From Semantic to Emotional Space in Sense Sentiment Analysis

Mitra Mohtarami

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Department of Computer Science

NATIONAL UNIVERSITY OF SINGAPORE

2013

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Mitra Mohtarami

Digitally signed by Mitra Mohtarami
Date: 2013.12.04
13:43:04 +08'00'

Abstract

From Semantic to Emotional Space in Sense Sentiment Analysis

Mitra Mohtarami

This thesis is focused on inferring sense sentiment similarity and indicating its effectiveness in natural language processing tasks, namely, Indirect yes/no Question Answer Pair (IQAP) inference and Sentiment Orientation (SO) prediction. Sense sentiment similarity models the relevance of words regarding their senses and underlying sentiments.

To achieve the aims of this thesis, we first investigate the differentiation of the semantic and sentiment similarity measures. It results that although the semantic similarities are good measures for relating semantically related words, they are less effective in relating words with similar sentiment. This result leads to a need of sentiment similarity measure. Thus, we then model the words in emotional space employing the association between the semantic space and emotional space of word senses to infer their emotional vectors. These emotional vectors are used to predict the sense sentiment similarity of the words. To map the words into emotional vectors, we first employ the set of basic human emotions that are central to other emotions: *anger, disgust, sadness, fear, guilt, interest, joy, shame, surprise*. Then, we assume that the number and types of the emotions are hidden and propose hidden emotional models for predicting the emotional vectors of the words and interpreting the hidden emotions that aim to infer sense sentiment similarity.

Experimental results through IQAPs inference and SO prediction tasks show that the sense sentiment similarity is more effective than semantic similarity measures. The experiments indicate that utilizing the emotional vectors of the words is more accurate than comparing their overall sentiments in IQAPs inference. In addition, in SO prediction, we can obtain a comparable result with the state-of-the-art approach, when we employ sense sentiment similarity along with a simple algorithm to predict the sentiment orientation.

Contents

List of Figures

List of Tables

Acknowledgments

This thesis would not have been possible without the support of a number of individuals and I gratefully acknowledge all of them.

First of all, I would like to sincerely thank my supervisor, Prof. Chew Lim Tan, for his guidance and help through my Ph.D. study. He has taught me how a real researcher can grow up in the world of science, and how science and research have positive impact on the researcher's mind, ethics and life. Prof. Tan has taught me that I should have hard enough effort to do research and enough patience to see a good result, and I should also follow this scenario in my life to be successful. I am most grateful to Prof. Tan for his support and it is a great chance in my life to be his student.

I am most grateful to Prof. Hwee Tou Ng for giving me insights over the NLP concepts. Taking the NLP and advanced NLP classes with Prof. Ng was one of my most exciting experiences. I am indebted to my dissertation committee, Prof. Hwee Tou Ng, Prof. Min-Yen Kan and Prof. Bing Liu for their insights and helpful suggestions that improve the quality of this thesis.

I would also like to thank my lab mates in CHIME lab, Tianxia Gong, Shimiao Li, Sun Jun, Richard Liu, Bolan Su, Chen Qi for their warm friendship from the first day of my Ph.D. study.

I owe my deepest gratitude to my parent, Maryam Rahimi and Fazlollah Mohtarami, for their endless love, support and encouragement. Thinking them makes me stronger and more diligent in all stages of my life.

I would like to thank my spouse, Hadi, for being beside me in study, research, life and love. I would like to thank him for all enjoyable time for discussing over our research ideas on the way back to home from university.

To my spouse, Hadi

To my parents, Maryam Rahimi and fazlollah Mohtarami

Chapter 1

Introduction

Natural language processing (NLP) is a form of human-to-computer interaction. Many challenges in NLP attempt to enable computers to derive meaning and sentiment from human/natural language as written or spoken inputs. To achieve this aim, various research areas have appeared that can be categorized into two groups. The first research group deals with extracting and interpreting the meaning of the natural language, for instance in the following research areas:

Speech processing: It aims at enabling the computer to model and manipulate the speech signal to be able to transmit (code) speech efficiently, produce (synthesis) natural sounding voice, and recognise (decode) spoken words (Jurafsky and Martin, 2009).

