word). Actually, opinion-bearing words are the smallest unit of opinion that can thereafter be used as a clue for sentence-level opinion detection, which we will discuss in the next section.

3.1.2 Sentence Level Sentiment Classification

Opinion detection at the sentence level is maybe the most popular approach. Although automated sentence-level opinion detection is feasible, it is still quite challenging because of the sparsity of opinion-bearing features in short text units.

Wiebe *et al.* (1999) described a corpus tagged at the sentence level for subjectivity and a Naive Bayes classifier using syntactic classes, punctuation and sentence position as features. By using only simple features, such as pronouns, adjectives, cardinal numbers, modals other than "will", and adverbs other than "not", they achieved an average accuracy of 21% points higher than the baseline[1] in 10-fold cross validation experiments. They also considered whether a sentence was at the beginning of a paragraph and the co-occurrence of word tokens and punctuation marks. A classification accuracy of 72.17% was achieved on the Wall Street Journal dataset. This modest result may have been due to insufficient features or to the difficulty of sentence level classification. Aligned with work related to subjectivity cues, corpus annotation with subjectivity mark-ups has been studied to investigate the use of subjective language as well as to create annotated corpora for future training and evaluation. Observations of the annotation procedure and its outcomes were valuable to find general characteristics of subjective language and to implement failure analysis of opinion detection tasks. In addition, the average accuracy of the classifier was 81.5% on the sentences the judges tagged with certainty.

Later, Yu & Hatzivassiloglou (2003) dealt with differentiating opinions from facts at document level as well as sentence level. They proposed a technique based on the subjectivity scoring proposed by Turney (2002) being a sentence positive (resp. negative) if most of the adjectives, adverbs, nouns and verbs in the sentence were positive (resp. negative). Similarly, to avoid the need to obtain individual sentence annotations for training and evaluation, they relied instead on the expectation that documents generally classified as opinion (e.g., editorials) would tend

[1] The baseline accuracy is the frequency of the most frequent class and it is 51%.

to have mostly opinion sentences, and conversely - documents placed in the factual category would tend to have mostly factual sentences. At document level, they presented a Bayesian classifier to discriminate between documents with a preponderance of opinions such as editorials from regular news stories. For their experiments, they used Wall Street Journal articles, which contain metadata for identifying the type of each article, such as Editorial, Letter to editor, Business and News. These labels were used only to provide the correct classification labels during training and evaluation, and were not included in the feature space. As features, they used single words, without stemming or stopword removal. At the sentence level, they described three unsupervised statistical techniques to detect opinions. Their first approach to classify sentences as opinions or facts explored the hypothesis that, within a given topic, opinion sentences would be more similar to other opinion sentences than to factual sentences. They used Simfinder (Hatzivassiloglou *et al*. (2001)), a state-of-the-art system to measure sentence similarity based on shared words, phrases and WordNet synsets. To measure the overall similarity of a sentence to the opinion or fact documents, they first selected the documents that were about the same topic as the sentence in question. They obtained topics as the results of Information Retrieval (IR) queries (for example, by searching document collection for "welfare reform"). Then, they averaged Simfinder provided similarities with each sentence in those documents. Then, they assigned the sentence to the category for which the average was higher. Alternatively, for the "frequency" variant, they did not use the similarity scores themselves but instead they counted how many of them, for each category, exceeded a determined threshold (empirically set to 0.65). Their second method trained a Naive Bayes classifier, using the sentences in opinion and fact documents as examples of the two categories. The features included words, bigrams and trigrams as well as the part-of-speech tags in each sentence. In addition, the presence of semantically oriented (positive and negative) words in a sentence was an indicator that the sentence is subjective and, therefore, they included in their features the counts of positive and negative words in the sentence, as well as counts of the polarities of sequences of semantically oriented words (e.g., "++" for two consecutive positively oriented words). They also included the counts of part-of-speech tags combined with polarity information (e.g., "JJ+" for positive adjectives), as well as features encoding the polarity (if any) of the head verb, the main subject, and their immediate modifiers. They obtained syntactic structure with a statistical parser. Finally, they used, as one of the features, the average semantic orientation score of the words in the sentence. As a third method, they applied an algorithm using multiple classifiers, each relying on a different subset of features. The goal was to reduce the training set to the sentences that were most likely to be correctly labeled, thus boosting classification accuracy. Given separate sets of features $\{F_1, F_2 \ldots F_n\}$, they trained separate Naive Bayes classifiers $\{C_1, C_2 \ldots C_n\}$ corresponding to each feature set. Assuming as ground truth the information provided by the document labels and that all sentences inherit the status of their document as opinions or facts, it first train C_1 on the entire training set and then use the resulting classifier

to predict labels for the training set. The sentences that received a label different from the assumed truth were then removed. Then, they trained C_2 on the remaining sentences. This process was repeated iteratively until no more sentences could be removed. Finally, they reported results using five feature sets, starting from words alone and adding in bigrams, trigrams, part-of-speech and polarity. Results from a large collection of news stories and a human evaluation of 400 sentences were reported, indicating that they achieved very high precision and recall (F-measure of 97%) in document classification, and respectable performance in detecting opinions and classifying them at the sentence level as positive, negative, or neutral (up to 91% precision and recall). This work is closely related to the one proposed by Turney (2002), where a review is classified as recommended if the average semantic orientation of its phrases is positive.

Although opinion detection at the sentence-level is the most popular approach, document-level opinion detection is directly related to our work. Also document-level opinion detection is useful to many real-world applications such as retrieving opinion posts from the blogosphere. It can be easily integrated into existing Web mining systems, which usually work with document-level information.

3.1.3 Document Level Sentiment Classification

At document level, Pang *et al.* (2002) first showed that the unigram model with SVM reached best results compared to more complex models in the domain of movie reviews. They evaluated three different supervised learning algorithms (Naive Bayes classification, Maximum Entropy classification and Support Vector Machines) and eight different sets of features, which are used in the traditional categorization task (unigrams, bigrams, etc.) to classify movie reviews as either containing positive or negative opinions. According to them, tagging adjectives alone did not give enough information and would perform badly, since there were many words indicative of opposite sentiments in the target documents. However, tagging unigrams with parts-of-speech gave better results as compared to the tagging of adjectives alone. The best result was 82.9%, by using SVM with features based on the presence or absence (rather than the frequency) of single words (rather than two-word phrases). Although this is worse than the accuracy of 90% that could be achieved in the topical classifications as reported by Joachims (1999), the machine learning algorithms clearly surpassed the random-choice baseline of 50%. They also handily beat the two human-selected-unigram baselines of 58% and 64% and, furthermore, performed well in comparison to the 69% baseline achieved via limited access to the test data statistics.

A later study by Pang & Lee (2004) found that performance increased up to 87.2% when considering only those portions of the text deemed to be subjective. This approach is based on the assumption that the semantic orientation of the document relies only on the sentences expressing the writer's subjectivity. In particular, a sentence cut-based classifier was used to identify

subjective parts in texts, which are then used for text classification. The cut-based classifier approach puts forward the hypothesis that text spans (items) occurring near each other (within discourse boundaries) might share the same subjectivity status. Based on this hypothesis, Pang & Lee (2004) supplied an algorithm with pair-wise interaction information (e.g., to specify that two particular sentences should ideally receive the same subjectivity label). This algorithm uses an efficient and intuitive graph-based formulation relying on finding minimum cuts. Suppose there are n items $\{x_1, x_2, \ldots, x_n\}$ to divide into two classes C_1 and C_2. We access to two types of information: $indj(x_i)$ (Individual scores) which is the non-negative estimate of each x_i preference for being in C_j based just on the features of x_i alone and $assoc(x_i, x_k)$ (Association scores) which is the non-negative estimate of how important it is that x_i and x_k are in the same class. The problem can be summarized to an optimized problem, which is to assign each x_i to C_1 and C_2, so as to minimize the partition cost in Equation 3.3

$$\sum_{x \in C_1} ind_2(x) + \sum_{x \in C_2} ind_1(x) + \sum_{x_i \in C_1, x_k \in C_2} assoc(x_i, x_k). \tag{3.3}$$

This situation can be represented in the following manner. Build an undirected graph G with vertices $\{v_1, \ldots, v_n, s, t\}$, where the last two are, respectively, the source and the sink. Add n edges (s, v_i), each with weight $ind_1(x_i)$, and n edges (v_i, t), each with weight $ind_2(x_i)$. Finally, add $\binom{n}{2}$ edges (v_i, v_k), each with weight $assoc(x_i, x_k)$. A cut (S, T) of G is a partition of its nodes into sets $S = \{s\} \bigcup S'$ and $T = \{t\} \bigcup T'$, where $s \notin S'$ and $t \notin T'$. Its cost $cost(S, T)$ is the sum of the weights of all edges crossing from S to T. A minimum cut of G is one of minimum cost. Then, finding a solution to this problem is changed into looking for a minimum cut of G. Their results showed that the created subjectivity extracts accurately represented the sentiment information of the originating documents in a much more compact form. In fact, depending on the choice of the polarity classifier, they were able to achieve highly statistically significant improvements (from 82.8% to 86.4%) for the polarity classification task while retaining only 60% of the words of the reviews.

Wiebe *et al.* (2004) is certainly the first work to propose a study of subjectivity in text and not just polarity. Specifically, they derived a variety of subjectivity cues (frequencies of unique words in subjective-element data, collocations with one or more positions filled in by a unique word and distributional similarity of adjectives and verbs) from corpora and demonstrated their effectiveness on classification tasks. Moreover, they determined a relationship between low frequency terms and subjectivity and found that their method to extract subjective n-grams was enhanced by examining those that occur with unique terms. Collocations identified in Wiebe *et al.* (2004) are n-grams (n=1..4) consisting of pairs of word-stem and part-of-speech tags (e.g., "could-modal have-verb") extracted from an annotated opinion expression. To avoid

selecting unnecessarily long collocations, an n-gram is only kept if its precision is higher than the maximum precision of its constituents. For example, the collocation "could-modal have-verb be-verb" is accepted only if it has higher precision than both "could-modal have-verb" and "be-verb," or if it has higher precision than both "could-modal" and "have-verb be-verb". These collocations are called fixed n-grams. Specifically, they calculated the precision of a set S with respect to subjective elements as shown in Equation 3.4.

$$K_4 = \frac{number\ of\ instances\ of\ members\ of\ S\ in\ subjective\ elements}{total\ number\ of\ instances\ of\ members\ of\ S\ in\ the\ data}. \tag{3.4}$$

Another form of collocations investigated by Wiebe et al. was extracted based on the uniqueness feature of opinions. All unique words in previous collocations were replaced by a placeholder, UNIQ, with the part-of-speech tags kept. These unique generalized n-grams (ugen-n-gram) turned out to be the best features in their test with a maximum increase in precision of more than four times over the baseline. So, the unique and creative nature of opinion expression proved to be a strong source of evidence for opinion detection. They observed that the difference between the proportion of unique words in opinionated documents and non-opinionated documents was also significant, with ($p < 0.001, z >= 22$). Wiebe *et al.* (2004) pointed out that the quality of the unique words was related to the size of the corpus and only in a corpus of sufficient size these cues were informative. When the training set was small, using unique terms as features degraded performance, probably due to over-generalization. Therefore, they recommended using additional un-annotated data (e.g., the entire corpus) to identify uniqueness features. Other clues learned from annotated data include distributionally similar adjectives and verbs. The adjectives and verbs were learned from the Wall Street Journal data using word clustering method according to their distributional similarity. The seed words for this process were the adjectives and verbs in editorials and other opinion-piece articles. Wiebe *et al.* (2004) reported the difference between the proportion of selected adjectives in opinion documents and non-opinion documents as significant with $(p < 0.001, z >= 9.2)$. The precision of opinionated verbs was also consistently higher than the baseline. The difference between the proportion of instances of those verbs in opinion documents and non-opinion documents was indeed significant ($p < 0.001, z >= 4.1$).

Finally, Chesley *et al.* (2006) examined the novel idea of using linguistic features such as verb class information and Wiktionary[1] for subjectivity classification. They used verb-class information in the sentiment classification task, since exploiting lexical information contained in verbs showed to be a successful technique for classifying documents, unlike previous research. They fed a mix of part-of-speech tag based features (first-person pronouns, second-person pronouns, adjectives and adverbs), verb classes, positive/negative adjectives and punctuation marks into a

[1] http://www.wiktionary.org/ [16th $November, 2010$].

SVM classifier and got 76.3%, 86.8% and 80.3% classification accuracy for objective, positive and negative blog post classification, respectively. These results were considered promising given the noisy nature of blog posts. In particular they found that the "asserting" and "approving" verb classes played a key role in improving the accuracy of classifying blog posts as positive opinion. Also, for objective and positive posts, positive adjectives acquired from Wictionary seemed to play a key role in increasing the overall accuracy. They also observed a slight negative effect when applying the first and the second person pronouns as features. The inconsistency in these findings might have been due to the frequent use of pronouns even in objective statements (e.g., "I heard that there is a big sale at Macy's" or "Here is the recipe you asked for"). Another conclusion from their results was that the number of exclamation marks improved the performance by around 4%, while the number of question marks showed no effect in classifying non-opinionated blog posts. It is reasonable to make the assumption that, when people express their opinion in informal Web media such as personal blogs, they tend to use more than one exclamation mark to emphasize their feelings (e.g., "This movie is wonderful!!!") than would a formal news report or product description. To derive the polarity of adjectives, they queried Wiktionary for each adjective and fetched its entry page. This entry page is normally organized into sections for pronunciation and word classes (e.g., adjective, verb) and is then divided into subsections such as definitions, synonyms, antonyms, translations and examples of use. In order to avoid part-of-speech ambiguity, non-adjective sections on the entry page were excluded. After filtering out antonyms and usage examples, the entry page only contained definitions, synonyms, related words and other descriptive information on all meaning senses of the target word. As a consequence, an adjective was labeled as positive if its entry page contained more matches with a positive seed set than with a negative seed set. This method yielded classification accuracy of 90.9% for manually labeled adjectives in the test data. The success of this simple approach may have been due to the rich representation of the target word as well as to lowered sense ambiguity in Wiktionary given that, when compared to WordNet, Wiktionary holds only the most common word senses.

3.2 Cross-Domain Single-View Sentiment Classification

Unfortunately, all studies presented so far learn domain-dependent classifiers or study subjectivity in a single domain, perhaps with the exception of Wiebe *et al.* (2004). However, within the context of real-world environments, especially the Web, sentiment analysis must be dealt across domains. But, as mentioned in Boiy *et al.* (2007), most research show difficulties in crossing domains. In particular, we showed in Lambov *et al.* (2009b) that accuracy losses of 35% can be reached when evaluating a unigram domain-dependent SVM classifier against a cross-domain data set. Indeed, sentiment is orthogonal to topic and sentiment classification is clearly more

difficult than topic classification. Tests have been done by Aue & Gamon (2005), Finn & Kushmerick (2006), Boiy *et al.* (2007) and Blitzer *et al.* (2007) to assess this statement. Generally, they showed that sentiment analysis is a domain-specific problem, and it is hard to create a domain independent classifier. One possible approach to tackle cross-domain classification is to train a classifier on a domain-mixed data set instead of one specific domain. This idea is proposed by Aue & Gamon (2005) to learn a polarity classifier. They proposed and compared four strategies for utilizing opinionated data from one or more domains that were different from the target domain: (1) training with labeled data from a non-target domain only, (2) training with only those features from labeled data from a non-target domain that occurred in the target domain, (3) training with an ensemble of classifiers built on different subsets of the labeled data from a non-target domain and (4) training with a small number of pre-labeled data and labeled data from the target domain via bootstrapping. The last strategy generated the best performance, which confirmed the domain dependency characteristics of opinion and suggested the use of training data from the same or similar domains when possible. They based their experiments on data from four different domains. The different types of data they considered ranged from lengthy movie reviews to short, phrase-level user feedback to Web surveys. Due to the significant differences in these domains (different subject matter as well as different styles and lengths of writing), simply applying the classifier learned on data from one domain can barely outperform the baseline for another domain. After establishing that naive cross-domain classification results in poor classification accuracy, they compared results obtained by using each of the four approaches. Aue & Gamon (2005) based their approach on feature vectors consisting of the presence of unigrams, bigrams or trigrams. Each document was represented as a feature vector. Only features that occurred 3 times or more in any of the domains were included in the feature vectors for that domain. N-gram features were binary, i.e. only absence versus presence of an n-gram in a document was indicated, not the frequency of that n-gram. They tested different compositions of training material and a reduction of feature space to shared features across domains. Depending on domains and chosen training methods, accuracies between 74% and 82% were reported. In fact, with 100 or 200 labeled items in the target domain, an expectation-maximization (EM) algorithm that utilized in-domain unlabeled data and ignored out-of-domain data altogether outperformed the method based exclusively on both in- and out-of-domain labeled data. They observed that source domains closer to the target helped more. Adding more labeled data always helped, but diversifying training data did not. For example, when classifying kitchen appliances, for any fixed amount of labeled data, it was always better to draw from electronics as a source than use some combination of all three other domains.

Another possibility is to propose high-level features, which do not depend so much on topics such as part-of-speech statistics as in Finn & Kushmerick (2006). In this case, the part-of-speech representation does not reflect the topic of the document, but rather the type of text used in the

document. Just by looking at part-of-speech statistics, improved results can be obtained comparatively to unigram models (low-level models) when trying to cross domains. In particular, they explored three different ways to encode a document as a vector of features. The first approach represented each document as a bag-of-words (BOW), a standard approach in text classification. A document was encoded as a feature-vector, with each element in the vector indicating the presence or absence of a word in the document. The second approach used the output of Brill's part-of-speech tagger (Brill (1994)) as the basis for its features. By using this information, they expected that part-of-speech tagging statistics would reflect the style of the language sufficiently for their learning algorithm to distinguish between different genre classes. A document was represented as a vector of 36 part-of-speech features, expressed as a percentage of the total number of words of the document. The part-of-speech tags representation does not reflect the topic of the document, but rather the type of text used in the document. So, if a part-of-speech tag feature-set is capable of differentiating genre class, they could expect that it would do so in a domain independent manner as it does not have any information about the topic of the document. The third approach used a set of shallow text statistics. They used many of these features because they showed to have discriminatory value between genre classes in the related literature. This feature set included average sentence length, the distribution of long words, average word length. Additional features are based on the frequency of occurrence of various function words and punctuation symbols. They evaluated classifiers using two measures: accuracy of the classifier in a single topic domain and accuracy when trained on one topic domain but tested on another. For an in-domain classification task, all three feature-sets achieved good accuracy on this classification task, indicating that any of these feature-sets alone was sufficient to build classifiers within a single topic domain. However, bag-of-words performed best in all three topic domains on this task, reaching average accuracy of 87.3%. This indicates that there are keywords within each topic domain that indicate the subjectivity of a document. In the case of domain transfer for the subjectivity classification task, the part-of-speech feature-set performed best reaching average accuracy of 78.5%, while the bag-of-words feature-set performed worst with average accuracy of 63.7%.

Recently, an interesting language-independent methodology has been proposed to leverage the problem of cross-domain classifiers. Blitzer *et al.* (2007) explicitly addressed the domain transfer problem for sentiment polarity classification by extending the structural correspondence learning algorithm (SCL), achieving an average of 46% improvement over a supervised baseline for sentiment polarity classification of 5 different types of product reviews mined from Amazon.com. The SCL is a state-of-the-art sentiment transfer algorithm, which automatically induces correspondences among features from different domains. It identifies correspondences among features from different domains by modeling their correlations with pivot features, which are features that behave in the same way for discriminative learning in two different domains.

The success of the SCL depends on the choice of pivot features in two domains, based on which the algorithm learns a projection matrix that maps features in the target domain into the feature space of the source domain. In this case, the pivots are chosen not only based on their common frequency but also according to their mutual information with the source labels. Depending on the domain, a small number (50) of labeled examples allows the model to adapt itself to a new domain. However, the performance and the minimum number of in-domain labeled examples were found to depend on the similarity between both domains. For example, while many of the features of a good cell phone review are the same as a computer review i.e. the words "excellent" and "awful", many words are totally new, like "reception". At the same time, many features, which are useful for computers, such as "dual-core" are no longer useful for cell phones. Based on these assumptions, their key intuition was that even when "good-quality reception" and "fast dual-core" are completely distinct for each domain, if they both have high correlation with "excellent" and low correlation with "awful" on unlabeled data, then they can tentatively align them. So, after learning a classifier for computer reviews, a cell-phone feature like "good quality reception", should behave in a roughly similar manner as "fast dual-core". Given labeled data from a source domain and unlabeled data from both source and target domains, the SCL first chooses a set of m pivot features, which occur frequently in both domains. Then, it models the correlations between the pivot features and all the other features by training linear pivot predictors to predict occurrences of each pivot in the unlabeled data from both domains. Any pivot predictor is characterized by a weighted vector, where positive entries mean that a non-pivot feature (e.g. "fast dualcore") is highly correlated with the corresponding pivot (e.g. "excellent"). Then, the projection is able to capture correspondences (in terms of expressed sentiment polarity) between "predictable" for book reviews and "poorly designed" for kitchen appliance reviews. Furthermore, they also showed that a measure of domain similarity can correlate well with the ease of adaptation from one domain to another, thereby enabling better scheduling of annotation efforts. For that purpose, they evaluated the A-distance (Shai Ben-David & Pereira (2006)) between domains as a measure of the loss due to adaptation from one to the other. The A-distance can be measured from unlabeled data and was designed to take into account only divergences, which affect classification accuracy. They showed that it correlates well with adaptation loss, indicating that they can use the A-distance to select a subset of domains to label as sources. As a consequence, they identified a measure of domain similarity that correlated well with the potential for adaptation of a classifier from one domain to another. Within that context, best polarity results across domains reached an excellent score of 82.1% accuracy[1].

[1] Although the text data set is based on reviews for each domain, which may bias the results.

3.3 Cross-Domain Multi-View Sentiment Classification

Finally, over the past few years, semi-supervised and multi-view learning proposals have emerged. Multi-view learning refers to a set of semi-supervised methods which exploit redundant views of the same input data (Blum & Mitchell (1998), Collins & Singer (1999), Brefeld *et al.* (2005), Sindhwani & Niyogi (2005)). However, although semi-supervised learning is usually associated to small labeled data sets and tries to automatically label new examples, multi-view learning aims at learning a compromise model of the different views. The most important work in multi-view sentiment classification is proposed by Ganchev *et al.* (2008), who presented a new algorithm called stochastic agreement regularization (SAR), which outperformed the results proposed earlier by Blitzer *et al.* (2007) on the same data set. In particular, they proposed a co-regularization framework for learning across multiple related tasks with different output spaces. They presented a new algorithm for probabilistic multi-view learning, which uses the idea of stochastic agreement between views as regularization. Their SAR algorithm works on structured and unstructured problems and generalizes to partial agreement scenarios. It models a probabilistic agreement framework based on minimizing the Bhattacharyya distance (Kailath (1967)) between models trained using two different views. They regularize the models from each view by constraining the amount by which they permit them to disagree on unlabeled instances from a theoretical model. Their co-regularized objective, which has to be minimized is defined in Equation 3.5 where L_i for $i = 1, 2$ are the standard regularized log likelihood losses of the models $p1$ and $p2$, $E_u[B(p_1(\theta_1), p_2(\theta_1))]$ is the expected Bhattacharyya distance between the predictions of the two models on the unlabeled data, and c is a constant defining the relative weight of the unlabeled data within the learning process.

$$Min[L_1(\theta_1) + L_2(\theta_2) + cE_u[B(p_1(\theta_1), p_2(\theta_1))]]. \tag{3.5}$$

The algorithm is discussed in more detail later in section 6.2.1. For all the two-view methods, they weighted the total labeled data equally with the total unlabeled data. They regularized the Maximum Entropy classifiers with a unit variance Gaussian prior. Out of the 12 transfer learning tasks, their method performed best in 6 cases and although not reaching the best accuracy result of 86.8% for SCL (Blitzer *et al.* (2007)), SAR performed best in most of the test cases. The best result obtained by SAR reached accuracy of 85.8 % for $electronics \rightarrow kitchen$ domain transfer classification task.

Finally, Wan (2009) proposed a co-training approach to improve the classification accuracy of polarity identification of Chinese product reviews. They used an annotated English corpus for sentiment polarity identification of Chinese reviews in a supervised framework, without using any Chinese resources. First, machine translation services were used to translate English training

reviews into Chinese reviews and also translate Chinese test reviews and additional unlabeled reviews into English reviews. Then, the classification problem could be viewed as two independent views: Chinese view with only Chinese features and English view with only English features. They then used the co-training approach to learn two classifiers and finally the two classifiers were combined into a single sentiment classifier. In the classification phase, each unlabeled Chinese review is first translated into an English review and the learned classifier is applied to give either positive or negative class. These steps are illustrated in Figure 3.1. A SVM classifier is adopted as the basic classifier in the proposed approach. Experimental results show that the proposed approach can outperform the baseline inductive classifiers and more advanced transductive classifiers, by reaching accuracy of 81.3%. They also examined the influence of the number of iteration, feature size and growth size at each iteration. The proposed co-training approach can outperform the best baseline after 20 iterations. After a large number of iterations, the performance of the co-training approach decreased because noisy training examples might be selected from the remaining unlabeled set. Finally, the performance of the approach did not change any more as the algorithm ran out of all possible examples in the unlabeled set. About the growth size, they examined that the performance of the co-training approach with the balanced growth ($p = n$ in Algorithm 3) could be improved after a few iterations. And the performance of the co-training approach with large p and n would more quickly become unchanged, because the approach would run out of the limited examples in the unlabeled set more quickly. However, the performance of the co-training approaches with two unbalanced growths always went down quite rapidly, because the labeled unbalanced examples hurt the performance badly at each iteration. They also showed that the feature selection technique slightly influenced on the classification accuracy of the methods. It can be seen that the co-training approach can always outperform the two baselines (TSVM(ENCN1)[1] and TSVM(ENCN2)[2]) with different feature sizes. The co-training algorithm they used is illustrated in algorithm 3.

3.4 Our Proposal

Having presented the main relevant approaches that we have identified as being closely related to our work, we finish this chapter by highlighting the main issues of our work, which are explained in detail in the following chapters. Unlike all proposed methods so far, our approach aims at taking advantage of different view levels. We propose to combine high-level features (e.g. level of affective words, level of abstraction of nouns) and low-level features (e.g. unigrams, bigrams) to learn models of subjectivity, which may apply to different domains. For that purpose, we propose different methods based on the multi-view learning over two and

[1]This method applies the transductive SVM with both English and Chinese features for sentiment classification in the two views.

[2]This method combines the results of TSVM(EN) and TSVM(CN) by averaging the prediction values.

Algorithm 3 The co-training algorithm for cross-lingual sentiment classification.

Input: F_{en} and F_{cn} are redundantly sufficient sets of features, where F_{en} represents the English features, F_{cn} represents the Chinese features;
- L is a set of labeled training reviews;
- U is a set of unlabeled reviews;

for I iterations **do**
  Learn the first classifier C_{en} from L based on F_{en};
  Use C_{en} to label reviews from U based on F_{en};
  Choose p positive and n negative the most confidently predicted reviews E_{en} from U;
  Learn the second classifier C_{cn} from L based on F_{cn};
  Use C_{cn} to label reviews from U based on F_{cn};
  Choose p positive and n negative the most confidently predicted reviews E_{cn} from U;
  Removes reviews $E_{en} \bigcup E_{cn}$ from U
  Add reviews $E_{en} \bigcup E_{cn}$ with the corresponding labels to L
end for

three views and join two different classifiers LDA and SVM to maximize the optimality of the approach. Our methodology uses state-of-the-art characteristics that have been used to classify opinionated texts and proposes a new feature to classify sentiment texts, based on the level of abstraction of nouns. Another important contribution in this area (in order to reach language-independence) is the automatic construction of labeled data sets. Indeed, supervised classification techniques require large amounts of labeled training data. However, the acquisition of these data can be time-consuming and expensive. Based on that assumption, we propose to automatically produce learning data from Web resources. In particular, we compare Wikipedia and Weblogs texts to reference objective and subjective corpora and conclude that Wikipedia texts convey objective messages while Weblogs present subjective contents.

The next chapter starts the analysis of the automatically built large data sets of learning examples based on common sense judgments.

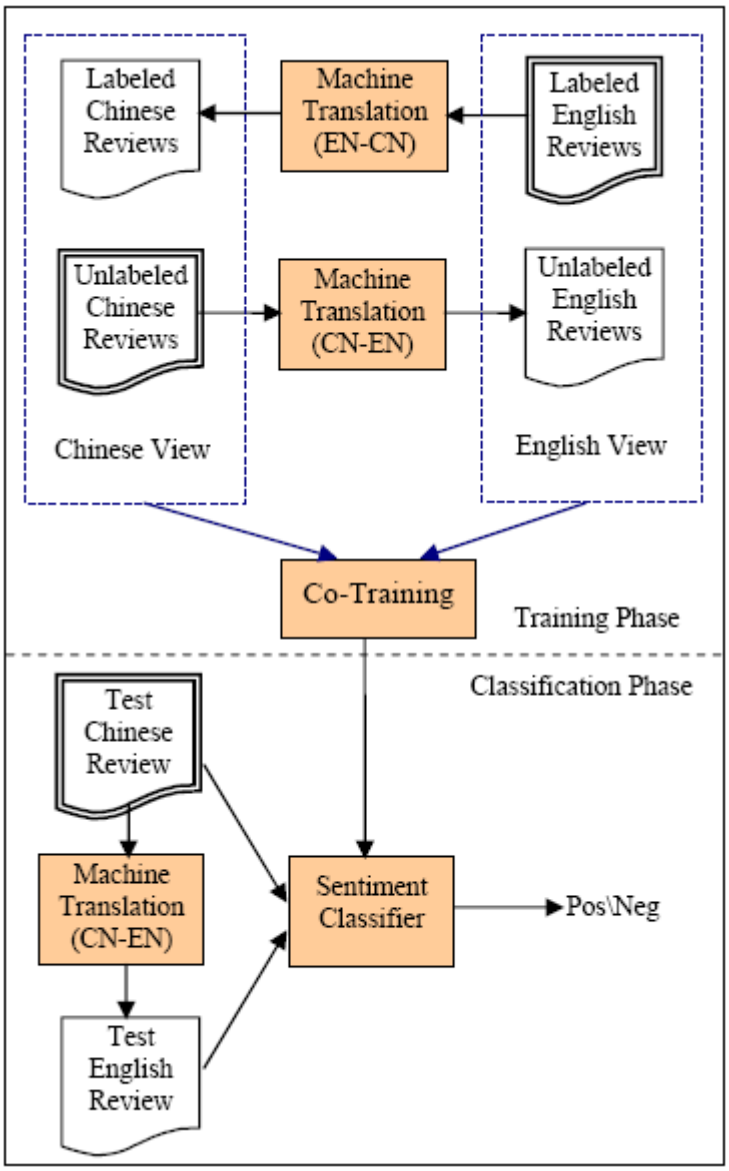

Figure 3.1: Framework of the approach proposed in Wan (2009).

Chapter 4

Resources for Sentiment Analysis

> "*You guys are both saying the same thing. The only reason you're arguing is because you're using different words.*"
>
> **S. I. Hayakawa**

Opinion-annotated training data are obviously required to create and evaluate a supervised classifier, which is built by learning the characteristics of pre-labeled data. Even though an annotated corpus is not always necessary to extract opinion-bearing features, it is a requisite for empirical evaluation of those opinion-bearing features. When creating an opinion training set, manual labeling is relatively accurate but labor-intensive and can be performed only on small amounts of data. Moreover, most training sets are built for English and little has been done for other languages. Based on that assumption, we propose to automatically produce learning data from Web resources. To do so, we propose to compare Wikipedia and Weblogs texts to reference objective and subjective corpora. This work has been carried out in collaboration with Sebastião Pais from the University of Beira Interior. So, for the experiments in this thesis, we used three manually annotated standard corpora (the MPQA, the subjectivity dataset v1.0 and documents used in Chesley *et al*. (2006)) and one corpus automatically built from Wikipedia texts and Weblogs.

4.1 Existing Resources for Sentiment Classification

Usually, the best results in sentiment classification task are achieved by supervised methods, in which a learner is trained on a set of manually labeled examples. The rationale behind the use of manually labeled data is that the occurrence of good features should be a part of opinion expressions that have been manually tagged in the training data. So, in this sub-section, we will describe the three manually annotated standard corpora used in our experiments.

4.1.1 The MPQA Dataset ($\{MPQA\}$)

The first data set is provided by the news articles given by the Multi-Perspective Question Answering (MPQA) Opinion Corpus[1]. This corpus has been used for sentiment analysis by different research groups (Mihalcea & Banea (2007), Wilson *et al.* (2006)). The MPQA contains 10 657 sentences in 535 documents from the world press on a variety of topics. All documents in the collection are marked with expression-level opinion annotations. The documents are from 187 different news sources in a variety of countries and date from June 2001 to May 2002. The corpus has been collected and manually annotated with respect to subjectivity as part of the summer 2002 NRRC Workshop on Multi-Perspective Question Answering 2005. The annotations were at expression (subsentence) level, but the subjective annotations at sentence and document level can be derived from these annotations. The annotation scheme contains two main components: a type of an explicit private state and speech event, and a type of an expressive subjective element. Several detailed attributes and strengths are annotated as well. We used the sentence-level annotations associated as follows: a sentence is labeled as subjective if it contains only subjective phrases and respectively, it is labeled as objective if it contains only objective phrases. From the approximately 9700 sentences in this corpus, we labeled 60% of them as subjective, while only 11% were objective (the dimensions of the corpora are given in Table 4.1). Based on the work done by Pang & Lee (2004) who proposed to classify texts based only on their subjective/objective parts, we built a corpus of 100 objective texts and 100 subjective texts by randomly selecting sentences containing only subjective or objective phrases (without overlap of sentences). This case represents the "ideal" case where all the sentences in texts are either subjective or objective. Examples of sentences, extracted from the MPQA corpus and labeled as subjective or objective are presented bellow. The first two examples are labeled as subjective.

(1) *President Hugo Chavez is once again at the helm of power, and the military has returned to its barracks.*

(2) *When he revisited the Great Wall, he could hardly hide his excitement.*

The next two are labeled as objective.

(3) *As a result, at least four Palestinians were reportedly killed and more than 30 wounded.*

(4) *Officials said 19 crew members were plucked up from the sea and one died later.*

Table 4.1: Dimensions of the $\{MPQA\}$ corpora.

Corpora	MQPA Subjective	MPQA Objective
Unique Sentences	6 363	1 100
Unique Words	10 080	4 200

[1] http://www.cs.pitt.edu/mpqa/[$16^{th} November, 2010$].

4.1.2 The Subjectivity Dataset v1.0 ($\{RIMDB\}$)

The second corpus is the subjectivity dataset v1.0[1], which contains 5000 subjective and 5000 objective sentences collected from movie reviews data (Pang & Lee (2004)). To gather subjective sentences, Pang & Lee (2004) collected 5000 movie review snippets from the Rottentomatoes Website[2] (e.g., "imaginative, and impossible to resist"). To obtain (mostly) objective data, they took 5000 sentences from plot summaries available from the Internet Movie Database[3]. The corpus is summarized in Table 4.2. The assumption is that all the snippets from the Rotten Tomatoes pages are subjective as they come from a review site, while all the sentences from IMDB are objective as they focus on movie plot descriptions. They only selected sentences or snippets with at least ten words long and drawn from reviews or plot summaries of movies released post-2001, thus preventing overlapping of their polarity dataset. Similarly to what we did for the MPQA corpus, we built a corpus of 100 objective texts and 100 subjective texts by randomly selecting 50 sentences containing only subjective or objective phrases (without overlap of sentences). Examples of subjective and objective sentences from the subjectivity dataset are presented below.The first two examples are labeled as subjective.

(1) *it is not a mass-market entertainment but an uncompromising attempt by one artist to think about another.*

(2) *the script is a tired one , with few moments of joy rising above the stale material.*

The next two are labeled as objective.

(3) *the movie begins in the past where a young boy named sam attempts to save celebi from a hunter.*

(4) *david is a painter with painter's block who takes a job as a waiter to get some inspiration.*

Table 4.2: Dimensions of the $\{RIMDB\}$ corpora.

Corpora	Rotten Subjective	IMDB Objective
Unique Sentences	5 000	5 000
Unique Words	11 900	12 013

4.1.3 The CHESLEY Dataset ($\{CHES\}$)

Chesley *et al.* (2006) gathered a dataset[4] (CHES) of objective and subjective documents via RSS[5] and Atom Web[6] syndication feeds. It contains 496 subjective and 580 objective documents. Objective feeds are from sites providing content such as world and national news (CNN, NPR, etc.), local news (Atlanta Journal and Constitution, Seattle Post-Intelligencer, etc.) and various sites

[1] http://www.cs.cornell.edu/People/pabo/movie-review-data/[16th $November, 2010$].
[2] http://www.rottentomatoes.com[16th $November, 2010$].
[3] http://www.imdb.com[16th $November, 2010$].
[4] http://www.tc.umn.edu/ ches0045/data/[16th $November, 2010$].
[5] http://www.rssboard.org[16th $November, 2010$].
[6] http://www.atomenabled.org[16th $November, 2010$].

focusing on topics such as health, science, business and technology. Subjective feeds include content from newspaper columns (Charles Krauthammer, E. J. Dionne, etc.), letters to the editor (Washington Post, Boston Globe, etc.), reviews (dvdverdict.com, rottentomatoes.com, etc.), and political blogs (Powerline, Huffington Post, etc.). Then, they manually verified each document to confirm they strongly corresponded to objective, positive, or negative categorization. In subjective cases, a further judgment was made to verify a sufficiently positive or negative polarization. Documents not belonging to any of these categories were discarded, and retained documents were grouped into three data sets. While documents from the objective feeds were almost always verified as such, this was unsurprisingly much less the case for positive and negative documents from subjective feeds. The average rate of document retention from subjective feeds was approximately 19%, which illustrated the difficulty of obtaining quality subjective data. Finally, given varying degrees of sentiment expression in the feeds, they decided to include only documents of very clear polarity in the training data. They considered documents that were categorically positive, negative, or objective. For subjective texts, this meant that the posts only expressed one opinion (positive or negative). For objective texts, this meant that the only goal of the author was to inform, and not to offer any kind of opinion. Finally, they obtained 263 positively oriented documents, 233 negatively oriented documents, and 580 objective documents. The corpus is summarized in Table 4.3. The following are examples of subjective texts.
*"WARNING: this post is written in anger, therefore it contains tons of curse words. If you are easily offended by curse words, please read no further.*****What..the FUCK! America has been hit by our biggest natural disaster and our worst terrorist attack in the past 4 years. The poverty rate is climbing; racial tensions are at all time high; America is in the midst of wars with no end in sight; thousands of our boys have been killed; and what is the Republican solution? ...".*

The next are examples of objective texts.
"Despite having tobacco juice spit in her face last Tuesday, Jane Fonda has continued to promote My Life So Far with appearances at a number of bookstores. Two independent booksellers who hosted events shortly after the "spit and run" incident at the book signing at Rainy Day Books in Kansas City April 19 reported no problems, and Random House says there are no plans to alter Fonda's tour...".

Table 4.3: Dimensions of the $\{CHES\}$ corpora.

Corpora	CHES Subjective	CHES Objective
Unique Sentences	3 228	3 803
Unique Words	8 572	10 300

4.2 Automatic Construction of Labeled Dataset ($\{WBLOG\}$)

Until now, most of the supervised and semi-supervised methodologies, that have been proposed to learn subjective characteristics of texts are based on a limited set of learning examples almost exclusively for the English[1] language. In fact, most learning data sets are manually collected and labeled, and therefore they are usually small and do not cover most characteristics of subjective language. To deal with this problem, some authors Wiebe *et al.* (2004) and Chesley *et al.* (2006) proposed to analyze texts, which should express opinions (e.g. letters to the editor, news columns, reviews, political Weblogs) and facts (e.g. world, national and local news) by definition. As such, only manual post-editing is necessary. Although this is the most popular approach, there is no clear theoretical background and the characterization of subjectivity and objectivity is defined upon common sense beliefs. So, we followed this idea, but based on a more theoretical background. As such, Pais (2007) proposed the assumption that Wikipedia texts are representative of objectivity and Weblogs are representative of subjectivity. But, somehow, this needs to be proved. As a consequence, we proposed to compare Wikipedia texts and Weblogs to a reference sentiment (subjective/objective) corpus, the subjectivity v1.0 corpus[2] built by Pang & Lee (2004), in order to confirm our hypothesis (Dias (2010)). Therefore, we proposed an exhaustive evaluation based on (1) the Rocchio classification method (Rocchio (1971)) for different part-of-speech tag levels and (2) language modeling.

In order to have a more complete view about subjectivity and objectivity in language, we collected large quantities of texts from Weblogs and Wikipedia. The English static version of Wikipedia [3] was downloaded first and all the sentences were extracted, giving rise to a corpus of 40 Gb. Then, Weblogs domains from different topics, which can be found in Pais (2007) were crawled, gathered 12 Gb of Weblogs text sentences. The corpus is summarized in Table 4.4. In comparison, the subjectivity v1.0 corpus contains 5 000 objective sentences collected from plot summaries available at the Internet Movie Database www.imdb.com. and 5 000 subjective sentences gathered from movie reviews available at www.rottentomatoes.com. (Table 4.2). So, a random sample of both Wikipedia and Weblogs data sets was used, in order to afford an impartial comparison, maintaining statistical significance.

Table 4.4: Dimensions of the $\{WBLOG\}$ corpora.

Corpora	Wikipedia	Weblogs
Unique Sentences	411 293	984 682
Unique Words	224 112	79 680

[1]With a few exceptions as proposed in (Mihalcea & Banea (2007), Banea *et al.* (2008)) for Romanian and Spanish, and (Wan (2009)) for Chinese.

[2]http://www.cs.cornell.edu/people/Pabo/movie-review-data [$16^{th} November, 2010$].

[3]http://download.wikimedia.org/enwiki/20071018 [$6^{th} September, 2007$].

In order to verify our initial assumptions, we first applied the simple Rocchio relevance feedback algorithm adapted for text classification (Rocchio (1971)). Rocchio relevance feedback algorithm is one of the most widely applied learning algorithms for text categorization. It uses standard $tf.idf$ weighted vectors to represent text documents. For each category, it calculates a prototype vector by summing the vectors of the training documents in the same category. Finally, the closest prototype vector in terms of cosine similarity measure (see Equation 4.1) with any given text classifies the text.

$$cos(X_i, X_j) = \frac{\sum_{k=1}^{p} X_{ik} \times X_{jk}}{\sqrt{\sum_{k=1}^{p} X_{ik}^2} \times \sqrt{\sum_{k=1}^{p} X_{jk}^2}}. \tag{4.1}$$

Within the context of our work, instead of the $tf.idf$, we used the $tf.isf$ measure, because we deal with classified sentences and not documents and an evaluation at different part-of-speech tag levels was performed. Results are presented in table 4.5, where the test vector is the set of Wikipedia sentences and the trained vectors are the subjective and objective sentences from the subjectivity v1.0 corpus. Our initial assumption that texts from Wikipedia are representative of objectivity was confirmed, although the role of verbs seems less clear with respect to subjectivity as opposed to what is stated in Chesley *et al.* (2006). Similarly, the same experiment was performed where the test vector is the set of Weblogs sentences and the trained vectors are the subjective and objective sentences from the subjectivity v1.0 corpus. Results presented in Table 4.6 clearly show that at any part-of-speech level, Weblogs embody subjectivity. Finally, in order to confirm the assumptions formulated in Wiebe *et al.* (2004) and Chesley *et al.* (2006) without any support, we proposed to test the extent to which news article convey an objective language. For that purpose, a statistically significant random sample of the Reuters corpus[1] was extracted, and classification with the Rocchio algorithm was performed . Results presented in Table 4.7 confirm common sense judgments made by Wiebe *et al.* (2004) and Chesley *et al.* (2006), although verbs still seem to cause some confusions.

Table 4.5: Results with the Wikipedia test data set.

Part-of-speech level	Subjective	Objective	Class
All Words	0.76	0.79	Objective
All ADJ	0.54	0.61	Objective
All V	0.71	0.67	Subjective
All N	0.66	0.69	Objective
All ADJ + All V	0.65	0.66	Objective
All ADJ + All N	0.65	0.68	Objective
All N + All V	0.70	0.69	Subjective
All ADJ + All N + All V	0.68	0.69	Objective

[1] http://trec.nist.gov/data/reuters/reuters.html [$16^{th} November, 2010$].

Table 4.6: Results with the Weblogs test data set.

Part-of-speech level	Subjective	Objective	Class
All Words	0.60	0.56	Subjective
All ADJ	0.52	0.49	Subjective
All V	0.53	0.48	Subjective
All N	0.47	0.43	Subjective
All ADJ + All V	0.49	0.48	Subjective
All ADJ + All N	0.48	0.44	Subjective
All N + All V	0.50	0.45	Subjective
All ADJ + All N + All V	0.47	0.46	Subjective

Table 4.7: Results with the Reuters test data set.

Part-of-speech level	Subjective	Objective	Class
All Words	0.64	0.68	Objective
All ADJ	0.30	0.40	Objective
All V	0.38	0.37	Subjective
All N	0.34	0.47	Objective
All ADJ + All V	0.36	0.38	Objective
All ADJ + All N	0.35	0.49	Objective
All N + All V	0.36	0.47	Objective
All ADJ + All N + All V	0.37	0.47	Objective

In spite of encouraging classifications, the values of the cosine similarity measure within the same morphological level between the trained and the test vectors are usually very close. This does not provide much confidence in the results. For that purpose, we proposed another methodology based on language modeling. The basic idea is that objective and subjective languages are intrinsically different. Consequently, if we build a language model based on Weblogs, the subjective part of the subjectivity v1.0 corpus should be more probable than the objective part, and vice and versa[1]. This probability is transformed into perplexity (Px) and entropy (H) measures within the CMU-Toolkit[2]. The results of this experiment are given in Table 4.8 for a trigram language model.

To summarize the results in Table 4.8, the trained model Wikipedia shows lower perplexity and entropy for the objective sentences than for the subjective sentences. When using the trained model Weblogs, quite the opposite happens . Lower perplexity and entropy are shown

[1] As language models need large quantities of texts, they are built from the Wikipedia, Weblogs and Reuters corpora in our experiments. In fact, evaluation is done the other way round.

[2] http://www.speech.cs.cmu.edu/SLM_info.html [$16^{th} November, 2010$].

Table 4.8: Results obtained with the Language Model.

	Wikipedia	Weblogs	Reuters
Objective	$Px = 691.27$	$Px = 2027.06$	$Px = 1104.03$
	$H = 9.43$	$H = 10.99$	$H = 10.11$
Subjective	$Px = 880.67$	$Px = 1991.09$	$Px = 1226.34$
	$H = 9.75$	$H = 10.96$	$H = 10.26$

for the subjective sentences in that case, than for the objective sentences. Once again, our assumptions are confirmed as objective (resp. subjective) sentences are intrinsically closer to the Wikipedia (resp. Weblogs) model than subjective (resp. objective) ones. Furthermore, the lower the perplexity and the entropy are, the closer to the model the sentences are. Finally, the trained model Reuters shows similar behavior as the Wikipedia model, thus validating the common sense judgments made by Wiebe *et al.* (2004) and Chesley *et al.* (2006).

4.3 Summary

In this chapter, we have presented three existing resources and a corpus based on Web resources and automatically annotated. We proved that there is a great similarity between Wikipedia sentences and objective sentences as well as between Weblogs sentences and subjective sentences. For that purpose, we used two different methodologies: the Rocchio Method and the Language Model. Thanks to this analysis, we are now able to automatically build large data sets of learning examples based on common sense judgments. Moreover, Wikipedia texts exist in many languages as well as Weblogs, which emerge everyday all over the world. As a consequence, multilingual sentiment data sets can easily be compiled and new studies may rise to reach multilingual sentiment analysis based on corpus analysis as show in Rodrigues (2009) where they learn a subjective lexicon for the Portuguese language, without linguistic tools or resources as in Mihalcea & Banea (2007) and Banea *et al.* (2008).

In the following chapter, we propose to study the characteristics for analyzing subjective content in documents. For that purpose, we first study some state-of-the-art features used in learning models and then present a new relevant characteristic based on level of abstraction of nouns. In order to evaluate to what extent the given set of characteristics are discriminative and allow representing distinctively the datasets, we propose to do feature selection by applying the Wilcoxon rank-sum test and to visualize the datasets using multidimensional scaling.

Chapter 5

Feature Selection and Visualization

> "*One accurate measurement is worth a thousand expert opinions.*"
>
> **Admiral Grace Hopper**

The determination of important opinion-bearing features lies at the core of sentiment classification. Opinion-bearing features can be extracted through statistical methods based on term presence or frequency in the training set or can be selected through linguistic methods based on word attributes, syntactic relationships or other linguistic characteristics. In this chapter, we will first discuss some important opinion-bearing features that have been examined in related works and will then propose a new feature based on the level of abstraction of nouns. Finally, we propose two feature selection techniques in order to evaluate the extent to which the given set of features allows to distinguish subjective texts from objective ones.

5.1 Feature Definition

A few researches dealt with cross-domain subjectivity classification and all argued that it is hard to learn a domain-independent classifier. One possible approach is to train the classifier on a domain-mixed set of data instead of training it on one specific domain (Aue & Gamon (2005)). Another possibility is to propose high-level features, which do not depend so much on topics such as part-of-speech statistics (Finn & Kushmerick (2006)). As a consequence, we proposed in Lambov *et al.* (2009a) a methodology, which aims at classifying texts at the subjectivity level (i.e. subjective vs. objective) based on high-level semantic features, which can apply to different domains. On the other hand, in particular domains, words in the texts are excellent features for sentiment classification (Pang *et al.* (2002)). Therefore, some new techniques have been proposed to leverage the problem of using bag-of-words features for cross-domain classification task (Ganchev *et al.* (2008), Blitzer *et al.* (2007)). Within this context, we have been working on the development of cross-domain subjectivity/objectivity classifiers based on low-level features (e.g. unigrams and bigrams) and high-level features (e.g. level of abstraction of words). So, in this section we list the features we used in our experiments. First, we employ

"classic" text analysis features - features used in other sentiment classification tasks. We study some well-known state-of-the-art characteristics and propose a new one based on the level of abstraction of nouns.

5.1.1 High-Level Features

Users often use some different words when they express sentiment in different domains. If we apply directly a classifier trained in one domain to other domains, the performance will be very low due to the differences between these domains as shown in related work. Therefore, we propose to use the following high-level semantic features, which can easily apply to different domains.

5.1.1.1 Intensity of Affective Words

Sentiment expressions mainly depend on some words, which can express subjective sentiment orientation. Therefore many approaches are based on the use of Sentiment Lexicons. In these methods words are tagged with their positive/negative prior polarity and a final opinion score is calculated on behalf of polarities of the words (Missen & Boughanem (2009)). Within this scope, Strapparava & Mihalcea (2008) used a set of words extracted from the WordNet Affect lexicon proposed in Strapparava & Valitutti (2004). This is an extension of WordNet (Miller (1990)), including a subset of synsets suitable to represent affective concepts. For example, the WordNet Affect lexicon encodes the fact that both "horror" and "hysteria" express negative fear as well as that "enthusiastic" denotes positive emotion and "glad" refers to joy (some examples are given in Table 5.1). As they are based on the classical WordNet, most of these words are occurrences of general language. As such, the level of affective words in texts may successfully express subjectivity across domains. So, we propose to evaluate the level of affective words in texts as shown in Equation 5.1.

$$K_1(d_j) = \frac{total\ affective\ words\ in\ d_j}{total\ words\ in\ d_j}. \tag{5.1}$$

Table 5.1: Examples for Affective words.

Affective Category	Words
fury	furious, maddened, enraged, angered
distress	worrying, disturbed, upset, worried, unhappy
stupefaction	stupid, dazed, stun, baffle, amaze
disgust	disgust, revolt, repel, sicken, wicked

5.1.1.2 Dynamic, Gradable and Semantically Oriented Adjectives

Adjectives are often connected to the expression of attitudes and reported to have a positive and statistically significant correlation with subjectivity (Bruce & Wiebe (1999); Wiebe *et al.* (1999); Missen & Boughanem (2009); Missen *et al.* (2009)). Missen *et al.* (2009) used adverbs and adjectives to represent the emotiveness of a document. They calculated emotiveness of a document by counting the numbers of adverbs and adjectives, assuming that an important clue of subjectiveness of a document. Wiebe *et al.* (2004) reported the difference between the proportion of selected adjectives in opinion documents and non-opinion documents as significant, where the selection criterion (distributional similarity) chooses adjectives used in a similar way as those used in pre-labeled opinionated pieces. Hatzivassiloglou & Wiebe (2000) noted that dynamic and semantic-oriented adjectives were found to be stronger subjective cues than adjectives as a whole, as described in Chapter 3. In particular, dynamic adjectives are adjectives with the "qualities that are thought to be subject to control by the possessor and hence can be restricted temporally" (Quirk *et al.* (1985)). Bruce & Wiebe (1999) assumed this stative/dynamic distinction between adjectives to be related to subjectivity and manually identified a list of 124 dynamic adjectives from about 500 sentences, which were more subjective than the rest of the adjectives in the Wall Street Journal Treebank Corpus.

In the same domain but using a different corpus, Hatzivassiloglou & Wiebe (2000) confirmed the strong correlation between dynamic adjectives and subjectivity with more than 30% improvement in precision over adjectives as a whole. They suggested that another type of adjectives, gradable adjectives, were also useful opinion indicators that had 13-21% higher precision than adjectives as a whole. Gradable adjectives are those that can participate in comparative constructs (e.g., "This movie is more exciting than the other") and accept modifying expressions that act as intensifiers (e.g., "This game is very exciting", where "very" is an intensive modifier). The third type of adjective is more intuitive as opinion evidence. Semantic-oriented adjectives are polar words that are either positive or negative. Adjectives with polarity, such as "good", "bad", or "beautiful", are inherently connected to opinions (see Table 5.2). Hatzivassiloglou & Wiebe (2000) showed that a reasonably high accuracy of either opinion detection or polarity detection could be achieved by using polar adjectives alone. Whitelaw *et al.* (2005) also showed that a reasonably high accuracy of either opinion detection or polarity detection could be achieved by using polar adjectives. They used semi-automated methods to build a lexicon of appraisal adjectives and their modifiers and classified movie reviews using features based on these taxonomies combined with standard bag-of-words features and reported the accuracy of 90.2%. Finally Chesley *et al.* (2006) also found that positive adjectives played a major role in classifying opinionated blog posts. As a consequence, we used the proportion of these adjectives in texts to characterize their subjectivity level as shown in Equation 5.2, 5.3 and 5.4. For that purpose, we utilized the set of all dynamic and gradable adjectives manually identified in

Hatzivassiloglou & Wiebe (2000) and the set of semantic orientation labels assigned as in Hatzivassiloglou & McKeown (1997).

$$K_{21}(d_j) = \frac{total\ dynamic\ adjectives\ in\ d_j}{total\ adjectives\ in\ d_j}. \tag{5.2}$$

$$K_{22}(d_j) = \frac{total\ gradable\ adjectives\ in\ d_j}{total\ adjectives\ in\ d_j}. \tag{5.3}$$

$$K_{23}(d_j) = \frac{total\ semantic\ adjectives\ in\ d_j}{total\ adjectives\ in\ d_j}. \tag{5.4}$$

Table 5.2: Examples for Dynamic, Gradable and Semantically Oriented Adjectives.

Affective Category	Words
Dynamic Adjectives	abusive, careful, clever, foolish, brave
Gradable Adjectives	other, good, new, many, high, military
Semantically Oriented Adjectives	abnormal, banal, attractive, boring

5.1.1.3 Classes of Verbs

Although less significant than adjectives, verbs are also found to be good indicators of opinion information. Wiebe *et al.* (2004) showed that precision of opinionated verbs was consistently higher than the baseline, which took into account all the words in opinion pieces. The difference between the proportion of instances of those verbs in opinion documents and non-opinion documents was significant ($p < 0.001, z >= 4.1$). Chesley *et al.* (2006) also presented a method using verb class information. Their verb classes expressed objectivity and polarity. To obtain relevant verb classes, they used InfoXtract (Srihari *et al.* (2006)), an automatic text analyzer, which groups verbs according to classes that often correspond to their polarity. They found that the "asserting" and "approving" verb classes played a key role in improving the accuracy of classifying blog posts as positive opinion. As InfoXtract is not freely available, we reproduced their methodology by using the classification of verbs available in Levin's English verb classes and alternations (Levin (1993)). There are 193 Levin's verb classes, which are grouped into 51 sections with two further levels of subsections. A common way to use these verb classes is to use as opinion-bearing feature verbs from an opinion related Levin's verb class such as positive judgment (e.g., "bless","excuse") or marvel (e.g., "anger","fear"). So, we proposed to evaluate the proportion of each corresponding class of verbs (i.e. conjecture verbs, marvel verbs, see verbs and positive verbs) as an interesting clue to identify subjectivity in texts. In Table 5.3

we present a list of corresponding verbs for each Levin's class and the corresponding features in Equation 5.5, 5.6, 5.7 and 5.8.