Information extraction: It aims at enabling the computer to extract the semantic information from text. This covers the NLP tasks such as named entity recognition, co-reference resolution, relationship extraction, etc (Manning and Schütze, 1999).

Information retrieval: It aims at enabling the computer to find materials (usually documents) of an unstructured nature (usually raw text) that satisfies an information need from large collections of documents (Manning, Raghavan, and Schütze, 2008).

Question answering: It aims at enabling the computer to answer natural language questions. Given a collection of documents, a QA system attempts to retrieve correct answers to questions posed in natural language and in some cases reason about the resultant answer (Ferrucci et al., 2010).

The second research group deals with extracting and interpreting the sentiment of the natural language that are subtopics of ***Sentiment analysis***. Sentiment analysis is the research on computational study of opinions, sentiments, subjectivity, attitudes, appraisal, affects, views, and emotions etc., expressed in text or speech. It is one of the most active research areas in natural language processing and is also widely studied in data mining, Web mining, and text mining (Liu, 2007).

Sentiment analysis is technically challenging and practically very useful. For example, companies always want to find public or consumer opinions about their products and services, potential customers also want to know the opinions of existing users before they use a service or purchase a product, recommendation systems need to automatically recommend new products or services to their users, Ads placement software needs to find pages that contain positive sentiments about a service or product, and etc.

"Sentiment Analysis" and "Opinion Mining" are often used interchangeably as their basic definitions about sentiment or opinion are the same. An opinion is simply a positive or negative sentiment, view, attitude, emotion, or appraisal about an entity or an aspect of the entity (Hu and Liu, 2004; Liu, 2010) from an opinion holder (Bethard et al., 2004; Kim and Hovy, 2004; Wiebe, Wilson, and Cardie, 2005). The following is a list of the most commonly research tasks in sentiment analysis or opinion mining (Pang and Lee, 2008; Liu, 2010).

Document sentiment classification: It is the research on classifying a whole opinion document (e.g., a review) based on the overall sentiment of the opinion holder (Pang, Lee, and Vaithyanathan, 2002; Turney, 2002) as positive, negative, and possibly neutral.

Sentence subjectivity and sentiment classification: Document-level sentiment classification is too coarse for most applications. Thus, these research works moved to the finer-grained levels like sentence. Most of the early work on sentence level analysis focuses on identifying subjectivity in sentences which is about classifying a sentence into objective or subjective classes (Wiebe, Bruce, and O'Hara, 1999).

Aspect-based sentiment analysis: Given a set of customer reviews of a particular product, the aspect-based sentiment analysis involves the following subtasks: (1) identifying features of the product that customers have expressed their opinions on (called product features); (2) for each feature, identifying positive or negative review sentences; and (3) producing a summary using the discovered information for the whole product (Hu and Liu, 2004).

Aspect-based opinion summarization: Aspect-based opinion summarization corresponds to the above third sub-task of aspect-based sentiment analysis. This is a multi-document summarization problem where aspects are the basis for producing a summary.

Opinion lexicon generation: It is the research on generating lists of words and expressions used to express people's subjective feelings and sentiments or opinions. The purpose is to generate not only individual words, but also phrases and idioms (such as "cost you an arm and a leg") that represent opinions.

Mining comparative opinions: Given a subjective document, this task focuses on extracting comparative opinions for the entity sets being compared based on their shared aspects, for example for products.

Opinion spam detection: Opinion spamming refers to fake or untruthful opinions. In this sub-task the users play important role in identifying spams.

Utility or helpfulness of reviews: This task aims to determine the usefulness, helpfulness, or utility of each review. It is desirable to rank reviews based on utilities or qualities when showing them to users, with the highest quality review first. This component can be utilized as a supporting mean for the summarization task.