Table 5.3: Verb Examples for Levin's verb classes.

Class Verb	Verbs
Conjecture	admit, allow, deny, guess, show, suspect, assert, guarantee
See	detect, see, feel, smell, taste, sense, notice, hear, discern
Marvel	anger, fear, cry, care, bleed, bother, glory, heart, obsess, suffer
Positive Judgment	bless, excuse, forgive, greed, thank, reward, compliment, pardon

$$K_{31}(d_j) = \frac{total\ Conjecture\ verbs\ in\ d_j}{total\ verbs\ in\ d_j}. \tag{5.5}$$

$$K_{32}(d_j) = \frac{total\ See\ verbs\ in\ d_j}{total\ verbs\ in\ d_j}. \tag{5.6}$$

$$K_{33}(d_j) = \frac{total\ Marvel\ verbs\ in\ d_j}{total\ verbs\ in\ d_j}. \tag{5.7}$$

$$K_{34}(d_j) = \frac{total\ Positive\ Judgment\ verbs\ in\ d_j}{total\ verbs\ in\ d_j}. \tag{5.8}$$

5.1.1.4 Level of Abstraction of Nouns

There exists linguistic evidence that the overall level of generality is a characteristic of opinionated texts, i.e. subjectivity is usually expressed in more abstract terms than objectivity (Osgood *et al.* (1971), Boiy *et al.* (2007)). Indeed, descriptive texts tend to be more precise and more objective, hence more specific. Usually, abstract and general terms represent ideas or concepts (such as "justice" or "hatred"), while concrete words refer to things we can touch, see, hear, smell, measure and taste, such as "sandpaper", "soda", "birch trees", "smog", "cow", "sailboat", "chair" and "pancake". By using language at a lower level of abstraction, in more concrete and specific vocabulary, each word carries more information. For example, in technical and business writing, the purpose is to convey information clearly and accurately and therefore authors use more specific words instead of general words. In particular, Boiy *et al.* (2007) define specificity as *the extent to which a conceptualized object is referred to by name in a direct and clear way; or is only implied, suggested, alluded to, generalized, or otherwise hinted at*. In other words, a word is abstract when it has few distinctive features and few attributes that can be pictured in the mind (Osgood *et al.* (1971)). For example, a person having concrete thinking

will describe the Statue of Liberty as a statue of a lady with a torch. A person with abstract thinking will describe the Statue of Liberty as a symbol of liberty and freedom.

So, one way of measuring the abstractness of a word is by using the hypernym relation in WordNet. In particular, a hypernym metric can easily be defined as the path length to the root via the hypernym relation. For example, "chair" (as a seat) has 7 hypernym levels: $chair \Rightarrow furniture \Rightarrow furnishings \Rightarrow instrumentality \Rightarrow artifact \Rightarrow object \Rightarrow entity$. So, a word having more hypernym levels is more likely to be concrete than one with fewer levels. Thus, the average hypernym level of all nouns in a given text may provide a good clue for sentiment classification. So, we propose to evaluate the hypernym levels of all the nouns in texts as shown in Equation 5.9.

$$K_4(d_j) = \frac{total\ hypernym\ levels\ for\ nouns\ in\ d_j}{total\ nouns\ in\ d_j}. \tag{5.9}$$

Calculating the level of abstraction of nouns should be preceded by word sense disambiguation. Indeed, it is important that the correct sense is taken as a seed for the calculation of the hypernym level in WordNet. However, in practice, taking the most common sense of each word gives similar results as taking all the senses on average, as shown in section 5.2.1.

5.1.2 Low-Level Features

The most common set of features used for text classification is information regarding the occurrences of words or word n-grams in texts. Most of the text classification systems treat documents as simple bags-of-words and use the word presence or counts as features. Pang *et al.* (2002) found that (surprisingly) unigrams beat other features in their evaluations. Similarly, Dave *et al.* (2003) experimentally showed that trigrams and higher failed to show consistent improvement. Using a basic text classifier with only length-normalized unigrams, Ng *et al.* (2006) achieved an accuracy rate as high as 99.8% on movie reviews and 97% on book reviews in document-level opinion identification.

In this study, we consider texts as bags-of-words of lemmatized unigrams or lemmatized bigrams for which we compute their $tf.idf$ (Salton *et al.* (1975)) weights as in Equation 5.10, where tf_{ij} is the normalized frequency of term j in document i, N is the total number of documents in the collection, and n is the number of documents where the term j occurs at least once.

One problem with the bag-of-words feature is that when people talk about different topics, they tend to use different vocabularies. So, in most cases, differences in vocabulary are not sufficient

to distinguish opinions from non-opinions texts for cross-domain sentiment classification task.

$$tf.idfij = tf_{ij} * \log_2 \frac{N}{n}. \tag{5.10}$$

Once we have defined features, we need to analyze their usefulness for sentiment classification task. For that purpose, we used two feature selection techniques to evaluate the quality of opinion-bearing features that will be reported in the next subsections. First, we proposed to apply the Wilcoxon rank-sum test (Wilcoxon (1945)). Second, we performed a visual analysis of the distribution of the data sets in the space of our opinion-bearing features.

5.2 Feature Selection

Before performing any classification task, it is useful to evaluate the extent to which the given features are discriminative and allow distinctive presentation of the datasets in the given space of characteristics. Selecting salient opinion-bearing features is as important as generating these features. The main advantages include (1) reducing the bias caused by specific training data or resources by removing too specific features, (2) minimizing the computing cost and achieving similar or higher accuracy by using less or simpler features and (3) improving the accuracy of the models and the classification results.
While traditional or revised feature selections for text classification can be applied to simple bag-of-words or high order n-grams features, opinion specific feature evaluation methods are recommended for other opinion-bearing features. For that purpose, we propose to do feature selection by applying the Wilcoxon rank-sum test and to visualize the datasets using multidimensional scaling.

5.2.1 Wilcoxon Rank-Sum Test

The Wilcoxon rank-sum test is a nonparametric alternative to the two sample t-test used to compare the locations of two populations i.e. to determine if one population is shifted with respect to another. The tests lay the foundation for hypothesis testing of nonparametric statistics, used for population data that can be ranked but do not have numerical values, such as customer satisfaction or music reviews. It is one of the best-known non-parametric significance tests. The test essentially calculates the difference between each set of pairs and analyzes these differences. The Wilcoxon Rank Sum test can be used to test the null hypothesis that two populations have the same continuous distribution. The Wilcoxon signed rank test assumes that there is information in the magnitudes and signs of the differences between paired observations. As the non-parametric equivalent of the paired student's t-test, the Wilcoxon signed rank can be

used as an alternative to the t-test when the population data does not follow a normal distribution. This test takes the paired observations, calculates the differences, and ranks them from the smallest to the largest by absolute value. It then adds all the ranks associated to positive differences, giving the statistic. Finally, the p-value associated to this statistic is found from an appropriate table of all possible distributions of ranks. The Wilcoxon test is an R-estimate. So, we propose to use the Wilcoxon test to show that the comparative results are statistically significant. The two sample Wilcoxon test with one-sided alternative is carried out for all experiments. The samples contain 200 values for each of the sets (100 objective texts and 100 subjective texts) and the exact p-value is computed. Our null hypothesis is that the distribution of the subjective and objective samples is not significantly different. If the result of the test shows that the values of subjective (resp. objective) data are shifted to the right of the values of objective (resp. subjective) data for all of the sets, we can say that the distributions of subjective and objective data differ by a positive shift. The estimations regarding the differences of the location parameters are also computed for each of the sets. The exact 95% confidence interval for the difference in the location parameters of each set is obtained by the algorithm described in Bauer (1972) for which the Hodges-Lehmann estimator is employed. As a consequence, for each set, we are 95% confident that the interval contains the actual difference between the features values of subjective and objective texts. The observed results are consistent with the hypothesis that most of the high-level features exposed in section 5.1 (intensity of affective words, dynamic and semantically oriented adjectives, classes of verbs and level of abstraction of nouns) have good discriminative properties for subjectivity/objectivity identification. In most cases the computed p-value is lower than the significance level $\alpha = 0.05$, so we can reject the null hypothesis, and accept the alternative hypothesis that the distributions of the two samples are significantly different. As illustrated in Table 5.4, we can see that the level of positive verbs and the gradable adjective do not significantly separate the objective sample from the subjective one over training corpora. Regarding the verbs, these results are mainly due to the fact that "positive verbs" do not occur frequently in texts thus biasing the statistical test. On the other hand, gradable adjectives are a good predictor of subjectivity in the datasets, where all sentences in texts are either subjective or objective, but in the remaining two datasets, which represent more accurately the real texts, they do not perform well. This is mainly due to some gradable adjectives such as "many", "other", "new" or "high", which are very common in texts from Wikipedia and objective texts from Chesley's dataset. As a consequence, we discarded these two features from our classification task. We can also see that Chesley's dataset shows uncharacteristic behavior. For example, in this corpus "marvel verbs" also show no significant difference between subjective and objective examples. This is mainly due to the fact that many objective documents extracted from news reports contain verbs such as "approve" or "suffer" as illustrated in the following two sentences.

1. "Not to mention the $50 billion they just approved to continue funding the war in Iraq."
2. "Doctors in Germany have determined that patients suffering from early-stage Hodgkin's lymphoma can receive a reduced dose of involved field radiation therapy, combined with chemotherapy, and still retain a high survival rate, according to a study presented October 17, 2005, at the American Society for Therapeutic Radiology and Oncology's 47th Annual Meeting in Denver."

Table 5.4: Results of the Wilcoxon Rank Sum test for 95% confidence

	$\{MPQA\}$	$\{RIMDB\}$	$\{CHES\}$	$\{WBLOG\}$
Affective words	$< 0,0001$	$< 0,0001$	$< 0,0001$	$< 0,0001$
Dynamic adjectives	$< 0,0001$	$< 0,0001$	$0,014$	$< 0,0001$
Gradable adjectives	$0,0004$	$< 0,0001$	$0,74$	$0,38$
Semantic adjectives	$< 0,0001$	$< 0,0001$	$0,045$	$< 0,0001$
Conjecture verbs	$0,00024$	$< 0,0001$	$0,021$	$< 0,0001$
Marvel verbs	$< 0,0001$	$< 0,0001$	$0,44$	$< 0,0001$
See verbs	$< 0,0001$	$< 0,0001$	$0,006$	$< 0,0001$
Positive verbs	$< 0,0001$	$0,00011$	$0,075$	$0,00061$

As word sense disambiguation is not handled, we propose to use the Wilcoxon test for level of abstraction with the most frequent sense (Level of abstraction) of each noun and then with the average sense of each noun (Level of abstraction$_{avg}$), i.e. the hypernym levels of all the senses of each noun divided by its number of senses. In Table 5.5, we show the results for each experiment.

Table 5.5: Results of the Wilcoxon Rank Sum test for Level of abstraction

	$\{MPQA\}$	$\{RIMDB\}$	$\{CHES\}$	$\{WBLOG\}$
Level of abstraction	$0,003$	$< 0,0001$	$< 0,0001$	$< 0,0001$
Level of abstraction$_{avg}$	$< 0,0001$	$< 0,0001$	$< 0,016$	$< 0,023$

The observed results are consistent with the hypothesis that taking the most common sense of each word gives similar results as taking all the senses on average. The conclusion is that the level of abstraction of nouns in subjective and objective texts differs by a positive shift. As a consequence, in our experiments we will use as a feature the level of abstraction of the most common sense of each noun.

For many feature selection problems, a human defines the features that are potentially useful, and then a subset is chosen from the original pool of features using an automated feature selection algorithm. For supervised learning, the goal is to find the feature subset that best discovers natural groupings from data (also known as clustering). For this purpose, data visualization is an important means of extracting useful information from large quantities of raw data. In the next

subsection, we introduce Multidimensional Scaling, which incorporates visualization techniques to enable a deeper understanding of the data.

5.2.2 Multidimensional Scaling

The visual performance of the information has always been useful for demonstrations and comparison between different methods. Visualization also can be used in order to evaluate feature extraction techniques within an exploratory data analysis framework. So, in this subsection, we propose a visual analysis of the distribution of the datasets in the space of features. The goal of this study is to give a visual interpretation of the data distribution to assess how well classification may perform. If objective and subjective texts can be represented in a distinct way in the reduced space of features, one may expect good classification results. To perform this study, we use a Multidimensional Scaling (MDS) process, which is a traditional data analysis technique. MDS (Kruskal & Wish (1977)) allows to display the structure of distance-like data into an Euclidean space. Multidimensional Scaling (MDS) is a data analysis method, which is widely used in marketing and psychometrics. The aim of the method is to detect meaningful underlying dimensions that allow the researcher to explain observed similarities or dissimilarities (distances) between the investigated objects. To build an optimal representation, the MDS algorithm minimizes a criterion called stress. The closer the stress to zero, the better the representation. For instance, suppose we have a set of objects (e.g., a number of text documents) and the measure of the similarity between the objects is known. This measure, called proximity, indicates how similar or how dissimilar two objects are or are perceived to be. It can be obtained in different ways, e.g., by computing the correlation coefficient or Euclidean distance from the vector representation of the text documents. What MDS does is to map each object to a lower dimensional space and the distances between objects resemble to the original similarity information (i.e., the larger the dissimilarity between two objects, the further apart they should be in the lower dimensional space). This geometrical configuration of points reflects the hidden structure of the data and may help to make it easier to understand.

So, in the following subsections, we propose a visual analysis of the distribution of the datasets in the space of high-level and low level features. In order to evaluate the difference between high level features with low level ones, we performed comparative studies on our four test sets. Therefore, we performed the MDS process on all corpora trying to visualize subjective texts from objective ones. For the high level features, we took into account 7 features (affective words, dynamic and semantically oriented adjectives, conjuncture verbs, see verbs, marvel verbs and level of abstraction of nouns), while for the low-level ones we used just the unigram representation of the datasets. In particular, we used the statistical analysis software XLSTAT[1]

[1] http://www.xlstat.com/ [16th *November*, 2010].

to perform the MDS process.

5.2.2.1 In-Domain Visual Representation

In this subsection, our aim is to show how the subjective and objective texts position themselves on a map in the space of features for each dataset. In practice, the projection space we build with the MDS from such a distance is sufficient to have an idea about whether data are organized into classes or not. For our purpose, we performed the MDS process on all corpora trying to visualize subjective texts from objective ones. To perform visual analysis we represent each document as a set of the following high-level features: dynamic adjectives, polarity adjectives, conjecture verbs, marvel verbs, see verbs, affective words and level of abstraction (left pictures), and unigrams as low-level features (right pictures). The obtained visualizations (Figures 5.1, 5.2, 5.3, 5.4) show distinctly that a particular data organization can be drawn from the data.

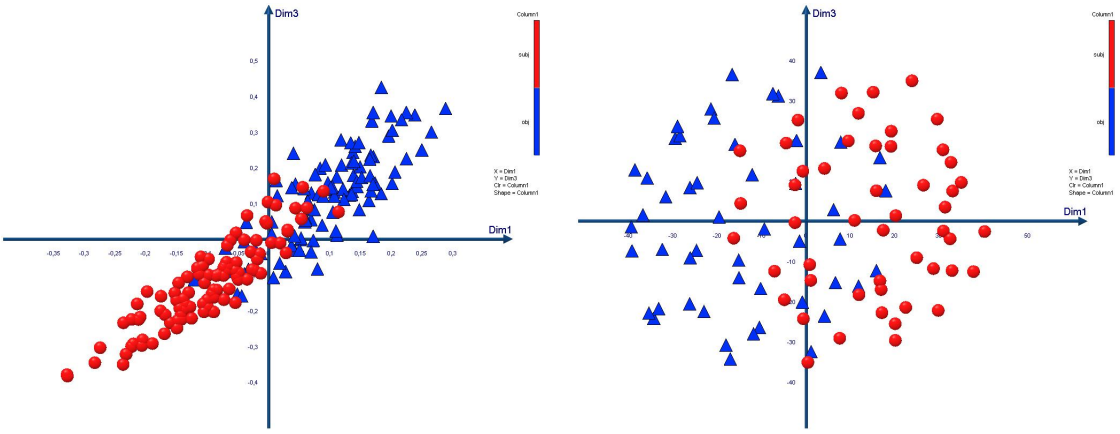

Figure 5.1: MDS: RIMDB visualization of subjective (red dots) and objective (blue triangles) texts In-Domain: $\{HLF\}$ (left) and $\{LLF\}$ (right).

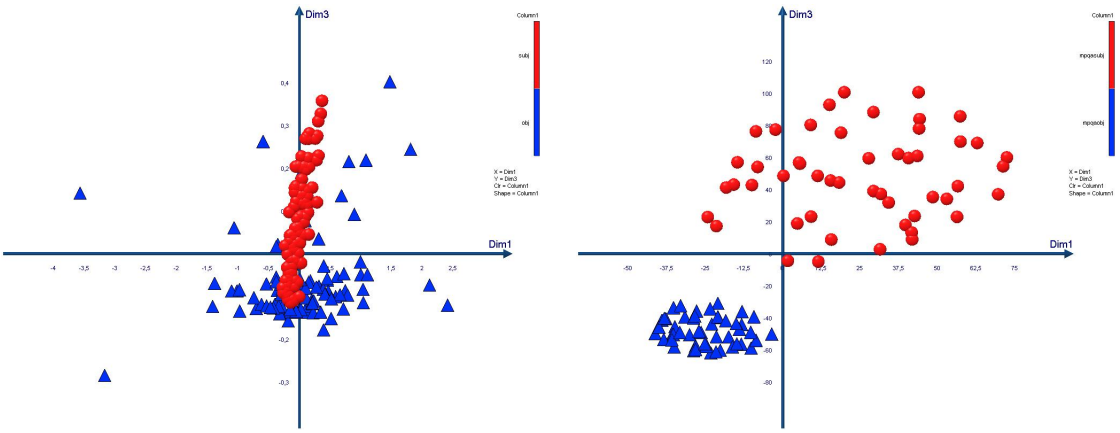

Figure 5.2: MDS: MPQA visualization of subjective (red dots) and objective (blue triangles) texts In-Domain: $\{HLF\}$ (left) and $\{LLF\}$ (right).

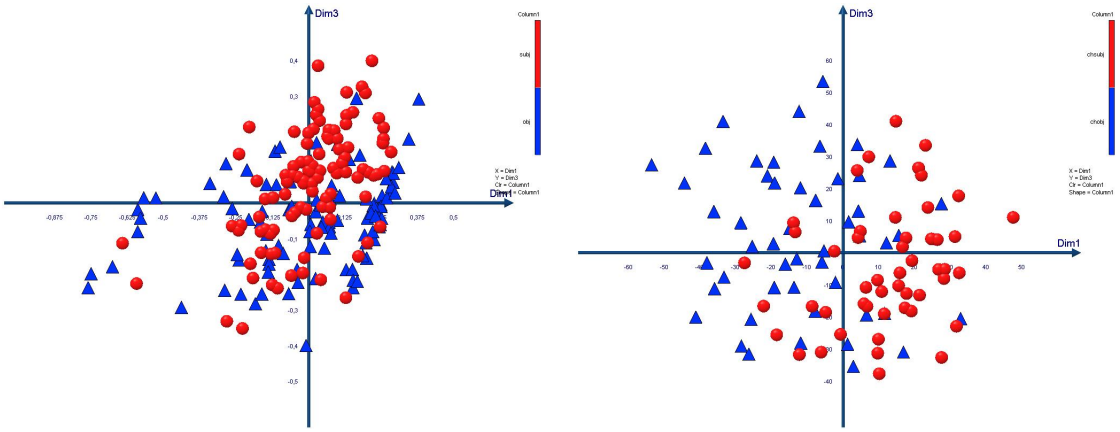

Figure 5.3: MDS: CHES visualization of subjective (red dots) and objective (blue triangles) texts In-Domain: $\{HLF\}$ (left) and $\{LLF\}$ (right).

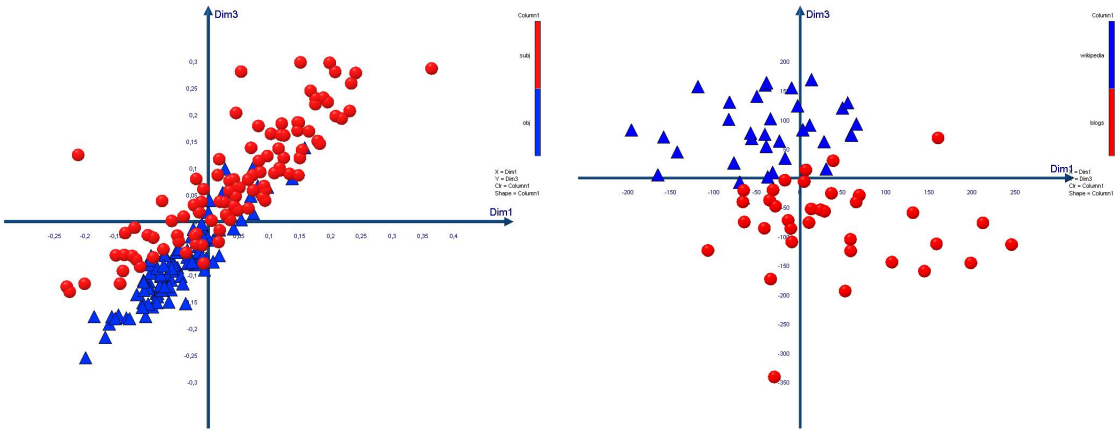

Figure 5.4: MDS: WBLOG visualization of subjective (red dots) and objective (blue triangles) texts In-Domain: $\{HLF\}$ (left) and $\{LLF\}$ (right).

These visualizations clearly show that exclusively objective and subjective texts (i.e. $\{RIMDB\}$ in Figure 5.1 and $\{MPQA\}$ in Figure 5.2) may lead to improved results as the data is well separated in the reduced 2-dimension space. In the case of the $\{CHES\}$ corpus (Figure 5.3) and $\{WBLOG\}$ (Figure 5.4) separating data in the space seems more difficult. Indeed, as these texts are not composed exclusively of subjective or objective sentences, the overlap in the space is inevitable. As Wiebe *et al.* (2004) state, 75% (resp. 56%) of the sentences in subjective (resp. objective) texts are subjective (resp. objective). The results of these visualizations also evidence an important gain with low level features (right pictures) compared to high level features (left pictures). Indeed, as it has already been stated in related works, objective language and subjective language hardly intersect, which means that one word, specific to one domain (i.e. which does not represent any subjective value outside it) can easily distinguish between objective and subjective texts. However, a pattern in the space seems to emerge comforting us in the choice of our high-level features for our classification task.

5.2.2.2 Cross-domain visual representation

In order to test models across domains, we propose to train different models based on one domain only at each time and test the classifiers over all domains together. Before doing the experiments it is important to visualize the data to understand how difficult the task may be. Indeed, when gathering texts from different domains, the best case is when all subjective texts represent one unique cloud in the space, which does not intersect with a unique cloud formed by objective texts. However, it is not usually the case. In Figures 5.5, 5.6 and 5.7, we show different situations, which illustrate that positive results may be obtained using high-level features and other cases where results may not be expected when using low level features to characterize each text. Indeed, in the left pictures, which represent each document as a set of high-level features, we can see that subjective texts (yellow and blue triangles) and objective texts (red and green dots) form two separated clouds, although many texts are mixed in the middle of the space. Unlikely, the right pictures, which represent each document as a set of low-level features (i.e. unigrams), evidence different clouds with almost no intersection between subjective (resp. objective) texts between different domains. This clearly indicates that subjectivity is expressed in different ways from domain to domain, thus complicating the learning process. Because of the difference in the vocabulary, by using low-level features, we can easily distinguish the documents of one domain from the documents of another domain (In Figure 5.7 - right red dots and yellow triangles represent texts from RIMBD, while green dots and blue triangles represent texts from the Chesley's dataset). Conversely, the task to separate the subjective (triangles) from the objective texts (dots) does not seem so easy. As a consequence, unreliable results may be expected using low level features across domains.

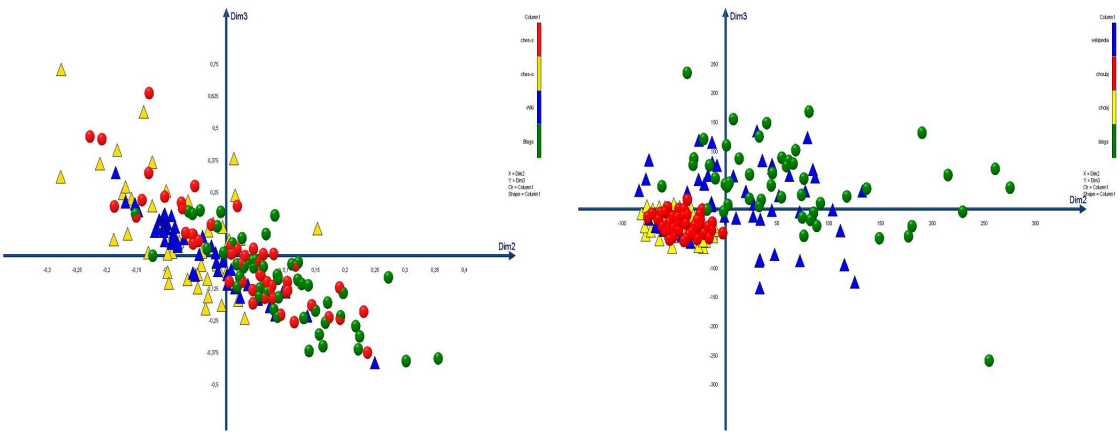

Figure 5.5: MDS: visualization over mixed dataset (WBLOG-CHES) of subjective (red and green dots) and objective (blue and yellow triangles) texts: $\{HLF\}$ (left) and $\{LLF\}$ (right).

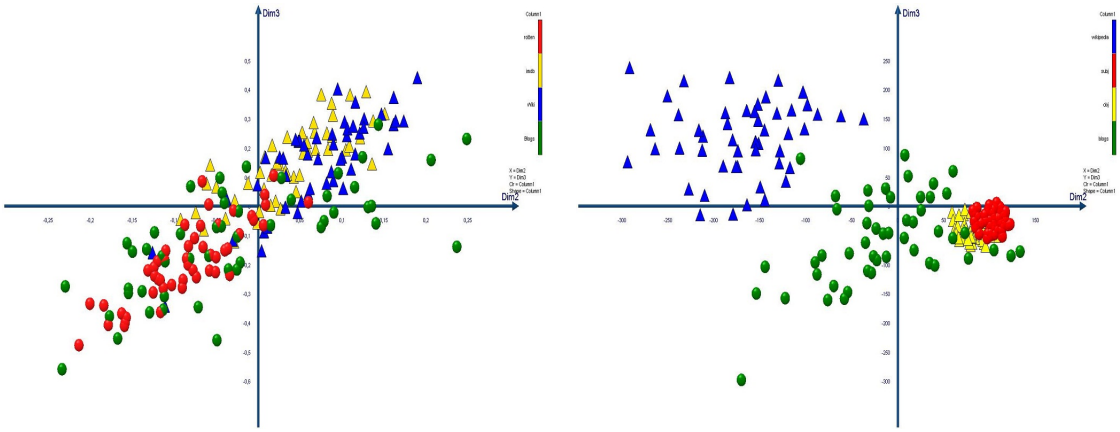

Figure 5.6: MDS: visualization over mixed dataset (WBLOG-RIMDB) of subjective (red and green dots) and objective (blue and yellow triangles) texts: $\{HLF\}$ (left) and $\{LLF\}$ (right).

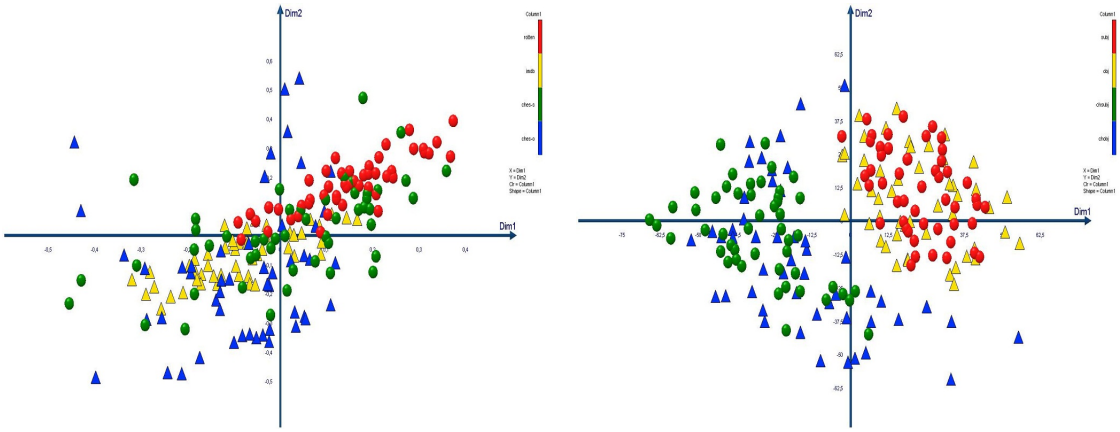

Figure 5.7: MDS: visualization over mixed dataset (RIMDB-CHES) of subjective (red and green dots) and objective (blue and yellow triangles) texts: $\{HLF\}$ (left) and $\{LLF\}$ (right).

5.3 Summary

In this chapter, we have presented state-of-the-art features used in learning models and introduced a new relevant characteristic based on level of abstraction of nouns. We also proposed feature selection and visualization techniques in order to evaluate how well the datasets can be represented in the given space of features. By means of the Wilcoxon rank-sum test and MDS we proved that all of the following text features (Intensity of Affective Words, Dynamic Adjectives and Semantically Oriented Adjectives, Classes of Verbs and Level of Abstraction of Nouns) have good properties to distinguish subjectivity from objectivity.

Once a desirable features set has been obtained, a variety of machine learning algorithms can be used to train sentiment classifiers. So, in the next chapter, we describe a new scheme based on the multi-view and semi-supervised learning over two views and joined two different classifiers LDA and SVM to maximize the optimality of the approach.

Chapter 6

Sentiment Classification

"*Every story has three sides. Yours, mine and the facts.*"

Foster M. Russell

In this chapter, we introduce single-view and multi-view supervised strategies to tackle sentiment classification. First, we review the standard formulations of Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM), which we have used for sentiment classification. Then, we describe new algorithms based on the Semi-supervised and Multi-view learning using both LDA and SVM to optimize the approach.

6.1 Single view Supervised Sentiment Classification

Some of the most popular approaches to sentiment classification are based on supervised and semi-supervised machine learning methods. The task of sentiment classification can be considered as a text categorization (i.e. text classification) task, where texts are classified into one of several predefined categories using information from training texts. In the text categorization task, various machine learning methods have been applied and have proven to be successful (Sebastiani (2002)). The same methods have been applied to the sentiment classification by many researchers (Pang *et al.* (2002), Yu & Hatzivassiloglou (2003), Chesley *et al.* (2006) etc.). In particular, SVM (Joachims (2002)) have consistently shown better performance than other classification algorithms for topical text classification in general (Joachims (1998)), and for sentiment classification in particular (Pang *et al.* (2002), Pang & Lee (2004)). However, other algorithms have been proposed for sentiment classification, which perform well and even better in some cases than SVM, e.g. K-nearest neighbors (Wiebe *et al.* (2004)), maximum entropy (Boiy *et al.* (2007), Naive Bayes (Yu & Hatzivassiloglou (2003), Aue & Gamon (2005)) and sequential minimal optimization (Aue & Gamon (2005)). In this study, we propose to use LDA as an alternative to SVM. Indeed, SVM proved to work better when the number of features is high, but in the context of our work, only seven features are used for classification. Within this context, LDA is particularly well-suited for our classification task as it constructs one or more

linear combinations of the predictor variables such that the different groups differ as much as possible on these discriminant equations.

6.1.1 Support Vector Machines (SVM)

Support Vector Machines (SVM) are the most favored supervised learning method of sentiment classification because of their consistently robust performances in natural language processing (Joachims (2002)). As a discriminative classifier, SVM do not require any prior probabilities or assumptions about the training data as does a generative classifier such as Naive Bayes classifiers. Rather, the key idea behind a binary SVM is to find a decision surface in the feature space that will separate positive and negative training examples. In the case of opinion detection, this means separating subjective from objective examples. Based on the intuition that, when there are no examples near the decision surface, the classification decision is less uncertain and has good generalization capability (Yu (2009)). The best decision surface is one with maximal margin to training examples (Figure 6.1). Where examples are not linearly separable, a kernel function Φ is sometimes used to map the original feature space onto a new space (Figure 6.2).

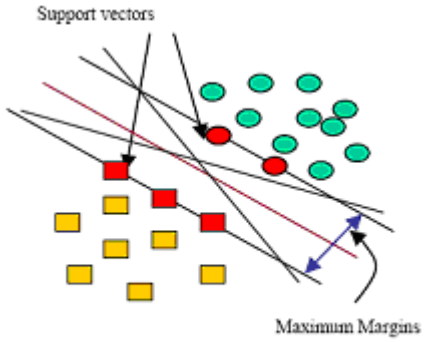

Figure 6.1: SVM maximum margins

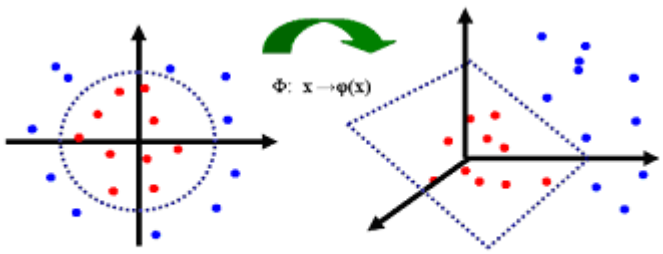

Figure 6.2: SVM kernel function.

Underlying the success of SVM are the mathematical foundations of the statistical learning theory. Rather than simply minimizing the training error, SVM minimizes the structural risk, which expresses an upper bound on generalization error. Assuming a linear decision boundary, the central idea is to find a weighted vector w such that the margin is as large as possible. Assuming that the data is linearly separable, we seek to find the smallest possible w or maximum separation (margin) between the two classes. This can be formally expressed as a quadratic optimization problem in Equations 6.1 and 6.2.

$$\min_{w \notin 0, b} \frac{1}{2} ||w||^2, \tag{6.1}$$

$$s.t.\ y_i(w^T x_i + b) \geq 1\ \forall i = 1, ...n. \tag{6.2}$$

By transforming the above convex optimization problem into its dual problem, the solution can be found in the form $w = \sum_{i=1}^{n} \alpha_i y_i x_i$, where only α_i corresponding to those data points, which achieve equality constraints in equation 6.2 are nonzero. These data samples are called support vectors (Xiong & Cherkassky (2005)). SVM is a local method in the sense that the solution is exclusively determined by support vectors whereas all other data points are irrelevant to the decision hyperplane.

Pang *et al.* (2002) made early efforts to adapt several popular supervised classification algorithms to sentiment classification in the field of movie reviews. They tested Naive Bayes Classifiers, Maximum Entropy and Support Vector Machines to see which would best classify the movie reviews in the Polarity Dataset[1]. The answer was fairly conclusive. SVM outperformed the other two algorithms with most combinations of features, and had the highest classification accuracy of 82.9% using bag-of-words features only. The advantages of utilizing the SVM algorithm for sentiment classification lie in its ability to handle a mix of different types of features, as well as to work with diverse Web content. Chesley *et al.* (2006) fed a mix of part-of-speech tag based features (first-person pronouns, second-person pronouns, adjectives and adverbs), verb classes, positive/negative adjectives and punctuation marks into an SVM classifier and got 76.3%, 86.8% and 80.3% classification accuracy for objective, positive and negative blog post classification, respectively. These results were considered promising given the noisy nature of blog posts. For in-domain polarity detection, SVM classifiers with various opinion-bearing features have been able to attain results comparable to those of the best topical classifiers (Yu (2009)). Thus far, SVM are the most consistently effective algorithms reported to classify opinions. The drawback of SVM, however, is their computing inefficiency compared to Naive Bayes classifiers, which can sometimes yield comparable results.

[1] http://www.cs.cornell.edu/People/pabo/movie-review-data/ [16th $November, 2010$].

6.1.2 Linear Discriminant Analysis (LDA)

In this study, we propose to use Linear Discriminant Analysis (LDA) as an alternative to SVM. Discriminative classifiers seek to find a decision boundary that maximizes certain measure of separation between classes. The earliest of such methods, Fisher's Linear Discriminant Analysis (Fukunaga (1990)), tries to find a linear combination of input variables that discriminates best between the two class distributions (estimated from available data). The classical LDA approach proved to be extremely useful in practice, and it has been successfully applied in many situations where the underlying assumptions (about the class distributions) for the LDA approach do not hold. For instance, LDA is often used to discriminate between the class distributions with different covariance matrices (where an optimal decision boundary is known to be nonlinear, i.e. quadratic). Practical attractiveness of LDA can be explained by its low model complexity, and its ability to capture the essential characteristics of the data distributions (mean and covariance) from finite training data, and then estimating the decision boundary using these global characteristics of the data (Xiong & Cherkassky (2005)). As a result, LDA benefits from feature combinations that produce the highest separation between classes. Therefore, classical LDA has been used and re-discovered in many recent learning techniques. It has been successfully applied in many applications such as face recognition (Belhumeur *et al.* (1997)) and text classification (Torkkola (2001)).

Classical LDA aims at finding optimal transformation by minimizing the within-class distance and maximizing the between class distance simultaneously, thus achieving maximum discrimination. The optimal transformation can be readily found by computing the eigen-decomposition on the scatter matrices. Mathematically, LDA (for the binary classification problem) tries to find a direction vector $w^* \in R^d$ as a solution of the optimization problem expressed in Equation 6.3.

$$w^* = arg \max_{w} \frac{w^T S_b w}{w^T S_w w} \tag{6.3}$$

where the between and within class covariance S_b and S_w are defined as in Equations 6.4 and 6.5.

$$S_b = (m_1 - m_2)(m_1 - m_2)^T \tag{6.4}$$

$$S_w = \sum_{i \in 1,2} \sum_{x \in Z_i} (x - m_i)^2 \tag{6.5}$$

where X denote the space of observations (e.g. $X \subseteq R^d$) and Y the set of possible binary labels ($Y = \{+1, 1\}$). $Z = \{(x_1, y_1), ..., (x_n, y_n)\} \subset X \times Y$ denote the i.i.d. training sample of size n drawn according to some probability measure $P(x, y)$. m_1 and m_2 are the empirical class means, i.e. $m_i = \frac{1}{n} \sum_{x \in Z_i}$, $x, i = 1, 2$. When S_w is not singular, one way to solve the above

optimization problem is to apply the eigen-decomposition to the matrix $S_w^{-1}S_b$. The eigenvector corresponding to the largest eigenvalue forms w. Ignoring the scaling factor, we obtain w, as in Equation 6.6.

$$w^* = S_w^{-1}(m_1 - m_2). \tag{6.6}$$

When S_w is singular, we can add a diagonal matrix to the within-class covariance matrix. Usually an identity matrix with a small scalar multiple is used, as defined in Equation 6.7.

$$w^* = (S_w + \lambda I)^{-1}(m_1 - m_2). \tag{6.7}$$

The success of LDA is partially due to the fact that only up to second order moments (mean and covariance) of the class distribution are used. As such, this approach is more robust than estimating the distribution of the data (Xiong & Cherkassky (2005)).

So, in this study we wish to find out a method for prediction of group membership for a number of texts using the following 7 high-level features: affective words, dynamic adjectives, polarity adjectives, conjecture verbs, marvel verbs, see verbs and level of abstraction. These predictors vary sufficiently over the different groups (the group of subjective texts and the group of objective texts). The criterion variable type of the text (also called grouping variable) is a categorical variable and it is the object of classification. The aim of the Linear Discriminant Analysis is to construct one or more discriminant equations (linear combinations of the predictor variables) such that the different groups differ as much as possible on these discriminant equations. In our task, one discriminant function distinguishes the first group (the group of objective texts) from the second one (the group of subjective texts).

6.2 Semi-Supervised and Multi-View Learning for two views

To date, many Multi-view and Semi-supervised learning algorithms have been developed to address the cross-domain text classification problem by transferring knowledge across domains. Some approaches use the multi-view setting to train learners and then let the learners to label unlabeled examples. The multi-view setting was first formalized by Blum and Mitchell (Blum & Mitchell (1998)), where there are several disjoint subsets of features (each sub-set is called a view), each of which is sufficient to learn the target concept. However, there is a slight difference between Semi-supervised and Multi-view learning. While Semi-supervised learning is usually associated to small labeled data sets and tries to automatically increase the number of labeled examples, Multi-view learning aims at learning a compromise model of the different views. The most important work in multi-view sentiment classification is proposed by Ganchev *et al.* (2008), who present a new algorithm called SAR, which outperformed the results proposed

earlier by Blitzer *et al.* (2007) on the same data set. However, Ganchev *et al.* (2008) only use low-level features, which are randomly divided to form two "artificial" views. Instead, we aim at combining high-level features and low-level features to learn models of subjectivity, which may apply to different domains. For that purpose, we proposed in Lambov *et al.* (2010) a new scheme based on the classical co-training algorithm over two views and joined two different classifiers LDA and SVM to maximize the optimality of the approach. As a consequence of this work, we propose several different approaches based on multi-view learning for two and more than views. These issues will be discussed in the final section of this chapter.

6.2.1 Multi-View Learning with Agreement

The field of Semi-Supervised Learning offers a number of possible strategies to compensate the lack of labeled data. These techniques may not be effective if the model induced from the initial set of labeled data is biased or ineffective because these strategies can exacerbate the weaknesses of the initial model. Therefore, we propose a novel approach based on multi-view learning using one view, which contains high-level features and a second view, which contains low-level features (unigrams or bigrams). Indeed, based on related works, we know that high-level features provide strong opinion evidence across domains (Finn & Kushmerick (2006), Lambov *et al.* (2009a)). On the other side, word-based models show remarkable results for in-domain classification task (Pang *et al.* (2002)). As a consequence, we expect that the low-level classifier will gain from the decisions of the high-level classifiers and will self-adapt to different domains based on the high results of high-level features for crossing domains. Based on this hypothesis we propose to use five different methodologies to combine high-level and low-level features, introduced in the following subsections.

6.2.1.1 Stochastic Agreement Regularization (SAR)

One of the most important work in multi-view sentiment classification is proposed by Ganchev *et al.* (2008), who present a new algorithm called SAR. It models a probabilistic agreement framework based on minimizing the Bhattacharyya distance (Kailath (1967)) between models trained using two different views. They regularize the models from each view by constraining the amount by which they permit them to disagree on unlabeled instances from a theoretical model. Their co-regularized objective which has to be minimized is defined in Equation 6.8, where L_i for $i = 1,2$ are the standard regularized log likelihood losses of the models p_1 and p_2, $E_u[B(p_1(\theta_1), p_2(\theta_1))]$ is the expected Bhattacharyya distance between the predictions of the two models on the unlabeled data, and c is a constant defining the relative weight of the unlabeled

data.

$$Min[L_1(\theta_1) + L_2(\theta_2) + cE_u[B(p_1(\theta_1), p_2(\theta_1))]]. \tag{6.8}$$

They used the Bhattacharyya distance as symmetric measure of distance between two distributions and it is given by Equation 6.9.

$$B(p_1, p_2) = -\log \sum_y \sqrt{p_1(y)p_2(y)}. \tag{6.9}$$

Proposition 1. *The Bhattacharyya distance:* $-\log \sum_y \sqrt{p_1(y)p_2(y)}$ *is equal to* $\frac{1}{2}$ *of the value of the convex optimization problem:*
$\min_{q \in Q} KL(q(y_1, y_2)||p_1(y_1)p_2(y_2))$
where $Q = \{q : E_q[\delta(y_1 = y) - \delta(y_2 = y)] = 0 \ \forall y\}$*, where* $\delta(cond)$ *is 1 if cond is true and 0 otherwise. Furthermore, the minimizer decomposes as* $q(y_1, y_2) = q_1(y_1)q_2(y_2)$ *and is given by* $q_i(y) \propto \sqrt{p_1(y)p_2(y)}$.

Replacing the Bhattacharyya regularization term in Equation 6.8 with the program of Proposition 1 yields the objective:
$min_\theta \ L_1(\theta_1) + L_2(\theta_2) + cE_u[\min_{q \in Q} KL(q(y_1, y_2)||p_1(y_1)p_2(y_2))]$.

They defined $agree(p_1, p_2)$ to be the minimizer of Proposition 1, and presented the optimization algorithm (Algorithm 4).

Algorithm 4 Minimizes co-regularized loss:
$L_1(\theta_1) + L_2(\theta_2) + cE_u[\min_{q \in Q} KL(q(y_1, y_2)||p_1(y_1)p_2(y_2))]$.

1: $\theta_1 \leftarrow min_\theta L_1(\theta_1)$
2: $\theta_2 \leftarrow min_\theta L_2(\theta_2)$
3: **for** n iterations **do**
4: $\quad q(y_1, y_2|x) \leftarrow \mathbf{agree}(p_1(y_1|x), p_2(y_2|x)) \ \forall x \in U$
5: $\quad \theta_1 \leftarrow min_\theta L_1(\theta) - c \ E_{x \sim U, y_1 \sim q}[\log p_1(y_1|x; \theta)]$
6: $\quad \theta_2 \leftarrow min_\theta L_2(\theta) - c \ E_{x \sim U, y_2 \sim q}[\log p_2(y_2|x; \theta)]$
7: **end for**

The idea is the following. They train a model for each view, and use constraints that the models should agree over the unlabeled distribution. They focus mainly on two discriminative log-linear models and start by considering the settings of complete agreement.

In the context of sentiment classification and multi-view learning, Ganchev *et al.* (2008) is certainly one of the best up-to-date reference, reaching accuracy levels of 82.8% for polarity

detection upon reviews from the kitchen and the dvd domains using random views of unigrams. In this work, we will test SAR on our dataset both on random views of unigrams and random views of bigrams and take its results as baselines.

6.2.1.2 Merged Agreement Algorithm (MAA)

The idea of SAR algorithm (Ganchev *et al.* (2008)) is to train a model for each view, and use constraints that the models should agree over the unlabeled distribution. As this method is based on agreement between the models, we propose to use three methods based on multi-view learning and also using agreement between predictions of both classifiers.

The Merged Agreement Algorithm (MAA) is an adaptation of the algorithm proposed in Wan (2009). It is based on the co-training algorithm with agreement, but instead of just taking into account unlabeled examples with similar predictions from both classifiers to update the set of labeled examples such as in Wan (2009), we impose that only the examples with highest confidence upon agreement are added to the labeled list. Basically, the MAA takes two main inputs: a set of labeled examples from one domain (L), the source domain, and a set of unlabeled examples from another domain (U), the target domain. After training on the source domain, both classifiers classify unlabeled documents from the target domain. If both classifiers agree on their predictions, the unlabeled document is added to an agree list for each classifier with the categorization label and the classification confidence. Finally, the P positive (subjective) and N negative (objective) documents with higher confidence values are selected from each agree list and transferred from the set of unlabeled documents to the labeled set. This method is described in Algorithm 5.

It is important to point at the fact that the MAA algorithm as the one proposed by Wan (2009) may produce unbalanced data sets. Indeed, from both agree lists of $H1$ and $H2$, we may update the labeled list with more positive examples than negative ones and vice and versa as classifiers may agree more on one class than another. Without a balancing parameter, it is not necessary to have a minimum number of positive or negative documents, which agree on labels for both classifiers. If $H1$ and $H2$ agree only on positive predictions then only positive examples will be added to L. As a consequence, in the next section, we propose to modify the MAA to balance the parameter values of P and N at each iteration.

6.2.1.3 Balanced Merged Agreement Algorithm (BMAA)

The second method is similar to the previous one, but here equal number of positive and negative documents must be selected from each of the agree lists and transferred from the set of unlabeled documents into the labeled set. In this algorithm (Algorithm 6), the class distribu-

Algorithm 5 Merged Agreement Algorithm (MAA).

1: Input: L a set of labeled examples from one domain, U a set of unlabeled examples from another domain
Output: Trained classifier $H2$
2: $H1.AgreeList \leftarrow \{\}$
3: $H2.AgreeList \leftarrow \{\}$
4: **for** k iterations **do**
5: Train a classifier $H1$ on view $V1$ of L
6: Train a classifier $H2$ on view $V2$ of L
7: Allow $H1$ and $H2$ to label U
8: **for all** $d \in U$ **do**
9: **if** $H1.Class[d] = H2.Class[d]$ **then**
10: $H1.AgreeList \leftarrow H1.AgreeList \cup \{< d; H1.Class[d] >\}$
11: $H2.AgreeList \leftarrow H2.AgreeList \cup \{< d; H2.Class[d] >\}$
12: **end if**
13: **end for**
14: $L \leftarrow L \cup$ {the most confidently predicted P positive and N negative examples from $H1$ on $U \in H1.AgreeList$}
15: $L \leftarrow L \cup$ {the most confidently predicted P positive and N negative examples from $H2$ on $U \in H2.AgreeList$}
16: **end for**

tion in the labeled data is maintained by balancing the parameter values of P and N at each iteration. If the number of predicted subjective or objective documents is equal to 0, it is used as a stopping criterion. Otherwise, the minimum number of positive or negative new labeled examples is chosen to update the source labeled example list L. This cycle is repeated for k iterations or until there are no positive or negative candidate documents in the agree lists. We called this method the Balanced Merged Agreement Algorithm (BMAA). This is our second algorithm proposal.

6.2.1.4 Balanced Merged Agreement Algorithm Using Documents Rank (BMAADR)

In the previous algorithms, although balanced data sets are kept, confidence is not taken into account for the agreement. So, we propose a new algorithm where most confidently classified pairs of examples are selected for new labeled data. With the MAA and the BMAA algorithms, the most confidently predicted P and N examples from $H1$ and $H2$ are selected to update L. However, for example, we may update L with a positive example from $H1$, which agrees on the classification of $H2$ but where the difference between each confidences is high. As a

Algorithm 6 The Balanced Merged Agreement Algorithm (BMAA).

1: Input: L a set of labeled examples from one domain, U a set of unlabeled examples from another domain, $P = N = X$
Output: Trained classifier $H2$
2: $H1.AgreeList \leftarrow \{\}$
3: $H2.AgreeList \leftarrow \{\}$
4: **for** k iterations **do**
5: Train a classifier $H1$ on view $V1$ of L
6: Train a classifier $H2$ on view $V2$ of L
7: Allow $H1$ and $H2$ to label U
8: **for all** $d \in U$ **do**
9: **if** $H1.Class[d] = H2.Class[d]$ **then**
10: $H1.AgreeList \leftarrow H1.AgreeList \cup \{< d; H1.Class[d] >\}$
11: $H2.AgreeList \leftarrow H2.AgreeList \cup \{< d; H2.Class[d] >\}$
12: **end if**
13: **end for**
14: **if** ((number of all positive examples $\in H1.AgreeList$)=0) or ((number of all negative examples $\in H1.AgreeList$)=0) **then**
15: $stop$
16: **else if** ((number of all positive examples $\in H1.AgreeList$)<X) or ((number of all negative examples $\in H1.AgreeList$)<X) **then**
17: X=min((number of all positive examples $\in H1.AgreeList$), (number of all negative examples $\in H1.AgreeList$))
18: **end if**
19: $L \leftarrow L \cup$ {the most confidently predicted P positive and N negative examples from $H1$ on $U \in H1.AgreeList$}
20: $L \leftarrow L \cup$ {the most confidently predicted P positive and N negative examples from $H2$ on $U \in H2.AgreeList$}
21: **end for**

consequence, we may update L with examples where only one of the classifiers is very confident about the classification although they agree on the classification. The idea of our new proposal is to measure an "average" confidence value for all examples for which there is agreement between classifiers so that the highest "on average" new labeled examples are added to L. For that purpose, after each classification on unlabeled data, both lists of documents are sorted by decreasing classification confidence. The best examples are at the top of the agree lists. So, each document is located at one position in the agree list of $H1$ and on another position in the agree list of $H2$. Based on these two positions in the different sorted agree lists, we

reckon a new position, which is the average of the positions of the document d in both lists. Finally, we sort the documents according to their new average position, which is their new confidence value. Then, the best P positive and N negative examples are added to the labeled data set L depending on their new confidence value. For example, if a document d is classified with the best confidence value from the first classifier $H1$, but its assessment from the second classifier $H2$ takes the 20th place in the sorted agree list of $H2$, the new confidence value of d will be 10.5. So, although this document is classified with the best confidence value by the first classifier, it is likely that it will not be added to the labeled examples, while there are documents with a higher average value, even if they are not so good for any of the two classifiers. This method is described in Algorithm 7 and is called the Balanced Merged Agreement Algorithm Using Documents Rank (BMAADR).

Algorithm 7 The BMAADR Algorithm.

1: Input: L a set of labeled examples from one domain, U a set of unlabeled examples from another domain, $P = N = X$
   Output: Trained classifier $H2$
2: $H1.AgreeList \leftarrow \{\}$
3: $H2.AgreeList \leftarrow \{\}$
4: **for** k iterations **do**
5: Train a classifier $H1$ on view $V1$ of L
6: Train a classifier $H2$ on view $V2$ of L
7: Allow $H1$ and $H2$ to label U
8: **for all** $d \in U$ **do**
9: **if** $H1.Class[d] = H2.Class[d]$ **then**
10: $H1.AgreeList \leftarrow AgreeList \cup \{< d; H1.Class[d] >\}$
11: $H2.AgreeList \leftarrow AgreeList \cup \{< d; H2.Class[d] >\}$
12: **end if**
13: **end for**
14: $Sort(H1.AgreeList, byDecrConf.)$
15: $Sort(H2.AgreeList, byDecrConf.)$
16: **for all** $d \in H1.AgreeList$ **do**
17: $Rank_d = \frac{\sum_{i \in \{H1,H2\}} Rank_d^{i.AgreeList}}{2}$
18: $topAgreeList \leftarrow (d, Rank_d)$
19: **end for**
20: $L \leftarrow L \cup$ {Balanced P positive and N negative examples with the lower rank from $topAgreeList$}
21: **end for**

6.2.2 Semi-Supervised Learning for two views

In this section, we describe two methods for which it does not matter whether both classifiers agree in predicted labels. The objective of these methods is to use the best classifier to guide the selection of the data that is added to the training set in the self-training process. We expect that the best classifier will play a role of a supervisor that prevents wrongly classified documents from being added to the training data, which would otherwise impact the future iterations of the self-training process. In these methods, a new validation labeled dataset is used as a classifier confidence threshold.

6.2.2.1 Co-Training

Semi-supervised learning has absorbed much attention from researchers because a large number of unlabeled examples may boost performance of a learning algorithm when only a smaller number of labeled examples is available. Blum & Mitchell (1998) first considered a problem setting that the description (feature set) of each example could be partitioned into two distinct views (two disjoint feature sets). For example, a Web page can be described by the words occurring on that Web page, and also by the words occurring in hyperlinks that point to that page. We can either use the set of words occurring on the Web pages to classify other pages, or use the set of words occurring in hyperlinks to classify the pages. The algorithms, which use the above problem setting are referred to co-training algorithms.

Blum & Mitchell (1998) showed that the co-training algorithm works well for semi-supervised learning, if the feature set division of dataset satisfies two assumptions, that is, (1) each set of features is sufficient for classification, and (2) the two feature sets of each instance are conditionally independent given the class. The first assumption is about the compatibility of the instance distribution with the target function. That is, the target functions over each feature set predict the same label for most examples. For example, the prediction of Web pages should be identifiable using either the content text or the hyperlink vocabulary. The second assumption is the most difficult one which is somewhat unrealistic in practice. In the example of Web page classification, this assumes that the words on a Web page are not related to the words on its incoming hyperlinks, except through the class of the Web page.

Co-training begins at using a weak initial hypothesis over one feature set and labeled examples. Under the conditional independence assumption, these examples may be randomly distributed to the other classifier. Though the classification noise from the weak hypothesis would be brought to the other classifier, the algorithm can learn from these labeled examples by an iterative procedure between the two classifiers. The general co-training algorithm is presented in algorithm 8.

Algorithm 8 The co-training algorithm for two views.

Input: L a set of labeled examples from one domain, U a set of unlabeled examples from another domain
Output: Trained classifier $H2$
for k iterations **do**
  Train a classifier $H1$ on view $V1$ of L
  Train a classifier $H2$ on view $V2$ of L
  Allow $H1$ and $H2$ to label U
  Add the most confidently predicted P positive and N negative examples to L
end for

In the remainder of this section, we propose new ways to look at semi-supervised and multi-view learning based on the co-training framework.

6.2.2.2 Guided Semi-Supervised Learning (GSL)

The Guided Semi-supervised Learning (GSL) algorithm takes three main inputs: a set of labeled examples from one domain (L), the source domain, the set of unlabeled examples from another domain (U), the target domain, and a validation (VL) data set (i.e. a small set of labeled examples from the target domain). The proposed technique uses the validation data set to guide the selection of training candidates, which are labeled by both classifiers in the self-training process. At the end of each learning iteration, both classifiers are applied to validation data set VL and receive an accuracy score. As a consequence, P positive and N negative examples with higher confidence values are added to L only from the classifier with the best validation classification accuracy at each iteration. This method is described in Algorithm 9 for two views.