Regarding the goal of NLP tasks that is generally inferring the meaning and sentiment from the natural language, this thesis revolves around sentiment analysis of natural language text or the so-called *User Generated Content* (UGC). There exists a wide range of sources of user generated contents, e.g. discussion boards, blogs, wikis, social networking portals, trip planners and customer review portals. Each of these sources contains a huge volume of subjective text. In fact, users have difficulty in identifying relevant sites and accurately summarizing their information and opinions on different entities. However, this difficulty can be handled by the sentiment analysis tasks.

In the domain of sentiment analysis, although there are various studies that have been done by existing works, there are still research issues that are unknown or receiving less-attention. For instance, sense sentiment similarity still needs intensive research as it is one of the fundamental concepts in sentiment analysis and is deemed very effective in NLP tasks. ***Sense sentiment similarity*** aims to infer the similarity between two entities based on the likeness of their sentiment. We will next provide a brief overview of *sense sentiment similarity*, and show its significance and applications in NLP, namely, in opinion question-answering and sentiment orientation prediction. A more detailed discussion of existing research will be presented in Chapter 2.

1.1 The Problem of Sense Sentiment Similarity

Prior research has proposed novel approaches and used existing resources to address the sentiment analysis tasks. For example, the majority of previous sentiment analysis research has employed the existing semantic similarity measures to estimate the sentiment similarity between entities like words, phrase, sentences, and etc (Kim and Hovy, 2007; Turney and Littman, 2003). The hypothesis is that two entities that are semantically correlated (e.g., synonyms at the word level) can have similar sentiment orientation. Otherwise, they may have opposite sentiment orientation (e.g., antonyms).

Semantic similarity computes the similarity between two entities based on the likeness of their meaning/semantic content. Latent Semantic Analysis (LSA), Point-wise Mutual Information (PMI), and WordNet-based similarity method are some examples of the semantic similarity measures (Pedersen, Patwardhan, and Michelizzi, 2004). These measures are good for relating semantically related words like "*car*" and "*automobile*", but are less effective in relating opinion words with similar sentiment. To date, sentiment similarity has not received enough attention. This limitation leads to a need to investigate sentiment similarity. Thus, the main aim of this thesis is to investigate the sentiment similarity between two entities with respect to their senses (e.g. word sense) and utilize it to improve different NLP tasks. In view of the literature review in Chapter 2, the current research gaps in existing works and the specific objectives of this thesis are summarized below:

- **Sentiment similarity vs. Semantic similarity**
  - [Gap] Semantic similarity measures are suitable to capture the similarity between entities with respect to their meanings/ semantics. However, they are less effective in capturing the sentiment similarity.
  - [Objective] We attempt to find an approach to accurately infer sentiment similarity, and attempt to investigate the difference between sentiment and semantic similarity measures that aim to indicate the significance of the sentiment similarity between entities in opinion- or sentiment-related NLP tasks.

- **Significance of the knowledge of word senses in similarity measures**
  - [Gap] The majority of the current research works on estimating semantic similarity only consider words or words along with their Part-of-speech (POS) tags. There are few studies that have considered the senses of the words to estimate the similarity.
  - [Objective] This thesis shows that the knowledge of the word senses can be useful in inferring sentiment similarity of the entities. The reason is that a word can have different meaning and sentiment in its various senses.
- **Indirect yes/no question answer pairs inference**
  - [Gap] This is a fundamental task in opinion question answering area which aims to infer the "*Yes*" or "*No*" answer from an indirect question-answer pair[1]. The state-of-the-art research work has employed total sentiment of the opinion words in the question and its corresponding answer to interpret the indirect answer. However, we will show that using only total sentiment of the words is less effective in predicting the certainty of the answer relative to its question.
  - [Objective] This thesis investigates this task and attempt to address it using sentiment similarity in which the semantic and sentiment spaces are combined.
- **Sentiment orientation prediction**
  - [Gap] This is a fundamental task in sentiment analysis area where the target is to determine the sentiment orientation (positive or negative) of a given entity. Existing research works ex-

[1] An indirect question-answer pair is a yes-no question that the corresponding answer is not an explicit *yes* or *no* while such answer should be inferred using context information.

plored this task by proposing different algorithms that employed semantic similarity measures.