The accuracy is the proportion of the total number of correct predictions. It is determined by equation 6.10:

$$Accuracy = \frac{TrueSubj + TrueObj}{TrueSubj + TrueObj + FalseSubj + FalseObj}. \tag{6.10}$$

6.2.2.3 Class-Guided Semi-Supervised Learning (C-GSL)

The Class-guided Semi-supervised Learning (C-GSL) algorithm takes the same inputs as the GSL algorithm: a set of labeled examples from one domain (L), the source domain, the set of unlabeled examples from another domain (U), the target domain, and a validation (VL) data set. The main difference with the GSL algorithm is that instead of relying on the global accuracy over the VL data set and choose the corresponding classifier to guide the learning process, we choose the classifier with higher precision for subjectivity to label the P positive examples from U and

Algorithm 9 The Guided Semi-Supervised Learning (GSL) algorithm for two views.

1: Input: L a set of labeled examples from one domain, U a set of unlabeled examples from another domain, VL a validation test set of labeled examples from target domain
Output: Trained classifier $H2$
2: **for** k iterations **do**
3: Train a classifier $H1$ on view $V1$ of L
4: Train a classifier $H2$ on view $V2$ of L
5: Allow $H1$ and $H2$ to label VL
6: Allow $H1$ and $H2$ to label U
7: **if** $H1.Acc[VL] > H2.Acc[VL]$ **then**
8: $L \leftarrow L \cup$ {the most confidently predicted P positive and N negative examples from $H1$ on U}
9: **else**
10: $L \leftarrow L \cup$ {the most confidently predicted P positive and N negative examples from $H2$ on U}
11: **end if**
12: **end for**

the classifier with higher precision for objectivity to label the N negative examples from U. So, at each learning iteration, we compare the subjective precision and the objective precision obtained by each classifier over the validation data set VL and choose the best one for each class. Here, unlike the previous method, we expect to reduce the number of wrong examples added to the labeled data set L mainly due to the fact that the best classification accuracy of one classifier is exclusively due to high accuracy over only subjective or only objective predictions. Indeed, with the GSL algorithm, we may label new examples from U as subjective or objective based on the classifier with best accuracy overall although its precision over one of the classes may be low. In this case, this method would take subjective (resp. objective) examples from the best subjective (resp. objective) classifier. This method is described in Algorithm 10 for two-views.

Subjective precision is the proportion of the correctly predicted subjective examples, as calculated using the Equation 6.11:

$$SubjPrecision = \frac{TrueSubj}{TrueSubj + FalseSubj}. \tag{6.11}$$

Objective precision is the proportion of the correctly predicted objective examples, as calcu-

Algorithm 10 The Class-Guided Semi-Supervised Learning algorithm for two views.

1: Input: L a set of labeled examples from one domain, U a set of unlabeled examples from another domain, VL a validation test set of labeled examples from target domain
Output: Trained classifier $H2$
2: **for** k iterations **do**
3: Train a classifier $H1$ on view $V1$ of L
4: Train a classifier $H2$ on view $V2$ of L
5: Allow $H1$ and $H2$ to label VL
6: Allow $H1$ and $H2$ to label U
7: **if** $H1.SubjPrec[VL] > H2.SubjPrec[VL]$ **then**
8: $L \leftarrow L \cup$ {the most confidently predicted P positive examples from $H1$ on U}
9: **else**
10: $L \leftarrow L \cup$ {the most confidently predicted P positive examples from $H2$ on U}
11: **end if**
12: **if** $H1.ObjPrec[VL] > H2.ObjPrec[VL]$ **then**
13: $L \leftarrow L \cup$ {the most confidently predicted N negative examples from $H1$ on U}
14: **else**
15: $L \leftarrow L \cup$ {the most confidently predicted N negative examples from $H2$ on U}
16: **end if**
17: **end for**

lated using the Equation 6.12:

$$ObjPrecision = \frac{TrueObj}{TrueObj + FalseObj}. \tag{6.12}$$

6.3 Semi-Supervised Learning for three views

In the proposed methods above, we used two views to guide the selection of training candidates and to minimize the risk of adding misclassified examples. Within this context, we propose to use Algorithm 9 and 10 with more than two views in order to obtain maximum performance.

6.3.1 Guided Semi-Supervised Learning (GSL) for three views

The first method is similar to Algorithm 9 but with three views. It takes three main inputs, the labeled (L), unlabeled (U) and validation (VL) data sets. The proposed technique uses the validation data set to guide the selection of training candidates, which are labeled by classifiers in the self-training process. At the end of each learning iteration, each classifier must classify

the validation test set, which contains labeled examples from target domain. So, P positive and N negative examples with higher confidence values are added to L only from the classifier with the best validation classification accuracy at each iteration. The idea of using three different views is that different features may have different weights in different domains. For example, Lambov *et al*. (2009b) showed that affective words are strong predictors of subjectivity/objectivity in the News domain, adjectives in Web Blogs and verbs in Movie reviews. Thus, the algorithms are more flexible in trying to choose the best examples in the self-training process. This method is fully described in Algorithm 11.

Algorithm 11 Guided Semi-Supervised Learning (GSL) algorithm for three views.

1: Input: L a set of labeled examples from one domain, U a set of unlabeled examples from another domain, VL a validation test set of labeled examples from target domain
Output: Trained classifier $H3$
2: **for** k iterations **do**
3: Train a classifier $H1$ on view $V1$ of L
4: Train a classifier $H2$ on view $V2$ of L
5: Train a classifier $H3$ on view $V3$ of L
6: Allow $H1$, $H2$ and $H3$ to label VL
7: Allow $H1$, $H2$ and $H3$ to label U
8: **if** $(H1.Acc[VL] > H2.Acc[VL])$ and $(H1.Acc[VL] > H3.Acc[VL])$ **then**
9: $L \leftarrow L \cup$ {the most confidently predicted P positive and N negative examples from $H1$ on U}
10: **else if** $(H2.Acc[VL] > H1.Acc[VL])$ and$(H2.Acc[VL] > H3.Acc[VL])$ **then**
11: $L \leftarrow L \cup$ {the most confidently predicted P positive and N negative examples from $H2$ on U}
12: **else**
13: $L \leftarrow L \cup$ {the most confidently predicted P positive and N negative examples from $H3$ on U}
14: **end if**
15: **end for**

6.3.2 Class-Guided Semi-Supervised Learning (C-GSL) for three views

This method is similar to Algorithm 10 but with three views. It takes three main inputs: the labeled (L), unlabeled (U) and validation (VL) data sets. It is similar to the previous one, but at the end of each learning iteration it is not the accuracy of both classifiers we compare, but the subjective precision and the objective precision obtained from the classification of the validation dataset. In this method, we add P positive examples with higher confidence values from the classifier with the best subjective precision and N negative examples, with higher confidence

values from the classifier with the best objective precision. The idea of this method is that different features may have different weights for subjective/objective classification in different domains. For example, Lambov *et al*. (2009b) showed that high level features performed better in subjective classification, while low-level features performed better in objective classification in the News domain. So, the goal of this method is to choose the best subjective and objective examples, comparing the precision of each class. This method is fully described in Algorithm 12.

Algorithm 12 Class-Guided Semi-Supervised Learning (C-GSL) algorithm for three views.

1: Input: L a set of labeled examples from one domain, U a set of unlabeled examples from another domain, VL a validation test set of labeled examples from target domain
Output: Trained classifier $H3$
2: **for** k iterations **do**
3: Train a classifier $H1$ on view $V1$ of L
4: Train a classifier $H2$ on view $V2$ of L
5: Train a classifier $H3$ on view $V3$ of L
6: Allow $H1$, $H2$ and $H3$ to label VL
7: Allow $H1$, $H2$ and $H3$ to label U
8: **if** $(H1.SubjPrec[VL] > H2.SubjPrec[VL])$ and $(H1.SubjPrec[VL] > H3.SubjPrec[VL])$ **then**
9: $L \leftarrow L \cup$ {the most confidently predicted P positive examples from $H1$ on U}
10: **else if** $(H2.SubjPrec[VL] > H1.SubjPrec[VL])$ and $(H2.SubjPrec[VL] > H3.SubjPrec[VL])$ **then**
11: $L \leftarrow L \cup$ {the most confidently predicted P positive examples from $H2$ on U}
12: **else**
13: $L \leftarrow L \cup$ {the most confidently predicted P positive examples from $H3$ on U}
14: **end if**
15: **if** $(H1.ObjPrec[VL] > H2.ObjPrec[VL])$ and $(H1.ObjPrec[VL] > H3.ObjPrec[VL])$ **then**
16: $L \leftarrow L \cup$ {the most confidently predicted N negative examples from $H1$ on U}
17: **else if** $(H2.ObjPrec[VL] > H1.ObjPrec[VL])$ and $(H2.ObjPrec[VL] > H3.ObjPrec[VL])$ **then**
18: $L \leftarrow L \cup$ {the most confidently predicted N negative examples from $H2$ on U}
19: **else**
20: $L \leftarrow L \cup$ {the most confidently predicted N negative examples from $H3$ on U}
21: **end if**
22: **end for**

6.4 Summary

In this chapter, we gave a formal definition of single-view, multi-view supervised and semi-supervised strategies for sentiment classification. We described different methodologies to combine high-level and low-level features, based on the hypothesis that the low-level classifiers will gain from the decisions of the high-level classifiers and will self-adapt to different domains based on the high results of high-level features to cross domains.

In the following chapter detailed experimental results obtained by each of the algorithms presented above will be presented. We will use the new automatically labeled data set based on Wikipedia and Weblogs texts and compare results with manually tagged corpora.

Chapter 7

Results and Discussion

"Results! Why, man, I have gotten a lot of results. I know several thousand things that won't work."

Thomas A. Edison

In this chapter we outline the detailed evaluation results, obtained by each of the algorithms presented in Chapter 6, both with SVM and LDA within and across domains. First, in order to evaluate the difference between high-level features (HLF) with low-level features (LLF), we performed a comparative study on the four data sets presented in chapter 5 for the single view classification task. Then, we propose to compare the multi-view and semi-supervised algorithms based on two views and combining high-level and low-level features. Finally, we propose to study GSL and C-GSL algorithms with 3 views. For that purpose, we defined four classes of features: affective words, adjectives (semantically oriented and dynamic), verbs (conjecture, marvel and see) and level of abstraction of nouns.

7.1 Evaluation

All experiments, both for SVM and LDA, were performed on a leave-one-out 5 cross validation basis using the SVMlight package[1] for SVM and the free software for statistical computing R[2] for LDA. As part-of-speech tagger, we used the MontyTagger module of MontyLingua[3] (Liu (2004)). For our classification experiments, we used sample of 200 documents for each of the datasets with equal sizes of both subjective and objective texts (i.e. 100 objective texts and 100 subjective). In particular, in order to test models across domains, we trained different models based on one domain only and tested the classifiers over all other domains. In order to train the learning models across domains, we performed as follows. First, we define a source domain and a target domain. After training, we test the learnt low-level classifier over the unseen examples of the target domain. This operation is repeated four times for each

[1] http://svmlight.joachims.org/ [16th *November*, 2010].
[2] http://www.r-project.org/ [16th *November*, 2010].
[3] http://web.media.mit.edu/∼hugo/montylingua/ [16th *November*, 2010].

source domain. For instance, we would train the model on the ($\{MPQA\}$,$\{RIMDB\}$) pair, where $\{MPQA\}$ is the source domain and $\{RIMDB\}$ the target domain, and test the model on the remaining examples of $\{RIMDB\}$. In fact, this process would be repeated for the following pairs ($\{MPQA\}$,$\{MPQA\}$), ($\{MPQA\}$,$\{CHES\}$) and ($\{MPQA\}$,$\{WBLOG\}$). As such, the presented results are the average accuracies for all four experiments. So, each percentage can be expressed as the average results over all data sets. In order to compute the standard evaluation measures (Precision, Recall and Accuracy), we noted positive to denote subjective sentences and negative to denote objective texts. In the tables, overall accuracy is given first, followed by precision and recall with respect to subjective documents. The metrics are defined in Equations 7.1, 7.2 and 7.3.

In the context of subjectivity classification tasks, the terms TrueSubj, TrueObj, FalseSubj and FalseObj are used to compare the given classification of a document(the class label assigned to the document by a classifier) with the desired correct classification (the class the document actually belongs to). This is illustrated by the table 7.1.

Table 7.1: Confusion Matrix

		Predicted	
		Subjective	**Objective**
Actual	**Subjective**	TrueSubj	FalseSubj
	Objective	FalseObj	TrueObj

The accuracy is the proportion of the total number of correct predictions. It is determined by equation 7.1:

$$Accuracy = \frac{TrueSubj + TrueObj}{TrueSubj + TrueObj + FalseSubj + FalseObj}. \tag{7.1}$$

The precision is the proportion of the correctly predicted positive cases, as calculated using the Equation 7.2:

$$Precision_{subj} = \frac{TrueSubj}{TrueSubj + FalseSubj}. \tag{7.2}$$

The recall or true positive rate (TP) is the proportion of positive cases that were correctly identified, as calculated using the Equation 7.3:

$$Recall_{subj} = \frac{TrueSubj}{TrueSubj + FalseObj}. \tag{7.3}$$

7.2 Single-view Classification

In order to evaluate the difference between high-level features and low-level ones, we first performed a comparative study within domains on our four data sets i.e. each model is tested with documents from the same domain of the training texts. For the high-level features, we took into account 7 features (affective words, dynamic and semantically oriented adjectives, conjuncture verbs, see verbs, marvel verbs and level of abstraction of nouns). For the unigram and bigram feature representations, we used all the lemmas inside the corpora withdrawing stop words and weighting lemmas with the classical $tf.idf$ measure (Salton *et al.* (1975)). All experiments were performed on a leave-one-out 5 cross validation basis combined with both SVM and LDA classifiers for high-level features and only SVM for low-level features due to the high number of features which does not suit to LDA classifiers. The results of these experiments are shown in Table 7.2, which respectively presents the accuracy levels for the SVM and the LDA classifiers within domains.

Table 7.2: Accuracy results within domain.

	$\{MPQA\}$	$\{RIMDB\}$	$\{CHES\}$	$\{WBLOG\}$
Unigrams (SVM)	85.4%	97.0%	84.8%	96.2%
Bigrams (SVM)	80.8%	98.0%	87.6%	93.8%
7 features (SVM)	71.2%	86.8%	64.4%	76.2%
7 features (LDA)	93.5%	96.5%	71.0%	94.0%

The results evidence an important gain with low-level features compared to high-level features for the case of the SVM. However, the results obtained with the LDA classifier show similar results to the ones presented by the SVM classifier with low-level features. The evaluation assesses that LDA reaches higher levels of accuracy than the SVM with high-level features, with a maximum of 96.5% for $\{RIMDB\}$ and seven features. In this case, LDA is more adequate to the distribution of data in the space of characteristics than SVM. However, the best results on average are obtained with SVM classifier and unigrams. One of the main reasons of the success of the SVM classifier based on low-level features is that objective language and subjective language hardly intersect. In practice, this means that a word specific to the subjective (resp. objective) part of the corpus can easily distinguish between objective and subjective texts, although it does not necessarily carry any subjective content. As a consequence, we are in the middle of sentiment classification and topical classification as evidenced in Table 7.3, which respectively presents the accuracy levels for the SVM and the LDA classifiers across domains.

In order to test models across domains, we proposed to train different models based on one domain only at each time and tested the classifiers over all domains together. So, each percentage in Table 7.3 can be expressed as the average results over all data sets. Within this context, best

Table 7.3: Accuracy results across domain.

	$\{MPQA\}$	$\{RIMDB\}$	$\{CHES\}$	$\{WBLOG\}$
Unigrams (SVM)	58.8%	64.4%	69.9%	63.9%
Bigrams (SVM)	57.5%	66.9%	66.5%	62.3%
7 features (SVM)	63.1%	70.45%	70.9%	70.2%
7 features (LDA)	69.4%	73.5%	73.9%	74.6%
6 features (LDA)	67.4%	67.9%	71.6%	73%

results overall were obtained for high-level features with the $\{WBLOG\}$ corpus as training data set and the LDA classifier with an average accuracy of 74.6%, which means that using the model constructed with LDA and $\{WBLOG\}$ training datasets and testing it over itself, $\{RIMDB\}$, $\{MPQA\}$ and $\{CHES\}$, the arithmetic mean of all accuracies is equal to 74.5%. Moreover, the results show that accuracy drops drastically with learning based on unigrams or bigrams. Best results with low-level features are obtained for the $\{CHES\}$ dataset and unigrams with average accuracy of 69.9% . The obtained results confirmed the visual interpretation of the data distribution, observed in Section 5.2.2. As illustrated in figures 5.5, 5.6 and 5.7, high-level features may lead to improved results across domains, as subjective and objective documents are better separated compared to when using low-level features.

In order to evaluate the importance of the level of abstraction of nouns as a clue for subjectivity identification, we propose to test classification accuracy of the models based on the six state-of-the-art features[1] without the level of abstraction of nouns and then compared with the full set of seven features. The experimental results clearly show that using the level of abstraction of nouns as a feature leads to improved performance on subjectivity classification tasks for each of the models.

Although these results are encouraging, new trends in sentiment classification recently appeared using multi-view learning such as Ganchev *et al.* (2008), Wan (2009) who evidenced improved results to cross domains. Within this scope, we proposed in Lambov *et al.* (2010) that high-level and low-level features can be treated as different views.

7.3 Multi-view Classification with Agreement

In this section, we present the results obtained by using the multi-view learning techniques, presented in section 6.2.1, combining high-level and low-level features. First, we present the results obtained by the SAR algorithm, which will be our baseline and then compare them to those obtained with the MAA, BMAA and BMAADR algorithms. All accuracy results were obtained

[1]The 6 features line means that the level of abstraction of nouns was omitted from the seven original high-level features.

from the low-level classifier for $N = P = 2$, after 25 learning iterations.

7.3.1 SAR

In order to better understand the behavior of multi-view learning, we first applied SAR[1] to our data sets. In particular, we used two views generated from a random split of low-level features together with a maximum entropy classifier to learn a domain-independent model. For that purpose, we performed a leave-one-out 5 cross validation, where both labeled and unlabeled examples were provided for the learning process and then new unlabeled examples were classified by the learnt model to evaluate accuracy. So, as we did previously in section 7.2, we proposed to train different models based on one domain only at each time and test the classifiers over all other domains. Thus, the accuracy results presented in Table 7.4 represent average values, which evaluate how well a model can cross different domains.

Table 7.4: SAR accuracy results for low-level features across domains.

	$\{MPQA\}$	$\{RIMDB\}$	$\{CHES\}$	$\{WBLOG\}$
Unigrams	63.7%	77.1%	72.3%	59.7%
Bigrams	59.8%	65.2%	64.9%	62.2%

The results show indeed interesting properties. First, models built upon unigrams constantly outperform models based on bigrams. However, as shown in table 7.2, this is not necessarily true in the case when we used low-level features for in-domain sentiment classification, thus highlighting the fact that combinations of words embody topical domain-specific properties rather than subjective values. Second, higher accuracy is reached compared to section 7.2 with less knowledge. Indeed, the baseline with single-view classification is 74.5% obtained with high-level features and the LDA algorithm with the $\{WBLOG\}$ as a training domain, while 77.1% can be obtained with the SAR algorithm upon a random split of unigrams with the $\{RIMDB\}$ dataset. One great advantage of only using low-level features is the ability to reproduce such experiments on different languages without further resources than just texts. However, a good training data set will have to be produced as the best results are obtained from the manually annotated corpus $\{RIMDB\}$, while the automatically labeled corpus $\{WBLOG\}$ provides worst results.

[1] Which was kindly provided by Ganchev *et al.* (2008).

7.3.2 Merged Agreement Algorithm (MAA)

In this subsection, we present results of the MAA algorithm, combining a first view with 7 high-level features and a second view with low-level features (unigrams or bigrams). The objective of the algorithm is to use the agreement between models without balancing the parameter values of P and N at each iteration, i.e. it is not necessary to have a minimum number of positive or negative documents, which agree on labels for both classifiers. In Table 7.5, we show the results obtained using two SVM classifiers i.e. one for each view. In table 7.6 we show the results obtained using a SVM classifier for the low-level view and a LDA classifier for the high-level view as we know that LDA outperforms SVM for high-level features. The best result is obtained by the combination of two SVM classifiers trained over the $\{CHES\}$ corpus. In this case, the average accuracy across domains reaches 75.55%, which is worse than the SAR best performance of 77.1%. It is also interesting to notice the results of subjective precision and recall in the cases of exclusively objective and subjective datasets ($\{RIMDB\}$ and $\{MPQA\}$). In the movie reviews domain $\{RIMDB\}$ we have very high precision and low recall. This shows that movie reviews domain is more subjective and its use as a model leads to a correct prediction of only a very subjective documents, while others are predicted as objective. Conversely, using the news domain $\{MPQA\}$ as a model leads to low precision and high recall, indicating that this domain is more objective and therefore its use as model predicts correctly only a very objective documents and all other as subjective.

Table 7.5: MAA accuracy results in %, using two SVM classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	69.35%	84.49%	47.4%	64.7%	100%	29.4%
$\{CHES\}$	75.55%	71.74%	84.3%	71.55%	76.84%	61.7%
$\{MPQA\}$	59.05%	54.98%	100%	57.5%	54.05%	100%
$\{RIMDB\}$	63.45%	100%	26.9%	69.85%	100%	39.7%

Table 7.6: MAA accuracy results in %, using one SVM and one LDA classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	69.2%	84.41%	47.1%	64.8%	100%	29.6%
$\{CHES\}$	74.5%	66.83%	97.3%	75.45%	80.41%	67.3%
$\{MPQA\}$	59.6%	55.31%	100%	57.5%	54.05%	100%
$\{RIMDB\}$	63.3%	100%	26.6%	70.15%	100%	40.3%

We illustrate the behavior of each classifier in Figure 7.1 for the method using SVM classifiers

for the two views. The graphic shows that both classifiers improve their accuracy just in the first few iterations and then start to lose in accuracy. This is mainly due to the fact that the unbalanced labeled examples started impairing the performance. In Figure 7.2 we present the Accuracy, Precision and Recall scores for the LLF classifier using unigrams.

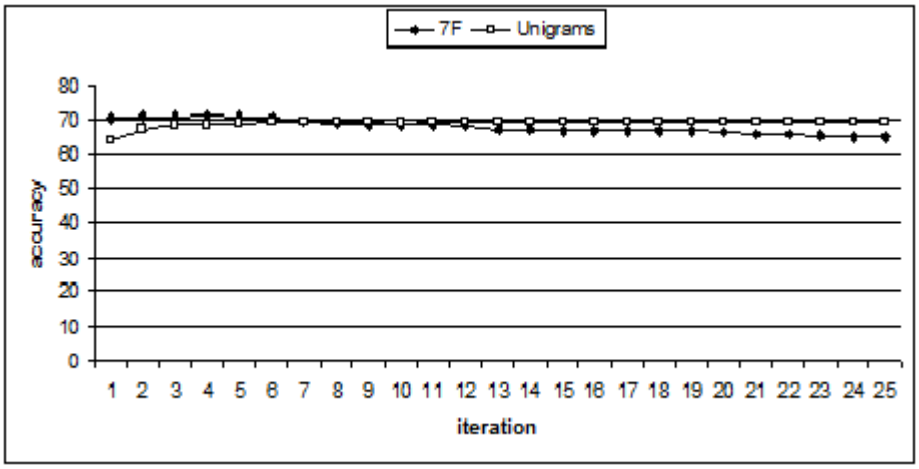

Figure 7.1: HLF and LLF classification accuracies using MAA for the $\{WBLOG\}$ dataset (SVM Unigrams)

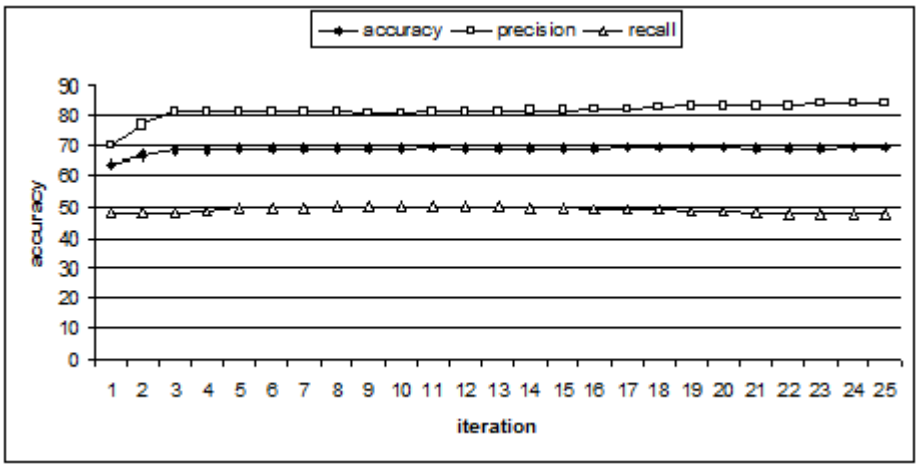

Figure 7.2: Accuracy, Precision and Recall scores using MAA for the $\{WBLOG\}$ dataset - LLF classifier (SVM Unigrams)

7.3.3 Balanced Merged Agreement Algorithm (BMAA)

In this subsection, we present results of the BMAA algorithm, in which the class distribution in the labeled data is maintained by balancing the parameter values of P and N at each iteration, i.e. an equal number of positive and negative documents are added to the labeled data set at each learning iteration. If the number of predicted subjective or objective documents is equal to 0, it is used as a stopping criterion. We illustrate the behavior of each classifier in

Figure 7.3 for the method using only SVM classifiers for the two views. In Figure 7.4 we present the Accuracy, Precision and Recall scores for the LLF classifier (unigrams). Again, just like the previous method, the accuracy of the low-level classifier is improved just in the first few iterations and then the performance of the approach does not change any more. However, unlike the previous method, where the performance starts to decrease, here it remains almost constant for both classifiers. In this method, best accuracy value is obtained by the combination of two SVM classifiers reaching 79.45%, which outperforms SAR best accuracy of 77.1%.

Table 7.7: BMAA accuracy results in %, using two SVM classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	69.65%	81.54%	50.8%	65.15%	99.03%	30.6%
$\{CHES\}$	79.45%	73.1%	93.2%	77.9%	82.52%	70.8%
$\{MPQA\}$	59.35%	55.16%	100%	57.5%	54.05%	100%
$\{RIMDB\}$	65.15%	99.35%	30.5%	76.55%	99.26%	53.5%

Table 7.8: BMAA accuracy results in %, using one SVM and one LDA classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	69.7%	82.83%	49.7%	65.25%	99.04%	30.8%
$\{CHES\}$	78.3%	70.66%	96.8%	77.1%	82.73%	68.5%
$\{MPQA\}$	59.65%	55.34%	100%	57.5%	54.05%	100%
$\{RIMDB\}$	64.8%	99.66%	29.7%	76.1%	99.06%	52.7%

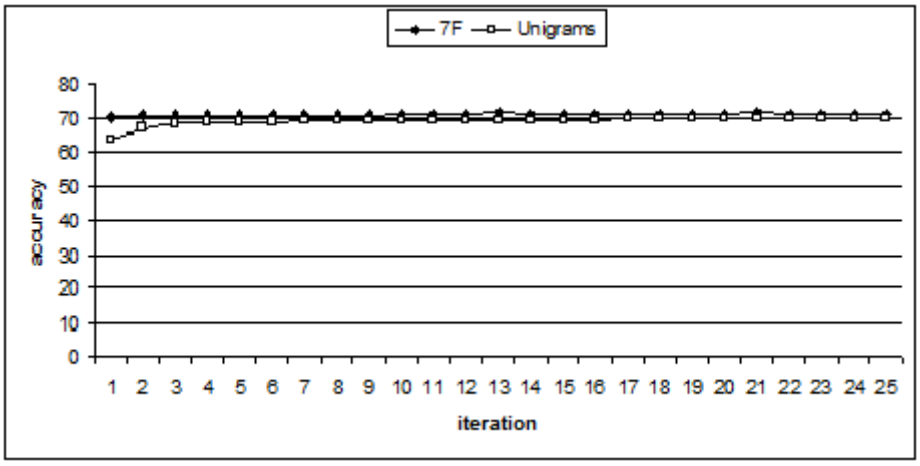

Figure 7.3: HLF and LLF classification accuracies using BMAA for the $\{WBLOG\}$ dataset (SVM Unigrams)

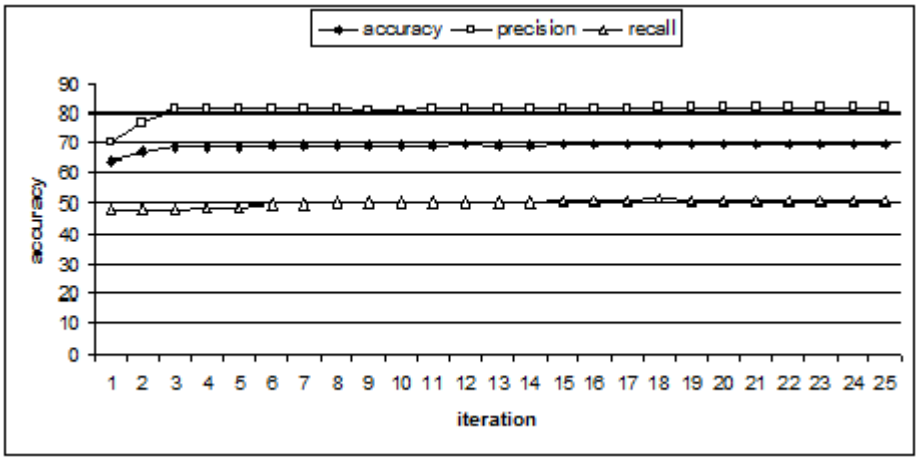

Figure 7.4: Accuracy, Precision and Recall scores using BMAA for the $\{WBLOG\}$ dataset - LLF classifier (SVM Unigrams)

7.3.4 Balanced Merged Agreement Algorithm Using Documents Rank (BMAADR)

In this subsection, we present results of the BMAADR algorithm, in which confidence is taken into account for the agreement. The confidence values are calculated as the average of the positions of each example in sorted lists (sorted by confidence values of each document) for both classifiers. As such, P positive and N negative examples with the best confidence values are added to the labeled data. Here we can see that the behavior of both classifiers is almost identical, but the obtained results are slightly better compared to the ones when the previous two methods are used.