- [Objective] We address this task by utilizing the proposed sentiment similarity measure in contrast to semantic similarity measures proposed in existing research.

The result of this investigation has significant impact on sentiment analysis area and could affect other natural language processing tasks, such as question-answering, etc.

The concept of sense sentiment similarity is a new finding and aims to infer the sentiment similarity using user generated contents like reviews. Thus, there may be a few general issues involved. For example, the user generated contents may contain grammatical and misspelling errors. In addition, the users may employ slangs that make their writing very complicated. However, these general issues are not central to this study and hence are beyond the scope of this proposed thesis.

1.2 Organization of the Thesis

In order to achieve the objectives described above, this thesis presents two novel methods to compute the Sense Sentiment Similarity (SSS) between words. In addition, this thesis indicates the significance of SSS in various NLP tasks and applies the proposed methods to address the fundamental problems in question-answering and sentiment analysis areas. The aforementioned problems in each area are shown in Figure 1.1.

As Figure 1.1 shows this thesis first attempts to address the indirect yes/no Question Answer Pairs (IQAPs) which is a problem in QA domain using some popular semantic similarity measures. In addition, this thesis investigates the effectiveness of word senses and the behaviour of ambiguous sentiment adjectives to solve the IQAPs problem. These topics are described in Chapter 3. Then, in Chapter 4, an effective method based on

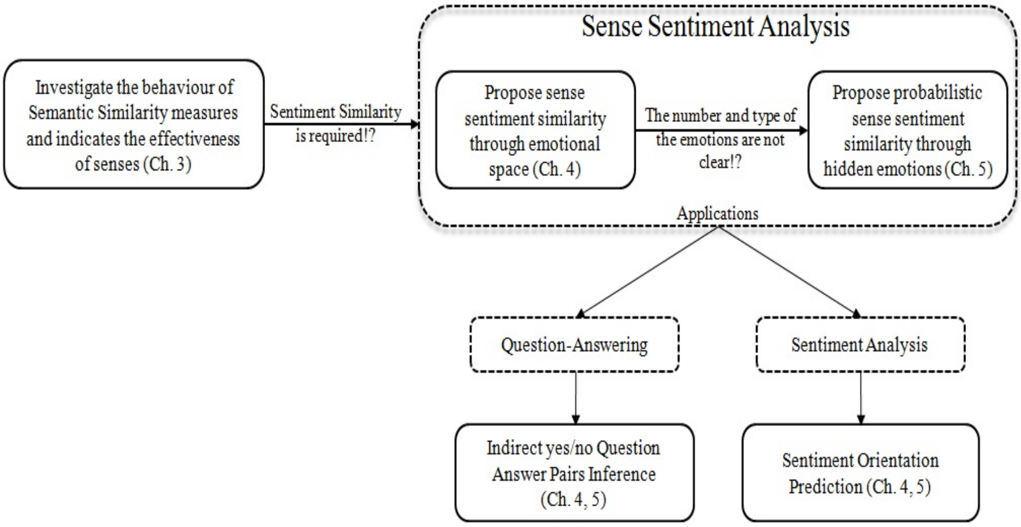

Sense Sentiment Analysis
Investigate the behaviour of Semantic Similarity measures and indicates the effectiveness of senses (Ch. 3)
Sentiment Similarity is required!?
Propose sense sentiment similarity through emotional space (Ch. 4)
The number and type of the emotions are not clear!?
Propose probabilistic sense sentiment similarity through hidden emotions (Ch. 5)
Applications
Question-Answering
Sentiment Analysis
Indirect yes/no Question Answer Pairs Inference (Ch. 4, 5)
Sentiment Orientation Prediction (Ch. 4, 5)

Figure 1.1: A quick glance at the thesis

the emotional space of the words is proposed to infer the sentiment similarity between the word pairs regarding their senses. The proposed method applies to address the IQAP problem, and predict the sentiment orientation of the words which is a fundamental task in sentiment analysis area. In Chapter 5, this thesis presents another method based on the probabilistic and hidden emotions. The proposed probabilistic method is also applied to the same NLP tasks. Finally, in Chapter 6, the contributions of this thesis are summarized and some future directions are presented.