Table 7.9: BMAADR accuracy results in %, using two SVM classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	69.9%	81.79%	51.2%	65.45%	98.74%	31.3%
$\{CHES\}$	80%	73.58%	93.6%	77.15%	80.13%	72.2%
$\{MPQA\}$	59.35%	55.16%	100%	57.5%	54.05%	100%
$\{RIMDB\}$	65.35%	98.73%	31.1%	76.65%	97.67%	54.6%

We illustrate the behavior of each classifier in Figure 7.5 for the method using only SVM classifiers for the two views. In Figure 7.6 we present the Accuracy, Precision and Recall scores for the LLF classifier (unigrams). Again, the best result is obtained from the manually annotated corpus $\{CHES\}$, using SVM classifiers for the two views. In this case, the average accuracy across domains is 80% outperforming the results of all other algorithms using agreement i.e SAR, MAA and BMAA. It is also interesting to notice that in almost all cases, unigram low-level features

Table 7.10: BMAADR accuracy results in %, using one SVM and one LDA classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	69.8%	82.78%	50%	65.45%	98.43%	31.4%
$\{CHES\}$	77.9%	70.31%	96.6%	76.8%	81.38%	69.5%
$\{MPQA\}$	59.7%	55.37%	100%	57.5%	54.05%	100%
$\{RIMDB\}$	65%	98.7%	30.4%	74.9%	98.44%	50.6%

provide better results than bigrams. The only exception is the $\{RIMDB\}$ training set, where using bigrams as low level features drastically improves the results compared to the unigram representation. Moreover, we see that automatically building a labeled data set, such as the $\{WBLOG\}$, can lead to interesting results as it shows the second best performance for unigrams, although it only presents the third best result for the bigram case.

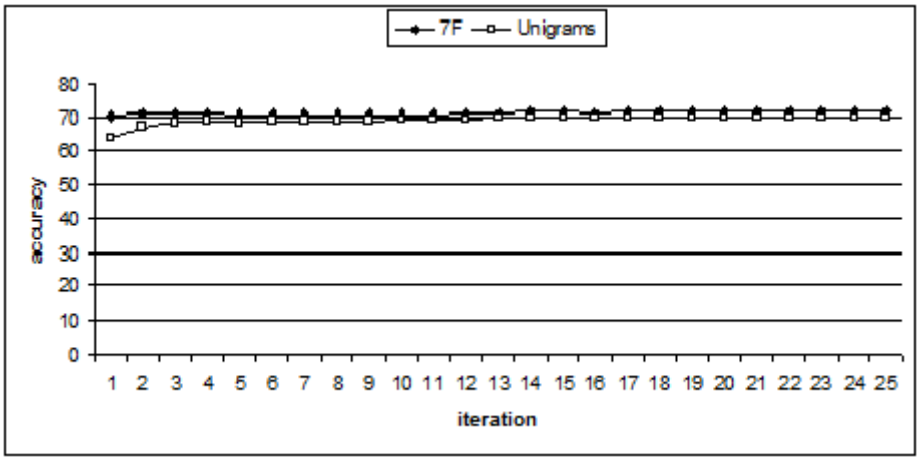

Figure 7.5: HLF and LLF classification accuracies using BMAADR for the $\{WBLOG\}$ dataset (SVM Unigrams)

7.3.5 Problems and Discussions

Multi-view methods typically assume that each view alone can yield a good predictor. In particular, SAR is based on this assumption (Ganchev *et al.* (2008)). But this is not exactly in our case. Indeed, we know that HLF classifiers outperform LLF classifiers across domain. In the proposed methods, we rely on the assumption that the domain independent view based on high-level features will restrict the addition of wrong predicted labels. In fact, these methods suffer because of the weakness of the low-level classifier in their initial state. By using agreement as a rule, it seems that too many examples are filtered out and as such limit the number of possible candidates, which may be interesting labeled new examples. Also, when both classifiers agree they do not learn much more, especially if they agree with high-level of confidence in

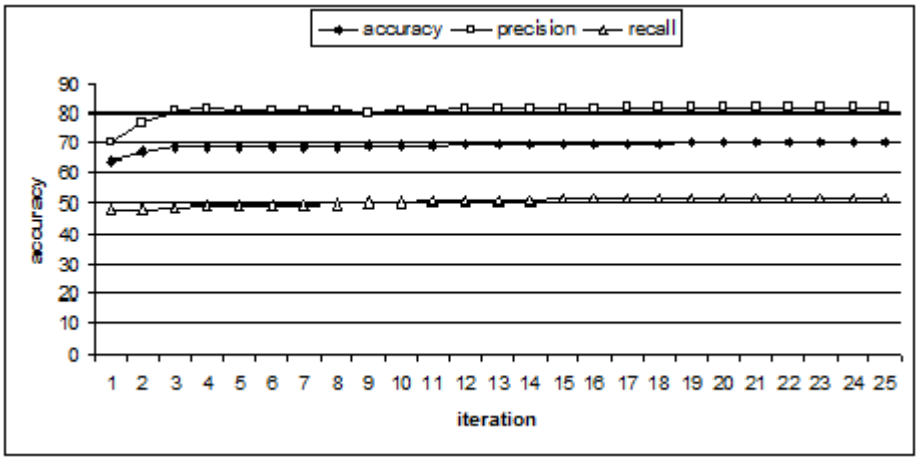

Figure 7.6: Accuracy, Precision and Recall scores using BMAADR for the $\{WBLOG\}$ dataset - LLF classifier (SVM Unigrams)

both classifiers. As a consequence, the accuracy is almost constant for both models based on different views after just a few iterations. In Figure 7.7, we compare the performance of the SAR algorithm with the other methods which are based on the idea of agreement. As shown in the figure, SAR performs better in the cases of exclusively objective and subjective datasets ($\{RIMDB\}$ and $\{MPQA\}$), while in the cases of the other two datasets annotated at document level (i.e. texts do not contain exclusively objective or subjective sentences), the best classification accuracy is obtained by the method using document ranks as confidence values. As a consequence, the BMAADR algorithm is the most performing algorithm for real-world texts situations. In particular, best results overall are obtained by the method using document ranks with two SVM classifiers, for the $\{CHES\}$ dataset reaching accuracy of 80% and thus outperforming the state-of-the-art algorithm SAR that reaches 77.1% trained over the $\{RIMDB\}$ dataset. The worst results overall are obtained by the MAA algorithm. This is mainly due to the fact that the unbalanced labeled examples impair the performance. In this case, best average accuracy across domains reaches 75.55% for the $\{CHES\}$ dataset. Another interesting dependence in the results is that best classification accuracy is reached when using two SVM classifiers compared to that obtained with a LDA and a SVM classifier, although the classifier using high-level features performs better with LDA than with SVM. This leads us to conclude that when we use the agreement between two models, it is generally better to use classifiers based on the same machine learning technique. Another interesting result is the fact that in all three methods the average accuracy of the low-level classifiers could not reach the average accuracy of the high-level classifiers in their initial state. As a consequence, big improvements can not be expected when using agreement between high-level and low-level features as a rule.

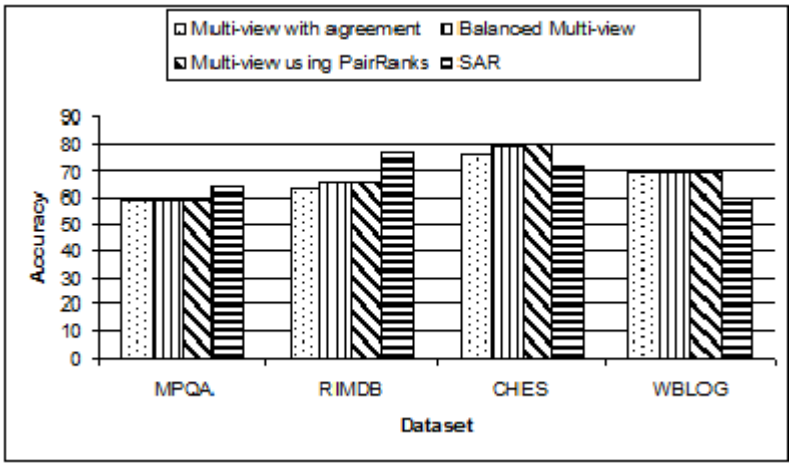

Figure 7.7: Accuracy scores for all methods with agreement: Unigrams

7.4 Two-views Semi-Supervised Learning

In this section, we present the results obtained by using the semi-supervised learning technique i.e. without agreement. First, we present the results obtained by the standard co-training approach and then we compare them with results of the two other methods in which the best classifier guides the selection of the data that is added to the training set in the self-training process. In these methods another one validation dataset is used as a classifier confidence threshold. All accuracy results were obtained from the low-level classifier for $N = P = 2$, after 25 learning iterations.

7.4.1 Co-training

In this subsection, we present results obtained by using the standard co-training algorithm, which combine a first view with 7 high-level features and a second view with low-level features (unigrams or bigrams). As a consequence, we expect that the low-level classifier will gain from the decisions of the high-level classifier and will self-adapt to different domains based on the high results of high-level features to cross domains. In Table 7.11, we show the results obtained using two SVM classifiers i.e. one for each view. In Table 7.12, we show the results obtained using a SVM classifier for the low-level view and a LDA classifier for the high-level view.

The benefit from the high-level features is clear. The best result is obtained by the combination of high-level features with the LDA classifier and unigram low-level features with the SVM classifier trained over the $\{CHES\}$ dataset. In this case, the average accuracy across domain is 91% outperforming SAR best performance of 77.1% over $\{RIMDB\}$ and 80% from the BMAADR algorithm over $\{CHES\}$ training dataset. Indeed, while the high-level classifier accu-

Table 7.11: Co-training accuracy results in %, using two SVM classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	84.4%	83.99%	85%	76.05%	75.56%	77%
$\{CHES\}$	89.25%	85.84%	94%	81.05%	76.98%	88.6%
$\{MPQA\}$	64.55%	58.53%	99.8%	57.5%	54.05%	100%
$\{RIMDB\}$	83.7%	86.31%	80.1%	76.8%	75.67%	79%

Table 7.12: Co-training accuracy results in %, using one SVM and one LDA classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	85.35%	83.38%	88.3%	77.35%	76.95%	78.1%
$\{CHES\}$	91%	86.74%	96.8%	82%	75.93%	93.7%
$\{MPQA\}$	82.5%	60.78%	99.2%	57.55%	54.08%	100%
$\{RIMDB\}$	80.6%	81.22%	79.6%	74.7%	74.36%	75.4%

racy remains almost steady iteration after iteration, the low-level classifier steadily improves its accuracy based on the correct guesses of the high-level classifier. We illustrate the behavior of each classifier in Figure 7.8, for the method using two SVM classifiers. The Accuracy, Precision and Recall scores for the LLF classifier in the co-training approach are compared in Figure 7.9. It is interesting to notice that in all cases, unigrams drastically improve the performance of the co-training as the difference between unigrams or bigrams as second views is huge. Unlike the previous methods which have used agreement between the model, in this method the best results are obtained by the combination of LDA and SVM classifiers. The presented results show that the performance of the LDA classifier with high-level features is significantly better than the performance of the SVM classifier with high-level features and therefore the classifier based on low-level features achieves better results in combination with the LDA classifier. This is clearly shown in Figure 7.10, which compares the accuracy curve of each classifier in the co-training approach.

7.4.2 Guided Semi-Supervised Learning (GSL)

In this subsection, we show the results of the GSL algorithm. The proposed algorithm, unlike the standard co-training approach, uses a validation data set to guide the selection of training candidates, which are labeled by both classifiers in the self-training process. As a consequence, at each learning iteration, P positive and N negative examples with higher confidence values are added to L only from the classifier with the best validation classification accuracy. In Table

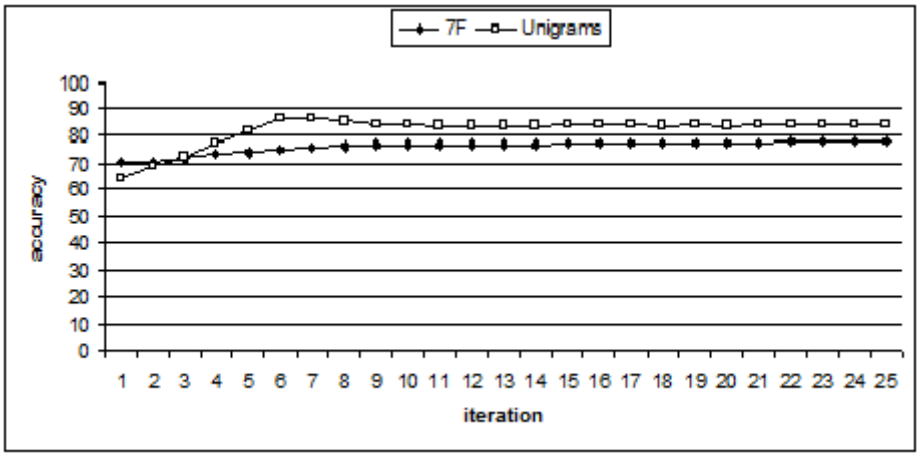

Figure 7.8: HLF and LLF classification accuracies using Co-training for the $\{WBLOG\}$ dataset (SVM Unigrams)

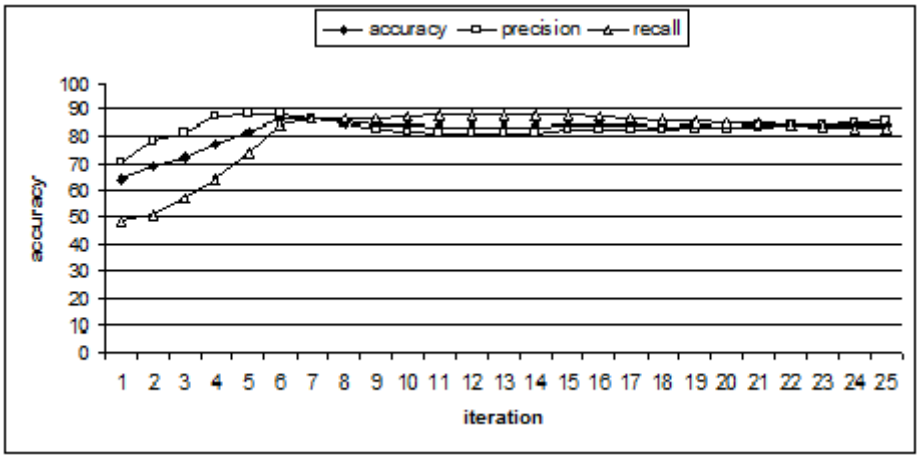

Figure 7.9: Accuracy, Precision and Recall scores using Co-training for the $\{WBLOG\}$ dataset- LLF classifier (SVM Unigrams)

7.13, we show the results obtained with two SVM classifiers. In Table 7.14, we present results obtained with a LDA classifier for the high-level features and a SVM classifier for the low-level ones. The obtained results are slightly worse compared to the results obtained by the co-training approach. In this case, the best average accuracy across domains is obtained by the combination of LDA and SVM classifiers, using unigram model for the $\{CHES\}$ dataset with average accuracy of 90.35%. However the GSL algorithm improves the results obtained over the $\{RIMDB\}$ model compared with the results obtained by using the standart co-training algorithm.

We illustrate the behavior of each classifier in Figure 7.11 for the method using only SVM classifiers for the two views, while in Figure 7.12 we present the Accuracy, Precision and Recall

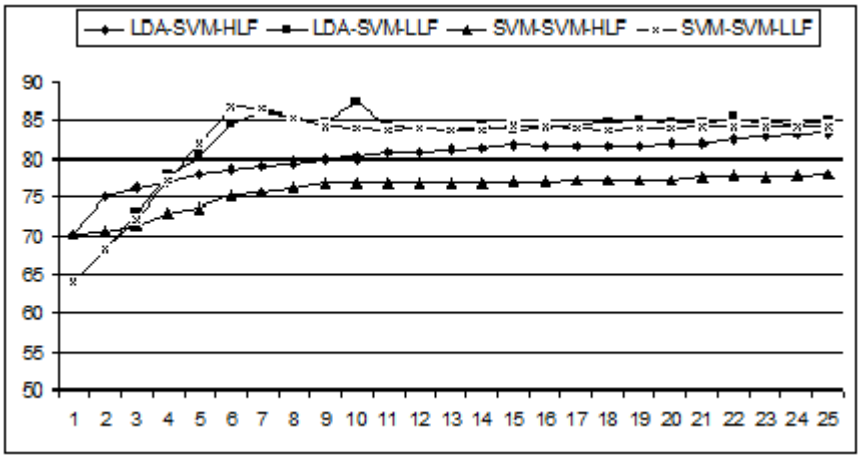

Figure 7.10: Accuracy scores for LDA and SVM classifiers using Co-training for the $\{WBLOG\}$ dataset - Unigrams and 7 high-level features

Table 7.13: GSL accuracy results in %, using two SVM classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	85.45%	85.56%	85.3%	75.95%	75.87%	76.1%
$\{CHES\}$	89.6%	85.36%	95.6%	80.35%	75.83%	89.1%
$\{MPQA\}$	67.45%	60.58%	99.9%	57.5%	54.05%	100%
$\{RIMDB\}$	84.1%	86.59%	80.7%	76.45%	74.79%	79.8%

Table 7.14: GSL accuracy results in %, using one SVM and one LDA classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	85.05%	83.03%	88.1%	77.6%	77.88%	77.1%
$\{CHES\}$	90.35%	86.52%	95.6%	81.8%	76.37%	92.1%
$\{MPQA\}$	82.75%	74.64%	99.2%	57.5%	54.05%	100%
$\{RIMDB\}$	83%	81.07%	86.1%	76.75%	74.03%	82.4%

scores for the LLF classifier (unigrams).

7.4.3 Class-Guided Semi-Supervised Learning (C-GSL)

In this subsection we show the results of the C-GSL algorithm. Here, the main difference with the GSL algorithm is that instead of relying on the global accuracy over the VL data set and choose the corresponding classifier to guide the learning process, we choose the classifier with

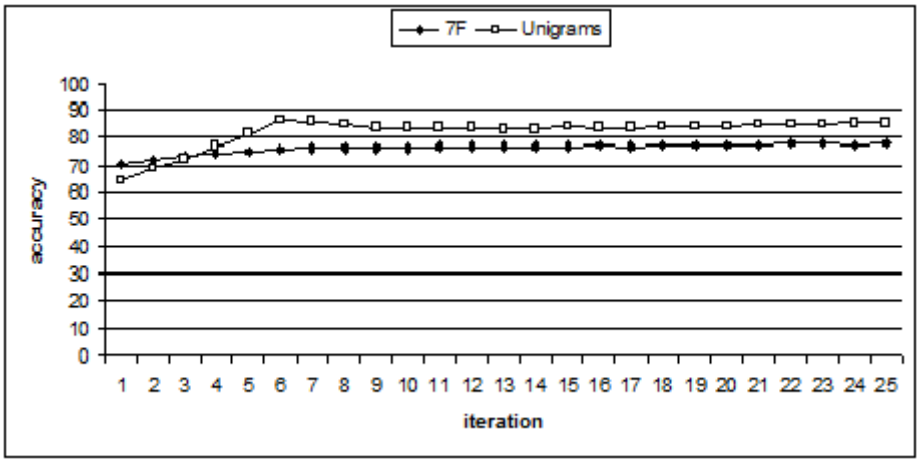

Figure 7.11: HLF and LLF classification accuracies using GSL for the $\{WBLOG\}$ dataset (SVM Unigrams)

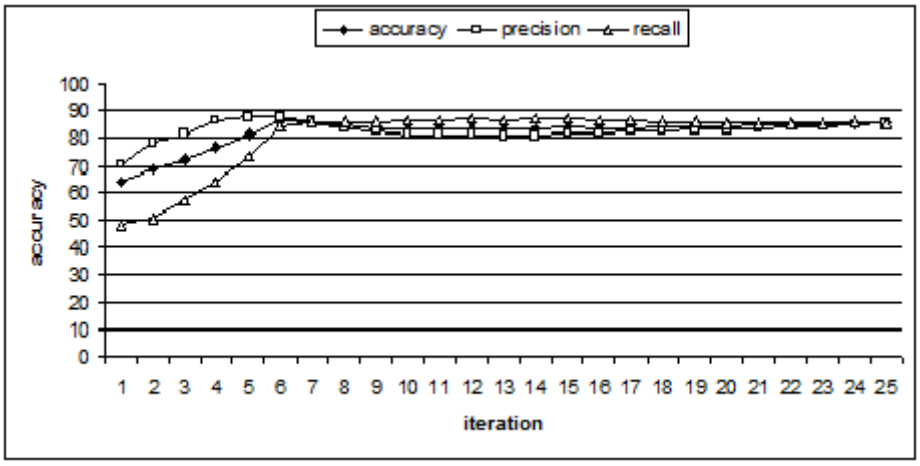

Figure 7.12: Accuracy, Precision and Recall scores using GSL for the $\{WBLOG\}$ dataset - LLF classifier (SVM Unigrams)

higher precision for subjectivity to label the P positive examples from U and the classifier with higher precision for objectivity to label the N negative examples from U. So, at each learning iteration, we compare the subjective precision and the objective precision obtained by each classifier over the validation data set VL and choose the best one for each class.

The results shown in Table 7.15 and Table 7.16 are slightly better than the results obtained from the previous method, but still weaker than the results obtained using the classical co-training approach. The difference in the best accuracy values between the GSL and the C-GSL algorithm is negligible and statistically not relevant. Again, best result is obtained by the combination of LDA and SVM classifiers, using unigram model for the $\{CHES\}$ dataset with average accuracy of 90.45%. We illustrate the behavior of each classifier in Figure 7.13 for the method using only

Table 7.15: C-GSL accuracy results in %, using two SVM classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	86.05%	86.6%	85.3%	75.85%	75.32%	76.9%
$\{CHES\}$	89.65%	85.24%	95.9%	83.1%	77.54%	93.2%
$\{MPQA\}$	67.2%	60.39%	100%	57.5%	54.05%	100%
$\{RIMDB\}$	85.05%	86.93%	82.5%	85.55%	89.99%	80%

Table 7.16: C-GSL accuracy results in %, using one SVM and one LDA classifiers across domains.

Training data set	Unigrams			Bigrams		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
$\{WBLOG\}$	85.65%	83.86%	88.3%	76%	75.54%	76.9%
$\{CHES\}$	90.45%	86.54%	95.8%	81.7%	76.29%	92%
$\{MPQA\}$	82.95%	74.94%	99%	57.5%	54.05%	100%
$\{RIMDB\}$	82.8%	83.27%	82.1%	75.75%	74.22%	78.9%

SVM classifiers for the two views. In Figure 7.14 we present the Accuracy, Precision and Recall scores for the LLF classifier (unigrams).

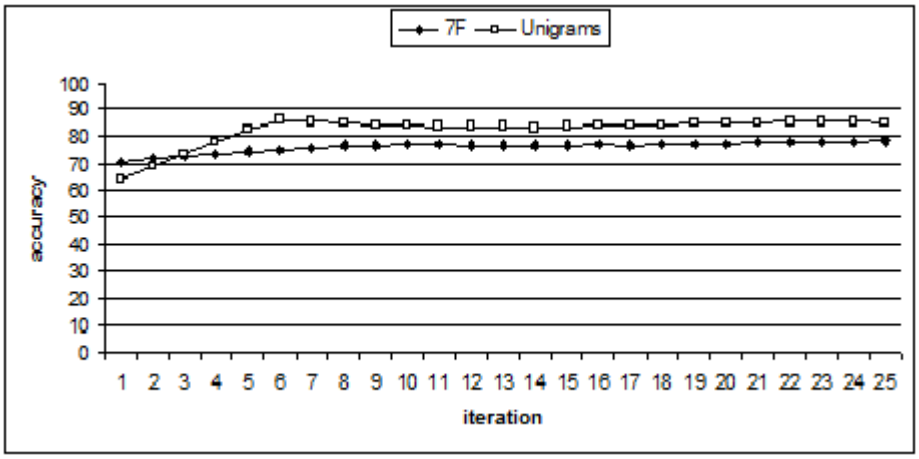

Figure 7.13: HLF and LLF classification accuracies using C-GSL for the $\{WBLOG\}$ dataset (SVM Unigrams)

7.4.4 Problems and Discussions

In order to better understand the situation, we propose a visual analysis of the distribution of the data sets in the space of high-level and low-level features. The goal of this study is to give a visual interpretation of the data distribution to assess how well co-training may perform

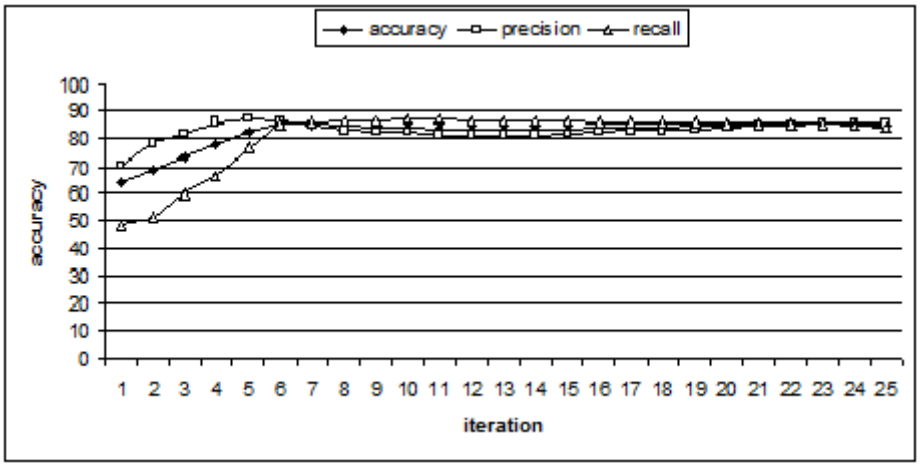

Figure 7.14: Accuracy, Precision and Recall scores using C-GSL for the $\{WBLOG\}$ dataset - LLF classifier (SVM Unigrams)

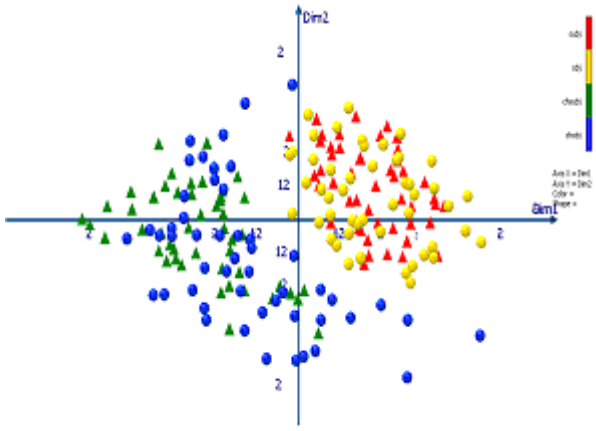

Figure 7.15: Low-level feature representation of subjective (red and green triangles) and objective (blue and yellow dots) texts before co-training.

using high-level and low-level features. If objective and subjective texts can be represented in a distinct way in a reduced space of features, one may expect good classification results. To perform this study, we use a MDS process. For our purpose, we performed the MDS process over pairs of corpora represented by low-level features and high-level features to try to visualize how texts evolve in the multidimensional space before and after co-training.

In Figure 7.15, we graphically represent texts from $\{RIMDB\}$ and $\{CHES\}$ data sets in a reduced space of the low-level features space. Red and green triangles represent subjective texts from $\{RIMDB\}$ and $\{CHES\}$ respectively. Yellow and blue dots represent objective texts from $\{RIMDB\}$ and $\{CHES\}$ respectively. Then, in order to simulate the co-training process, we added new examples from $\{CHES\}$ data set and label them as $\{RIMDB\}$. The obtained vi-

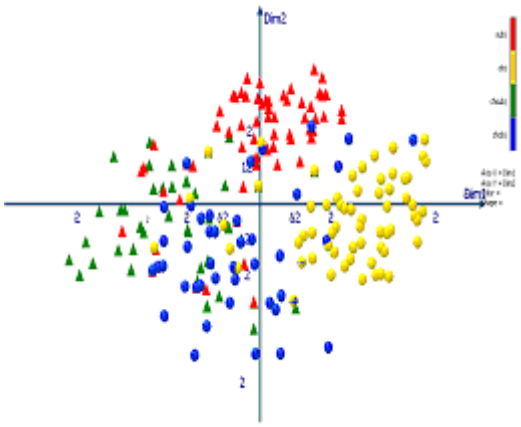

Figure 7.16: Low-level feature representation of subjective (red and green triangles) and objective (blue and yellow dots) texts after co-training.

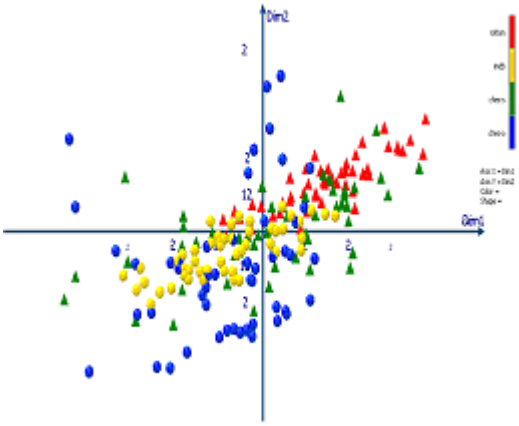

Figure 7.17: High-level feature representation of subjective (red and green triangles) and objective (blue and yellow dots) texts before co-training.

sualization is presented in Figure 7.16 and clearly shows that after co-training subjective and objective texts from different domains tend to approximate. Comparatively, in Figures 7.17 and 7.18, we graphically represent the same texts in a reduced space of the high-level features space. In this experiment, we clearly see that texts do not tend to approximate and remain difficult to separate, as such comforting us in the choice of using low-level classifiers for our classification task using the co-training approach.

In Figure 7.19 we compare the average results obtained by each of the methods without agreement presented above. In particular, best results overall are obtained by the method using co-training with one SVM classifier and one LDA classifier, for the $\{CHES\}$ dataset reaching accuracy of 91%. Here, unlike the methods based on agreement, the results obtained by the combination of high-level features with the LDA classifier and unigram low-level features with

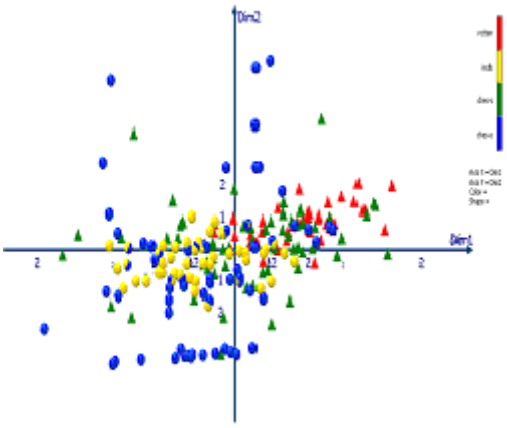

Figure 7.18: High-level feature representation of subjective (red and green triangles) and objective (blue and yellow dots) texts after co-training.

the SVM classifier constantly outperform the results obtained by the combination of two SVM classifiers. This clearly shows that the low-level classifier gain from the better results of the LDA classifier, using high-level features.

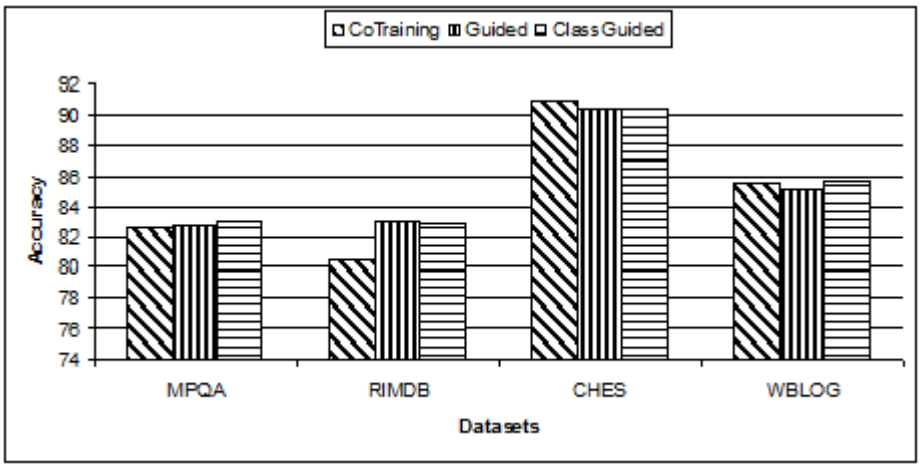

Figure 7.19: Accuracy scores for all methods without agreement: Unigrams

7.5 Three-views Semi-Supervised Learning

In this section, we present the results of the GSL and C-GSL algorithms for 3 views. To do so, we propose to divide high-level features into four different classes: affective words, adjectives (semantically oriented and dynamic), verbs (conjecture, marvel and see) and level of abstraction of nouns. In fact, we propose to combine these feature classes in a way in which each view will be based on different type of information:

(1) Bag-of words representation (unigrams, bigrams),

(2) Semantic information (adjectives, affective words, verbs) and
(3) Conceptual expression of subjectivity (level of abstraction of nouns).

The idea of using three different semantic views is that different features may have different weights in different domains. For example, Lambov *et al.* (2009b) showed that affective words are strong predictors of subjectivity/objectivity in the News domain, adjectives in Web Blogs and verbs in Movie reviews. Thus, the algorithms are more flexible in trying to choose the best examples in the self-training process.
The results presented in Section 7.4 showed that better results could be achieved by using a combination of a SVM classifier for the low-level features and a LDA classifier for the high-level features. Consequently, for the following experiments, we propose to use a LDA for each of the high-level classes of features and a SVM for unigram and bigram features.

7.5.1 Three-views Guided Semi-Supervised Learning

In this subsection, we present the results of the GSL algorithm for 3 views (Algorithm 11). The proposed technique uses a validation data set to guide the selection of training candidates, which are labeled by each classifier in the self-training process. At the end of each learning iteration, each classifier must classify the validation test set, which contains labeled examples from a target domain. Finally, P positive and N negative examples with higher confidence values are added to L only from the classifier with the best validation classification accuracy. Comparative results obtained through each combination of views are presented in Table 7.17. The best result is obtained by the combination of affective words and level of abstraction with the LDA classifier and unigram low-level features with the SVM classifier trained over the manually annotated corpus $\{CHES\}$. Higher accuracy compared to the same method, but with two views is reached with combination of affective words, level of abstraction and unigrams. Indeed, the best result for the $\{CHES\}$ dataset with two views is 90.35% while 90.75% can be obtained with the the same method, but using three views. Again like in the GSL method for two views, using unigrams as low level features drastically improve the accuracy compared to results obtained using bigrams as a second view.

We illustrate the behavior of each classifier in Figure 7.20 and the Accuracy, Precision and Recall scores for the low-level classifier in Figure 7.21. We can see that the accuracy of the classifier using level of abstraction consistently outperforms the accuracy of the classifier using affective words as a feature, therefore improving the accuracy of the low-level classifier is due to only one of the two classes of high-level features.

Table 7.17: GSL accuracy results in % with 3 views.

First view	Second view	Third View	$\{RIMDB\}$	$\{MPQA\}$	$\{CHES\}$	$\{WBLOG\}$
Adjectives (LDA)	Unigr. (SVM)	LA (LDA)	83%	83.6%	89.4%	84.7%
Affect.Words (LDA)	Unigr. (SVM)	LA (LDA)	80.7%	78.6%	90.75%	84.6%
Verbs (LDA)	Unigr. (SVM)	LA (LDA)	82.35%	81.2%	87.55%	83.5%
Adjectives (LDA)	Bigr. (SVM)	LA (LDA)	77.8%	57.65%	82.4%	77.45%
Affect.Words (LDA)	Bigr. (SVM)	LA (LDA)	75.5%	57.7%	82.5%	77.6%
Verbs (LDA)	Bigr. (SVM)	LA (LDA)	76.95%	57.55%	82.55%	76.75%

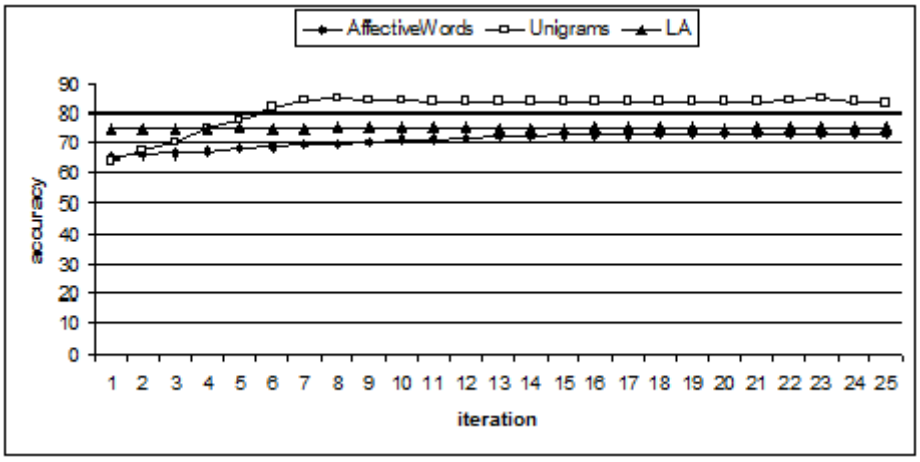

Figure 7.20: HLF and LLF classification accuracies using GSL with 3 views for the $\{WBLOG\}$ dataset (affective words and level of abstraction with the LDA classifier and unigrams with the SVM classifier)

7.5.2 Three-views Class-guided Semi-Supervised Learning

In this subsection, we present the results of the C-GSL algorithm for 3 views (Algorithm 12). The proposed technique uses a validation data set to guide the selection of training candidates, which are labeled by each classifier in the self-training process. This method is similar to the previous one, but at the end of each learning iteration, it is not the accuracy of both classifiers we compare, but the subjective precision and the objective precision obtained from the classification of the validation dataset. So, P positive examples with higher confidence values are added from the classifier with the best subjective precision and N negative examples with higher confidence values are added from the classifier with the best objective precision. Comparative results obtained through each combination of views are presented in Table 7.18. Again the best result is obtained by the combination of affective words and level of abstraction with the LDA classifier and unigram low-level features with the SVM classifier trained over the

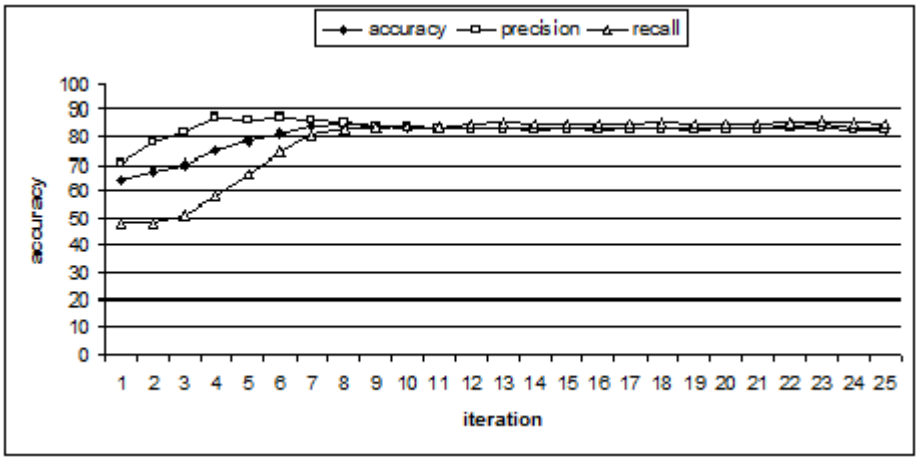

Figure 7.21: Accuracy, Precision and Recall scores using GSL with 3 views for the $\{WBLOG\}$ dataset (affective words and level of abstraction with the LDA classifier and unigrams with the SVM classifier) - LLF classifier

manually annotated corpus $\{CHES\}$. The proposed semi-supervised approach outperforms the previous one, due to the fact that the low-level classifier improves its accuracy using new labeled data from each of the other two classifiers. In this case, the best average accuracy across domain is 91.3% outperforming the results obtained by all methods applied so far. The results of these experiments show that using more than two views may lead to improved results over two views. We illustrate the behavior of each classifier in Figure 7.22 and the Accuracy, Precision and Recall scores for the low-level classifier in Figure 7.23.

Table 7.18: C-GSL accuracy results in % with 3 views.

First view	Second view	Third View	$\{RIMDB\}$	$\{MPQA\}$	$\{CHES\}$	$\{WBLOG\}$
Adjectives (LDA)	Unigr. (SVM)	LA (LDA)	83.15%	76.65%	90.25%	85.2%
Affect.Words (LDA)	Unigr. (SVM)	LA (LDA)	83%	77.8%	91.3%	84.8%
Verbs (LDA)	Unigr. (SVM)	LA (LDA)	83.55%	75.85%	88.6%	84.95%
Adjectives (LDA)	Bigr. (SVM)	LA (LDA)	75.85%	57.55%	81.8%	76.8%
Affect.Words (LDA)	Bigr. (SVM)	LA (LDA)	73.8%	57.8%	82.05%	77.6%
Verbs (LDA)	Bigr. (SVM)	LA (LDA)	75.6%	57.35%	81.6%	76.8%

The precision curve of each classifier are compared in Figure 7.24, in order to better understand the difference with the previous method. Here, just like in the previous method the accuracy of the classifier using level of abstraction consistently outperforms the accuracy of the classifier using affective words (Figure 7.22). But this is not the case when we compare subjective and objective precision separately, as shown in Figure 7.24. In the case of the C-GSL algorithm, for

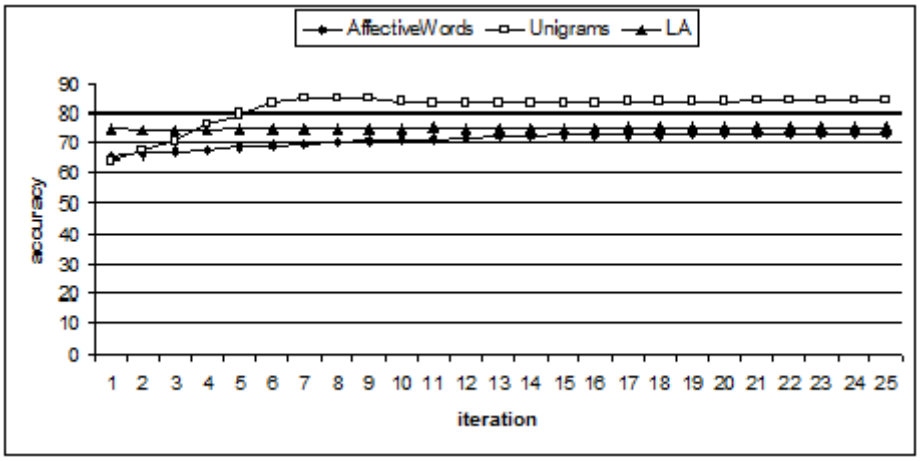

Figure 7.22: HLF and LLF classification accuracies using C-GSL with 3 views for the $\{WBLOG\}$ dataset (affective words and level of abstraction with the LDA classifier and unigrams with the SVM classifier)

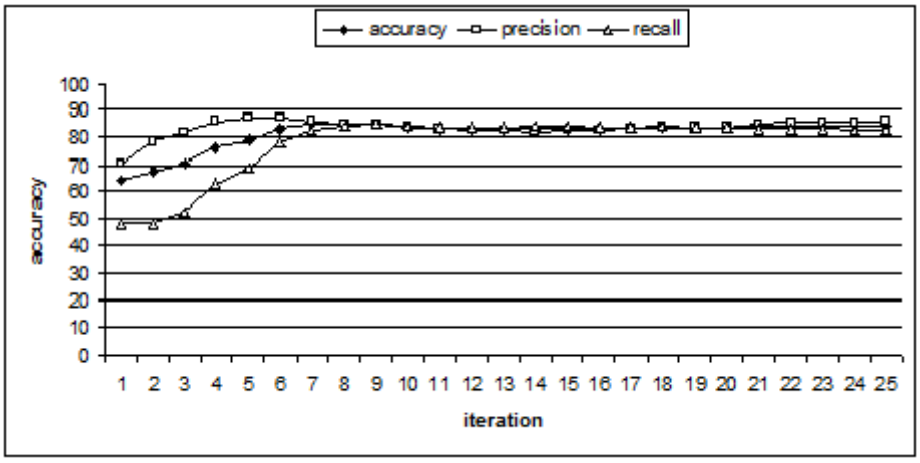

Figure 7.23: Accuracy, Precision and Recall scores using C-GSL with 3 views for the $\{WBLOG\}$ dataset (affective words and level of abstraction with the LDA classifier and unigrams with the SVM classifier) - LLF classifier)

the objective case, the best precision results at the beginning of the learning process are given by the Affective words view and then the best classifier is always the one based on unigrams. In this case, the level of abstraction view does not play any role. On the contrary, for the subjective part, the level of abstraction view "guides" the learning process until the unigram overtakes its precision levels at the sixth iteration. In this case, the Affective words view is useless for the subjective learning process. With these results, we clearly understand that the C-GSL algorithm for 3-views outperforms the GSL algorithm and using more than two views provide more accurate decisions when adding new examples into labeled data.

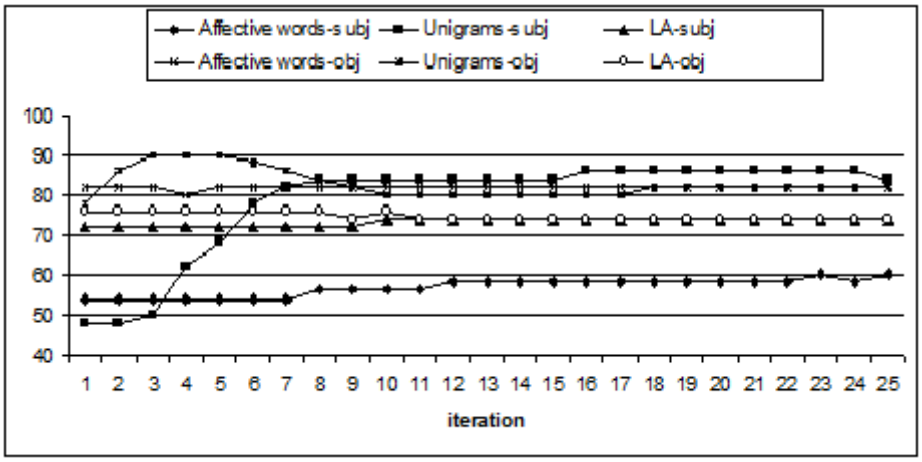

Figure 7.24: Subjective and objective precision scores using C-GSL with 3 views for the $\{WBLOG\}$ dataset (affective words and level of abstraction with the LDA classifier and unigrams with the SVM classifier) - LLF classifier)

Chapter 8

Conclusions and Future Work

8.1 Conclusions

Sentiment classification is a domain specific problem i.e. classifiers trained in one domain do not perform well in others. At the same time, sentiment classifiers need to be customizable to new domains in order to be useful in practice such as the Web. Within this context, different studies have been emerging to tackle cross-domain sentiment classification. Moreover, most of the studies have been focusing on polarity although subjectivity is a much more complex linguistic phenomenon as explained in Boiy *et al.* (2007). For that purpose, we presented different experiments based on single-view, multi-view and semi-supervised learning algorithms, using high-level and low-level features, and the results presented in Chapter 7 can be viewed as very encouraging. To summarize what has been accomplished within this scope, we will report the main issues of each chapter in the remainder of this section.

In Chapter 2, we traced some of the various efforts to classify the language of subjectivity, a goal which has become the focus of considerable interests and investments for the last two decades. In Chapter 3, we presented the relevant related work in the area of sentiment classification, giving detailed descriptions and addressing some critical aspects of each approach. In Chapter 4, we proposed the assumption, which states that texts from Wikipedia should embody objectivity while Weblogs should convey subjective contents. In order to prove this assumption, we proposed an exhaustive evaluation based on (1) the Rocchio classification method (Rocchio (1971)) for different part-of-speech tag levels and (2) language modeling. Thanks to this analysis, we were able to automatically build large data sets of learning examples based on common sense judgments. In Chapter 5, we presented and evaluated the features we have used in our experiments. In particular, we proposed a new feature based on the level of abstraction of nouns, which leads to improved results for sentiment classification. We also proposed feature selection and visualization techniques in order to evaluate how well the datasets can be represented in the given space of features. An exhaustive evaluation showed that (1) the level of abstraction of nouns is a strong clue to identify subjective texts across domains, (2) high-level features allow to learn enhanced cross-domain models. Moreover, our experimental results showed that

using levels of abstraction of nouns as a feature leads to improved performance on subjectivity classification tasks. In Chapter 6, we gave a formal definition of single-view, multi-view supervised and semi-supervised strategies, which we used for sentiment classification. We described different methodologies to combine high-level and low-level features, based on the hypothesis that the low-level classifiers will gain from the decisions of the high-level classifiers and will self-adapt to different domains based on the high results of high-level features to cross domains. Finally, in Chapter 7, we demonstrated the usefulness of the developed approaches through an exhaustive evaluation showing significant results. The experiments showed that using more than two views can lead to improved results over two views using guided semi-supervised learning strategies i.e. by dividing high-level features into different sets, thus providing new views. The results showed the effectiveness of the proposed approach. Best results showed accuracy of 91.3% across domains with the 3-views Class-Guided Algorithm obtained over the $\{CHES\}$ dataset compared to 77.1% for the SAR algorithm proposed by Ganchev *et al.* (2008) and 74.5% for single view classification with LDA proposed by Lambov *et al.* (2009a). Moreover, the results produced via automatic construction of data sets ($\{WBLOG\}$) got near, on average, to the best cross-domains classifiers reaching accuracy levels of 86.05%, the second best result framework. A direct application of this study is to automatically produce data sets for other languages than English and allow classification of multilingual opinionated texts. Indeed, building data sets for the classification of opinionated texts can be done automatically, at the document level, just by downloading Weblogs for the subjective part and Wikipedia texts for the objective part. Thus labor-intensive and time-consuming work can be leveraged.

8.2 Future Works

The work presented in this thesis showed that high accuracy could be achieved through the use of semi-supervised learning algorithms using high-level and low-level features. However, many extensions to this work still need to be carried out. On the first hand, the SAR algorithm must be adapted to accept views with different feature types so that an impartial evaluation can be carried out. Indeed, the well-defined mathematical background of SAR (which makes it one of the reference multi-view learning algorithms in the field) together with results obtained just on low-level features makes us think that improved results can be obtained with little adaptations. We are actually working on this issue with João V. Graça, co-author of Ganchev *et al.* (2008).

On the second hand, we showed that using three views can lead to improved results over two views. However, the SAR algorithm is only defined for two views. So, we mathematically defined its formalism for 3 views and experiments will soon be carried out to verify its behavior. This work is being carried out in collaboration with Guillaume Cleuziou and Lionel Martin from

the University of Orleans (France). The proof is presented in the appendix.

Finally, and certainly the most important point of this research, is to leverage the necessity of pre-existing mostly manually built linguistic resources. Indeed, one of our main research directions focuses on the development of language- and domain-independent applications. However, the solutions proposed for sentiment classification are far from our ultimate goal. Nevertheless, this issue is already being studied. In fact, we need to turn all language-dependent features into language-independent features, or at least propose language-independent methodologies to automatically build linguistic resources. In particular, the level of abstraction of words, which is one of the most relevant features, can be obtained by the methodology to assess the level of generality/specificity between words proposed in Dias *et al*. (2008).

Appendix A

Mathematical Logic

A.1 Propositional Logic

Proposition 2. *SAR 3-Views Problem :*

$\min_{q \in Q} KL(q(y_1, y_2, y_3)||p_1(y_1)p_2(y_2)p_3(y_3))$
where $Q = \{q : \sum_{y_1 y_2} q(y_1, y_2, y) = \sum_{y_1 y_3} q(y_1, y, y_3) = \sum_{y_2 y_3} q(y, y_2, y_3), \forall y\}$

Solution:

$\min_{q \in Q} KL(q(y_1, y_2, y_3)||p_1(y_1)p_2(y_2)p_3(y_3))$ = $\min_{q \in Q} \sum_{y_1, y_2, y_3} q(y_1, y_2, y_3) \log \frac{q(y_1, y_2, y_3)}{p_1(y_1)p_2(y_2)p_3(y_3)}$
where

$\forall y : \sum_{y_1 y_2} q(y_1, y_2, y) - \sum_{y_1 y_3} q(y_1, y, y_3) = 0$ *(Constraint 1)*
$\sum_{y_1 y_2} q(y_1, y_2, y) - \sum_{y_2 y_3} q(y, y_2, y_3) = 0$ *(Constraint 2)*
$\sum_{y_1 y_2 y_3} q(y_1, y_2, y_3) = 1$ *(Constraint 3)*

This problem can be solved by the following Lagrangian :

$L(q, \lambda_\epsilon, \lambda'_\epsilon, \lambda) = \sum_{y_1, y_2, y_3} q(y_1, y_2, y_3) \log \frac{q(y_1, y_2, y_3)}{p_1(y_1)p_2(y_2)p_3(y_3)}$
$+ \sum_y \lambda_y (\sum_{y_1 y_2} q(y_1, y_2, y) - \sum_{y_1 y_3} q(y_1, y, y_3))$
$+ \sum_y \lambda'_y (\sum_{y_1 y_2} q(y_1, y_2, y) - \sum_{y_2 y_3} q(y, y_2, y_3))$
$+ \lambda (\sum_{y_1 y_2 y_3} q(y_1, y_2, y_3) - 1)$

This Lagrangian aims to minimize the primal, which is to find value where the derivative is 0. We note $q_{ijk} = q(y_i, y_j, y_k)$*, the partial derivative is as follows:*

$$\frac{\delta L(q, \lambda_\epsilon, \lambda'_\epsilon, \lambda)}{\delta q_{ijk}} = \frac{\delta(q_{ijk} \log \frac{q_{ijk}}{p_1(y_i)p_2(y_j)p_3(y_k)})}{\delta q_{ijk}} + \frac{\delta(\lambda_{y_k} q_{ijk} - \lambda_{y_j} q_{ijk})}{\delta q_{ijk}} + \frac{\delta(\lambda'_{y_k} q_{ijk} - \lambda'_{y_i} q_{ijk})}{\delta q_{ijk}} + \frac{\delta \lambda (q_{ijk} - 1)}{\delta q_{ijk}}$$

$$\frac{\delta L(q, \lambda_\epsilon, \lambda'_\epsilon, \lambda)}{\delta q_{ijk}} = \log q_{ijk} + 1 - \log(p_1(y_i)p_2(y_j)p_3(y_k)) + (\lambda y_k - \lambda y_j) + (\lambda' y_k - \lambda' y_i) + \lambda$$

to solve the problem the derivative should be 0.

$\frac{\delta L(q,\lambda_\epsilon,\lambda'_\epsilon,\lambda)}{\delta q_{ijk}} = 0$

$\Leftrightarrow \log q_{ijk} + 1 - \log(p_1(y_i)p_2(y_j)p_3(y_k)) + (\lambda y_k - \lambda y_j) + (\lambda' y_k - \lambda' y_i) + \lambda = 0$

$\Leftrightarrow \log q_{ijk} = \log(p_1(y_i)p_2(y_j)p_3(y_k)) - (\lambda y_k - \lambda y_j) - (\lambda' y_k - \lambda' y_i) - \lambda - 1$

$\Leftrightarrow q_{ijk} = (p_1(y_i)p_2(y_j)p_3(y_k))e^{(\lambda y_j - \lambda y_k) + (\lambda' y_i - \lambda' y_k) - \lambda - 1}$

We note $\Delta_{ijk} = (\lambda y_j - \lambda y_k) + (\lambda' y_i - \lambda' y_k) - \lambda - 1$

As we want to find a function $q()$*, we present* q_{ijk} *as function* p_1, p_2, p_3 *in the equation to minimize. And so, we have:*

$Q(q,\lambda_\epsilon,\lambda'_\epsilon,\lambda) = \min_q L(q,\lambda_\epsilon,\lambda'_\epsilon,\lambda)$
$\Leftrightarrow Q(q,\lambda_\epsilon,\lambda'_\epsilon,\lambda) = \sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{\Delta_{123}}\Delta_{123}$
$+\sum_y \lambda y(\sum_{y_1,y_2} p_1(y_1)p_2(y_2)p_3(y).e^{\Delta_{12y}} - \sum_{y_1,y_3} p_1(y_1)p_2(y)p_3(y_3).e^{\Delta_{1y3}})$
$+\sum_y \lambda' y(\sum_{y_1,y_2} p_1(y_1)p_2(y_2)p_3(y).e^{\Delta_{12y}} - \sum_{y_2,y_3} p_1(y)p_2(y_2)p_3(y_3).e^{\Delta_{y23}})$
$+\lambda(\sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{\Delta_{123}} - 1)$

$\Delta_{123} = (\lambda y_2 - \lambda y_3) + (\lambda' y_1 - \lambda' y_3) - \lambda - 1$
$\Delta_{12y} = (\lambda y_2 - \lambda y) + (\lambda' y_1 - \lambda' y) - \lambda - 1$
$\Delta_{y23} = (\lambda y_2 - \lambda y_3) + (\lambda' y - \lambda' y_3) - \lambda - 1$

We develop the following equation:

$A = [\sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{(\lambda y_2 - \lambda y_3) + (\lambda' y_1 - \lambda' y_3) - \lambda - 1}] * [(\lambda y_2 - \lambda y_3) + (\lambda' y_1 - \lambda' y_3) - \lambda - 1]$

$= (\lambda y_2 - \lambda y_3).\sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{\Delta_{123}}$
$+(\lambda' y_1 - \lambda' y_3).\sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{\Delta_{123}}$
$+(-\lambda - 1).\sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{\Delta_{123}}$

We note $B = p_1(y_1)p_2(y_2)p_3(y_3).e^{\Delta_{123}}$

$$\Rightarrow \lambda y_2 \sum_{y_1,y_2,y_3} B - \lambda y_3 \sum_{y_1,y_2,y_3} B + \lambda' y_1 \sum_{y_1,y_2,y_3} B - \lambda' y_3 \sum_{y_1,y_2,y_3} B + (-\lambda - 1). \sum_{y_1,y_2,y_3} B$$

$$= \sum_{y} \lambda y. \sum_{y_1,y_3} p_1(y_1)p_2(y)p_3(y_3).e^{\Delta_{1y3}} - \sum_{y} \lambda y. \sum_{y_1,y_2} p_1(y_1)p_2(y_2)p_3(y).e^{\Delta_{12y}}$$
$$+ \sum_{y} \lambda' y. \sum_{y_2,y_3} p_1(y)p_2(y_2)p_3(y_3).e^{\Delta_{y23}} - \sum_{y} \lambda' y. \sum_{y_1,y_2} p_1(y_1)p_2(y_2)p_3(y).e^{\Delta_{12y}}$$
$$+ (-\lambda - 1). \sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{\Delta_{123}}$$

If we substitute A *in* $Q(q, \lambda_\epsilon, \lambda'_\epsilon, \lambda)$ *will obtain:*

$$Q(q, \lambda_\epsilon, \lambda'_\epsilon, \lambda) = - \sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{\Delta_{123}} - \lambda$$
$$= - \sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{[(\lambda y_2 - \lambda y_3) + (\lambda' y_1 - \lambda' y_3) - \lambda - 1]} - \lambda$$

$$Q(q, \lambda_\epsilon, \lambda'_\epsilon, \lambda) = -e^{-\lambda - 1} \sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{[(\lambda y_2 - \lambda y_3) + (\lambda' y_1 - \lambda' y_3)]} - \lambda$$

From Lagrangian, minimizing the primal Q *over the primal variable* q*, should maximize* $Q(q, \lambda_\epsilon, \lambda'_\epsilon, \lambda)$ *over Lagrangian coefficient. The dual problem can be solved as follows:*

$$\max_{\lambda_\epsilon, \lambda'_\epsilon, \lambda} Q(q, \lambda_\epsilon, \lambda'_\epsilon, \lambda)$$

$$\Leftrightarrow \max_{\lambda_\epsilon, \lambda'_\epsilon, \lambda} -e^{-\lambda - 1} \sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{[(\lambda y_2 - \lambda y_3) + (\lambda' y_1 - \lambda' y_3)]} - \lambda$$

We note $\beta = \sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{(\lambda y_2 - \lambda y_3) + (\lambda' y_1 - \lambda' y_3)}$

And so, the problem is to find the points where derivative is 0

$$\max_{\lambda_\epsilon, \lambda'_\epsilon, \lambda} -e^{-\lambda - 1}.\beta - \lambda$$

First the derivative is λ*.*

$$\frac{\delta Q(q, \lambda_\epsilon, \lambda'_\epsilon, \lambda)}{\delta \lambda} = \frac{\delta(-e^{-\lambda - 1}.\beta - \lambda)}{\delta \lambda} = e^{-\lambda - 1}.\beta - 1$$

Then, we look for points where the derivative is 0.

$$\frac{\delta Q(q,\lambda_\epsilon,\lambda'_\epsilon,\lambda)}{\delta\lambda} = 0$$

$$\Leftrightarrow e^{-\lambda-1}.\beta - \lambda = 0$$

$$\Leftrightarrow e^{-\lambda-1} = \frac{1}{\beta}$$

$$\Leftrightarrow -\lambda - 1 = -\log\beta$$

$$\Leftrightarrow \lambda = \log\beta - 1$$

Now the problem is the following:

$$\max_{\lambda_\epsilon,\lambda'_\epsilon,\lambda} -e^{-\lambda-1}.\beta - \lambda$$

$$\Leftrightarrow \max_{\lambda_\epsilon,\lambda'_\epsilon} -e^{-(\log\beta-1)-1}.\beta - (\log\beta - 1)$$

$$\Leftrightarrow \max_{\lambda_\epsilon,\lambda'_\epsilon} -e^{-\log\beta}.\beta - \log\beta + 1$$

$$\Leftrightarrow \max_{\lambda_\epsilon,\lambda'_\epsilon} -\log\beta$$

$$\Leftrightarrow \max_{\lambda_\epsilon,\lambda'_\epsilon} -\log(\sum_{y_1,y_2,y_3} p_1(y_1)p_2(y_2)p_3(y_3).e^{(\lambda y_2-\lambda y_3)+(\lambda' y_1-\lambda' y_3)})$$

$$\Leftrightarrow \max_{\lambda_\epsilon,\lambda'_\epsilon} -\log(\sum_{y_1,y_2,y_3} p_1(y_1).e^{\lambda' y_1}.p_2(y_2).e^{\lambda y_2}p_3(y_3).e^{-\lambda y_3-\lambda' y_3})$$

In our case λ_ϵ and λ'_ϵ are vectors such that:

$$\lambda_\epsilon = (\lambda_{y_1}, \lambda_{y_2}, \lambda_{y_3})$$

$\lambda'_\epsilon = (\lambda'_{y_1}, \lambda'_{y_2}, \lambda'_{y_3})$

In fact there is a parameter for each view and each class.

From the hypothesis we know that we should expect the following constraints:

$\sum_{y_1 y_2} q(y_1, y_2, y) - \sum_{y_1 y_3} q(y_1, y, y_3) = 0$ *(constraint 1)*
$\sum_{y_1 y_2} q(y_1, y_2, y) - \sum_{y_2 y_3} q(y, y_2, y_3) = 0$ *(constraint 2)*
$\sum_{y_1 y_2 y_3} q(y_1, y_2, y_3) - 1 = 0$ *(constraint 3)*

We also know that the optimum of the primal is:

$$q(y_1, y_2, y_3) = p_1(y_1) p_2(y_2) p_3(y_3).e^{(\lambda y_2 - \lambda y_3) + (\lambda' y_1 - \lambda' y_3) - \lambda - 1}$$

$$= p_1(y_1).e^{\lambda' y_1}.p_2(y_2).e^{\lambda y_2} p_3(y_3).e^{-\lambda y_3 - \lambda' y_3}.e^{-\lambda - 1}$$

Now we can change the constraints as follows:

(Constraint 1) $\Leftrightarrow \sum_{y_1 y_2} q(y_1, y_2, y) - \sum_{y_1 y_3} q(y_1, y, y_3) = 0, \forall y$

$$\Leftrightarrow \sum_{y_1 y_2} q(y_1, y_2, y) = \sum_{y_1 y_3} q(y_1, y, y_3), \forall y$$

$$\Leftrightarrow (\sum_{y_1} p_1(y_1).e^{\lambda' y_1})(\sum_{y_2} p_2(y_2).e^{\lambda y_2}).p_3(y).e^{-\lambda y - \lambda' y} = (\sum_{y_1} p_1(y_1).e^{\lambda' y_1})(\sum_{y_3} p_3(y_3).e^{-\lambda y_3 - \lambda' y_3}).p_2(y).e^{\lambda y}$$

(Constraint 2) $\Leftrightarrow \sum_{y_1 y_2} q(y_1, y_2, y) = \sum_{y_2 y_3} q(y, y_2, y_3), \forall y$

$$\Leftrightarrow (\sum_{y_1} p_1(y_1).e^{\lambda' y_1})(\sum_{y_2} p_2(y_2).e^{\lambda y_2}).p_3(y).e^{-\lambda y - \lambda' y} = (p_1(y).e^{\lambda' y})(\sum_{y_2} p_2(y_2).e^{\lambda y_2}).(\sum_{y_3} p_3(y_3).e^{-\lambda y_3 - \lambda' y_3})$$

Now if we note :

$$\beta_1 = \sum_{y_1} p_1(y_1).e^{\lambda' y_1}$$

$$\beta_2 = \sum_{y_2} p_2(y_2).e^{\lambda y_2}$$

$\beta_3 = \sum_{y_3} p_3(y_3).e^{-\lambda y_3 - \lambda' y_3}$

$\Rightarrow \beta = \beta_1.\beta_2.\beta_3$

We obtain the following constraints:

(Constraint 1) $\Leftrightarrow \forall y,\ \beta_1.\beta_2.p_3(y).e^{-\lambda y - \lambda' y} = \beta_1.\beta_3.p_2(y).e^{\lambda y}$

(Constraint 2) $\Leftrightarrow \forall y,\ \beta_1.\beta_2.p_3(y).e^{-\lambda y - \lambda' y} = p_1(y).e^{\lambda' y}.\beta_2.\beta_3$

By simplifying, we obtain the following system of equations:

$$\begin{cases} \beta_2.p_3(y).e^{-\lambda y}.e^{-\lambda' y} = \beta_3.p_2(y).e^{\lambda y}(1) \\ \beta_1.p_3(y).e^{-\lambda y}.e^{-\lambda' y} = \beta_3.p_1(y).e^{\lambda' y}(2) \end{cases}$$

First we substitute $e^{-\lambda y}$ *with* $e^{-\lambda' y}$*. From* (1) *we obtain:*

$(1) \Leftrightarrow \beta_2.p_3(y).e^{-\lambda y}.e^{-\lambda' y} = \beta_3.p_2(y).e^{\lambda y}$

$\Leftrightarrow \beta_2.p_3(y).\frac{1}{e^{\lambda y}}.\frac{1}{e^{\lambda' y}} = \beta_3.p_2(y).e^{\lambda y}$

$\Leftrightarrow \frac{\beta_2}{\beta_3}.\frac{p_3(y)}{p_2(y)}.\frac{1}{e^{\lambda' y}} = (e^{\lambda y})^2$

$\Leftrightarrow \sqrt{\frac{\beta_2}{\beta_3}}.\sqrt{\frac{p_3(y)}{p_2(y)}}.\sqrt{\frac{1}{e^{\lambda' y}}} = e^{\lambda y}$

Now we substitute in (2)*:*

(2) $\beta_1.p_3(y).\frac{1}{\sqrt{\frac{\beta_2}{\beta_3}}.\sqrt{\frac{p_3(y)}{p_2(y)}}.\sqrt{\frac{1}{e^{\lambda' y}}}}.\frac{1}{e^{\lambda' y}} = \beta_3.p_1(y).e^{\lambda' y}$

$\Leftrightarrow \beta_1.p_3(y).(\frac{\beta_2}{\beta_3})^{-\frac{1}{2}}.(\frac{p_3(y)}{p_2(y)})^{-\frac{1}{2}}.(\frac{1}{e^{\lambda' y}})^{-\frac{1}{2}} = \beta_3.p_1(y).(e^{\lambda' y})^2$

$\Leftrightarrow \beta_1.p_3(y).(\frac{\beta_3}{\beta_2})^{\frac{1}{2}}.(\frac{p_2(y)}{p_3(y)})^{\frac{1}{2}}.(e^{\lambda' y})^{\frac{1}{2}} = \beta_3.p_1(y).(e^{\lambda' y})^2$

$$\Leftrightarrow \beta_1.p_3(y).(\frac{\beta_3}{\beta_2})^{\frac{1}{2}}.(\frac{p_2(y)}{p_3(y)})^{\frac{1}{2}}.\frac{1}{\beta_3}.\frac{1}{p_1(y)} = \frac{(e^{\lambda' y})^2}{(e^{\lambda' y})^{\frac{1}{2}}}$$

$$\Leftrightarrow \frac{\beta_1}{\beta_3}.(\frac{\beta_3}{\beta_2})^{\frac{1}{2}}.\frac{p_3(y)}{p_1(y)}.(\frac{p_2(y)}{p_3(y)})^{\frac{1}{2}} = (e^{\lambda' y})^2.(e^{\lambda' y})^{-\frac{1}{2}}$$

$$\Leftrightarrow \frac{\beta_1}{\beta_3}.\frac{\beta_3^{\frac{1}{2}}}{\beta_2^{\frac{1}{2}}}.\frac{p_3(y)}{p_1(y)}.\frac{p_2(y)^{\frac{1}{2}}}{p_3(y)^{\frac{1}{2}}} = (e^{\lambda' y})^{\frac{3}{2}}$$

$$\Leftrightarrow \frac{\beta_1}{\beta_3^{\frac{1}{2}}.\beta_2^{\frac{1}{2}}}.\frac{p_3(y)^{\frac{1}{2}}p_2(y)^{\frac{1}{2}}}{p_1(y)} = (e^{\lambda' y})^{\frac{3}{2}}$$

$$\Leftrightarrow \frac{\beta_1^2}{\beta_3.\beta_2}.\frac{p_3(y).p_2(y)}{p_1(y)^2} = (e^{\lambda' y})^3$$

$$\Leftrightarrow (\frac{\beta_1^2}{\beta_3.\beta_2}.\frac{p_3(y).p_2(y)}{p_1(y)^2})^{\frac{1}{3}} = e^{\lambda' y}$$

$$\Leftrightarrow (e^{\lambda y})^3 = (\frac{\beta_2}{\beta_3})^{\frac{3}{2}}.(\frac{p_3(y)}{p_2(y)})^{\frac{3}{2}}.(\frac{1}{e^{\lambda' y}})^{\frac{3}{2}}$$

$$\Leftrightarrow (e^{\lambda y})^3 = (\frac{\beta_2}{\beta_3})^{\frac{3}{2}}.(\frac{p_3(y)}{p_2(y)})^{\frac{3}{2}}.\frac{1}{(e^{\lambda' y})^{\frac{3}{2}}}$$

$$\Leftrightarrow (e^{\lambda y})^3 = \frac{\beta_2^{\frac{3}{2}}}{\beta_3^{\frac{3}{2}}}.\frac{p_3(y)^{\frac{3}{2}}}{p_2(y)^{\frac{3}{2}}}.\frac{\beta_3^{\frac{1}{2}}.\beta_2^{\frac{1}{2}}}{\beta_1}.\frac{p_1(y)}{p_3(y)^{\frac{1}{2}}.p_2(y)^{\frac{1}{2}}}$$

$$\Leftrightarrow (e^{\lambda y})^3 = \frac{\beta_2^{\frac{3}{2}}.\beta_3^{\frac{1}{2}}.\beta_2^{\frac{1}{2}}}{\beta_3^{\frac{3}{2}}.\beta_1}.\frac{p_3(y)^{\frac{3}{2}}.p_1(y)}{p_3(y)^{\frac{1}{2}}.p_2(y)^{\frac{1}{2}}.p_2(y)^{\frac{3}{2}}}$$

$$\Leftrightarrow (e^{\lambda y})^3 = \frac{\beta_2^2}{\beta_1.\beta_3}.\frac{p_1(y).p_3(y)}{p_2(y)^2}$$

$$\Leftrightarrow e^{\lambda y} = (\frac{\beta_2^2}{\beta_1.\beta_3}.\frac{p_1(y).p_3(y)}{p_2(y)^2})^{\frac{1}{3}}$$

We know that :

$$\forall y,\ q(y) = \sum_{y_1 y_2} q(y_1, y_2, y) = \sum_{y_1 y_3} q(y_1, y, y_3) = \sum_{y_2 y_3} q(y, y_2, y_3)$$

Consider one of these equations with constraint q_{ijk} to be a point of derivative of the primal that we are looking for.

$\forall y,\ q(y) = \sum_{y_1 y_3} q(y_1, y, y_3)$

$= \beta_1.\beta_3.p_2(y).e^{\lambda y}.e^{\lambda-1}$

$= \beta_1.\beta_3.p_2(y).(\frac{\beta_2^2}{\beta_1.\beta_3}.\frac{p_1(y).p_3(y)}{p_2(y)^2})^{\frac{1}{3}}.(e^{-\lambda-1})$

$= \beta_1.\beta_3.p_2(y).\frac{\beta_2^{\frac{2}{3}}}{\beta_1^{\frac{1}{3}}.\beta_3^{\frac{1}{3}}}.\frac{p_1(y)^{\frac{1}{3}}.p_3(y)^{\frac{1}{3}}}{p_2(y)^{\frac{2}{3}}}.(e^{-\lambda-1})$

$\forall y,\ q(y) = \frac{\beta_1.\beta_3.\beta_2^{\frac{2}{3}}}{\beta_1^{\frac{1}{3}}.\beta_3^{\frac{1}{3}}}.\frac{p_2(y).p_1(y)^{\frac{1}{3}}.p_3(y)^{\frac{1}{3}}}{p_2(y)^{\frac{2}{3}}}.e^{-\lambda-1}$

$= \beta_1^{\frac{2}{3}}.\beta_3^{\frac{2}{3}}.\beta_2^{\frac{2}{3}}.p_2(y)^{\frac{1}{3}}.p_1(y)^{\frac{1}{3}}.p_3(y)^{\frac{1}{3}}.e^{-\lambda-1}$

$= e^{-\lambda-1}.\sqrt[3]{p_1(y).p_2(y).p_3(y)}.(\beta_1.\beta_2.\beta_3)^{\frac{2}{3}}$

$= e^{-\lambda-1}.\sqrt[3]{p_1(y).p_2(y).p_3(y)}.\beta^{\frac{2}{3}}$

So, we know that : $\beta = \frac{1}{e^{-\lambda-1}}$ *is the optimum of the dual problem.*

$\forall y,\ q(y) = e^{-\lambda-1}.\sqrt[3]{p_1(y).p_2(y).p_3(y)}.(\frac{1}{e^{-\lambda-1}})^{\frac{2}{3}}$

$= \frac{e^{-\lambda-1}}{(e^{-\lambda-1})^{\frac{2}{3}}}.\sqrt[3]{p_1(y).p_2(y).p_3(y)}$

$\forall y,\ q(y) = \sqrt[3]{e^{-\lambda-1}}.\sqrt[3]{p_1(y).p_2(y).p_3(y)}$

$\forall y,\ q(y) = agree(p_1, p_2, p_3) \propto \sqrt[3]{p_1(y).p_2(y).p_3(y)}$

$e^{-\lambda-1} = \frac{1}{\beta}$

And so, we have an equation which does not depend on $\lambda_\epsilon, \lambda'_\epsilon, \lambda$.

References

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

Banea, C., Mihalcea, R., Wiebe, J. & Hassan, S. (2008). Multilingual subjectivity analysis using machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, 127--135.

Bauer, D. (1972). Constructing confidence sets using rank statistics. *Journal of the American Statistical Association*, **67**, 687--690.

Belhumeur, P.N., Hespanha, J.P. & Kriegman, D.J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection.

Blitzer, J., Dredze, M. & Pereira, F. (2007). Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, 187--205.

Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT 1998)*, 92--100.

Boiy, E., Hens, P., Deschacht, K. & Moens, M.F. (2007). Automatic sentiment analysis of on-line text. In *Proceedings of the 11th International Conference on Electronic Publishing (ELPUB 2007)*, 349--360.

Boughanem, M., Missen, M.M.S. & Cabanac, G. (2009). Challenges for sentence level opinion detection in blogs. In *ACIS-ICIS*, 347--351.

Brefeld, U., Buscher, C. & Scheffer, T. (2005). Multi-view discriminative sequential learning. In *Proceedings of the 16th European Conference on Machine Learning (ECML 2005)*, 60--71.

Brill, E. (1994). Some advances in transformation-based parts of speech tagging. In *Proceedings of the twelfth national conference on Artificial intelligence AAAI*, 722--727.

Bruce, R.F. & Wiebe, J.M. (1999). Recognizing subjectivity: A case study in manual tagging. *Natural Language Engineering*, **5**.

Chesley, P., Vincent, B., Xu, L. & Srihari, R. (2006). Using verbs and adjectives to automatically classify blog sentiment. In *Proceedings of the AAAI Symposium on Computational Approaches to Analyzing Weblogs (AAAI/CAAW 2006)*, 27--29.

Church, K.W. & Hanks, P. (1989). Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL)*, 76--83.

Collins, M. & Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP 1999)*, 100--110.

Dave, K., Lawrence, S. & Pennock, D.M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of WWW*, 519--528.

Dias, G. (2010). *Information Digestion*. Hdr thesis, University of Orleans.

Dias, G., Mukelov, R. & Cleuziou, G. (2008). Unsupervised graph-based discovery of general-specific noun relationships from web corpora frequency counts. In *Proceedings of the 12th International Conference on Natural Language Learning (CoNLL 2008)*.

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

Esuli, A. & Sebastiani, F. (2006a). Determining term subjectivity and term orientation for opinion mining. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

Esuli, A. & Sebastiani, F. (2006b). SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of Language Resources and Evaluation (LREC)*.

Finn, A. & Kushmerick, N. (2006). Learning to classify documents according to genre. *American Society for Information Science and Technology, Special issue on Computational Analysis of Style*, **57**, 1506--1518.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA.

Ganchev, K., Graca, J., Blitzer, J. & Taskar, B. (2008). Multi-view learning over structured and non-identical outputs. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI 2008)*, 204--211.

Hatzivassiloglou, V. & McKeown, K. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics (EACL 1997)*, 174--181.

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

Hatzivassiloglou, V., Klavans, J.L., Holcombe, M.L., Barzilay, R., yen Kan, M. & McKeown, K.R. (2001). Simfinder: A flexible clustering tool for summarization. In *In Proceedings of the NAACL Workshop on Automatic Summarization*, 41--49.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of 10th the European Conference on Machine Learning (ECML 1998)*, 137--142.

Joachims, T. (1999). Making large-scale svm learning practical. In *Advances in Kernel Methods - Support Vector Learning, B. Scholkopf and C. Burges and A. Smola (ed.)*, 169--184, MIT-Press.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers.

Kailath, T. (1967). The divergence and bhattacharyya distance measures in signal selection. In *IEEE Transactions on Communications*, vol. 15, 52–60.

Kamps, J., Marx, M., Mokken, R.J. & de Rijke, M. (2004). Using WordNet to measure semantic orientation of adjectives. In *LREC*.

Kim, S.M. & Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Kruskal, J.B. & Wish, M. (1977). *Multidimensional Scaling*. Sage Publications.

Lambov, D., Dias, G. & Noncheva, V. (2009a). High level features for learning subjective language. In *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media (ICWSM 2009)*.

Lambov, D., Dias, G. & Noncheva, V. (2009b). Sentiment classification across domains. In *14th Portuguese Conference on Artificial Intelligence (EPIA 2009)*.

Lambov, D., Dias, G. & Graca, J. (2010). Multi-view learning for text subjectivity classification. In *Proceedings of the Workshop on Computational Approaches to Subjectivity and Sentiment Analysis of the 19th European Conference on Artificial Intelligence (ECAI 2010)*.

Lambov, D., Dias, G. & Pais, S. (2011). Merged agreement algorithms for domain independent sentiment analysis. In *Proceedings of 12th Conference of the Pacific Association for Computational Linguistics (PACLING 2011)*.

Levin, B. (1993). *English Verb Classes and Alternations*. University of Chicago Press.

Liu, H. (2004). Montylingua: An end-to-end natural language processor with common sense.

Mihalcea, R. & Banea, C. (2007). Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, 976--983.

Miller, G.A. (1990). Wordnet: an on-line lexical database. *International Journal of Lexicography*, **3**.

Missen, M.M. & Boughanem, M. (2009). Using wordnet's semantic relations for opinion detection in blogs. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, 729--733, Springer-Verlag, Berlin, Heidelberg.

Missen, M.M.S., Boughanem, M. & Cabanac, G. (2009). Comparing semantic associations in sentences and paragraphs for opinion detection in blogs. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, 80:483--80:488, ACM.

Mullen, T. & Malouf, R. (2006). A preliminary investigation into sentiment analysis of informal political discourse. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW*, 159--162.

Ng, V., Dasgupta, S. & Arifin, S.M.N. (2006). Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of COLING/ACL 2006 Main Conference Poster Sessions*, 611--618.

Osgood, C., Suci, G. & Tannebaum, P. (1971). *The Measurement of Meaning*. University of Illinois Press.

Pais, S. (2007). *Classification of Opinionated Texts by Analogy*. Master's thesis, University of Beira Interior.

Pang, B. & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, 271--278.

Pang, B. & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, **2**, 1--135.

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

Quirk, R., Greenbaum, S., Leech, G. & Svartvik, J. (1985). *A comprehensive grammar of the English language*. Longman.

Rocchio, J. (1971). Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, chap. 14, 313--323, Prentice-Hall.

Rodrigues, D. (2009). *Construcao Automatica de um Dicionario Emocional para o Portugues*. Master's thesis, University of Beira Interior.

Salton, G., Yang, C. & Yu, C. (1975). A theory of term importance in automatic text analysis. *American Society of Information Science*, **26**, 33--44.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, **34**, 1--47.

Shai Ben-David, K.C., John Blitzer & Pereira, F. (2006). Analysis of representations for domain adaptation. In *In Neural Information Processing Systems (NIPS)*.

Sindhwani, V. & Niyogi, P. (2005). A co-regularized approach to semi-supervised learning with multiple views. In *Proceedings of the Workshop on Learning with Multiple Views of the 22nd International Conference (ICML 2005)*, 1--6.

Srihari, R., Li, W., Cornell, T. & Niu, C. (2006). Infoxtract: A customizable intermediate level information extraction engine. *Natural Language Engineering*, 33--69.

Strapparava, C. & Mihalcea, R. (2008). Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC 2008)*, 1556--1560.

Strapparava, C. & Valitutti, A. (2004). Wordnet-affect: An affective extension of wordnet. In *Proceedings of the 4th Language Resources and Evaluation International Conference (LREC 2004)*, 1083--1086.

Torkkola, K. (2001). Linear discriminant analysis in document classification. In *In IEEE ICDM Workshop on Text Mining*.

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

Turney, P.D. & Littman, M.L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, **21**, 315--346.

Wan, X. (2009). Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL/IJCNLP 2009)*, 235--243.

Whitelaw, C., Garg, N. & Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, 625--631, ACM, New York, NY, USA.

Wiebe, J., Bruce, R. & O'Hara, T. (1999). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics (ACL 1999)*, 246--253.

Wiebe, J., Wilson, T., Bruce, R., Bell, M. & Martin, M. (2004). Learning subjective language. *Computational Linguistics*, **30**, 277--308.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, **1**, 80--83.

Wilson, T., Wiebe, J. & Hwa, R. (2006). Recognizing strong and weak opinion clauses. *Computational Intelligence*, **22**, 73--99.

Xiong, T. & Cherkassky, V. (2005). A combined svm and lda approach for classification. In *International Joint Conference on Neural Networks (IJCNN 05*, 1455 -- 1459.

Yi, J., Nasukawa, T., Bunescu, R. & Niblack, W. (2003). Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*.

Yu, H. & Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, 129--136.

Yu, N. (2009). *Opinion Detection for Web Content*. Phd thesis, Indiana University.