Chapter 2

Literature Review

Current research in the area of sentiment similarity and its applications can be divided into several categories. Here we discuss these research works in the following subsections: Semantic Similarity, IQAP Inference, Sentiment Orientation Prediction, and Emotion Analysis.

2.1 Semantic Similarity

Semantic similarity aims to compute the conceptual similarity between terms. The current approaches for determining semantic similarity between terms can be divided into the following categories based on the knowledge resources employed in the approaches.

2.1.1 Dictionary-Based Approaches

To measure the semantic similarity, most of the earlier research approaches employed a dictionary or a lexical resource to construct a network or directed graph and then explored this graph. WordNet is employed by most of the existing work as a dictionary, since it is a structured dictionary and presents a hierarchical categorization of natural language terms. In the WordNet hierarchies, the synsets (i.e., sets of synonyms) are related to other synsets higher or lower in the hierarchy by different types of relationships, namely, hyponym/hypernym (*Is-A* relationships). The dictionary-based approaches can be categorized into two main categories based on how they extract knowledge form the dictionary. The categories are "Glossary-Based" and "Path-Based".

- Glossary-based approaches use only information in the dictionary definitions. For example, The *Lesk* similarity (Lesk, 1986) of two concepts is defined as a function of the overlap between the corresponding definitions, as provided by a dictionary.
- Path-based approaches have taken advantage of the hierarchical information in WordNet and proposed similarity measures as following examples:
  - The Leacock and Chodorow's similarity (Leacock and Chodorow, 1998) is determined as: $Sim_{lch} = -\log \frac{length}{2 \times D}$, where *length* is the length of the shortest path between two concepts using node-counting, and D is the maximum depth of the taxonomy.
  - The similarity metric proposed in (Wu and Palmer, 1994) measures the depth of the two concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS), and combines these figures into a similarity score defined as follows:

$$Sim_{wup} = \frac{2 \times depth(LCS)}{depth(concept_1) + depth(concept_2)} \tag{2.1}$$

2.1.2 Hybrid Approach

To predict semantic similarity, the hybrid models utilize the knowledge derived from corpora or dictionaries, rather than just using edge counting in a dictionary. The fundamental knowledge-based semantic similarity measures are as follows:

- The measure introduced by Resnik (1995) returns the information content (IC) of the LCS of two concepts: $Sim_{res} = IC(LCS)$, where

IC is defined as: $IC(c) = -\log P(c)$, where $P(c)$ is the probability of encountering an instance of concept c in a large corpus.

- Another similarity measure is introduced by (Lin, 1998), which builds on Resnik's measure of similarity, and adds a normalization factor consisting of the information content of the two input concepts:

$$Sim_{lin} = \frac{2 \times IC(LCS)}{IC(concept_1) + IC(concept_2)} \tag{2.2}$$

- (Jiang and Conrath,) proposed the following formulation to compute the similarity score which basically corresponds to the above similarity measures:

$$Sim_{jnc} = \frac{1}{IC(concept_1) + IC(concept_2) - 2 \times IC(LCS)} \tag{2.3}$$

2.1.3 Corpus-Based Approaches

This type of semantic similarity measures employs the information derived from large corpora to compute similarity. Mutual Information (MI) measures the mutual dependence of two random variables X and Y using the following equation.

$$MI(X,Y) = \sum_{y \epsilon Y} \sum_{x \epsilon X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2.4}$$

Its value is always positive and a higher value means that two random variables are more dependent on each other. The MI of the random variables X and Y is the expected value of the Pointwise Mutual Information (PMI) over all possible instances. PMI measures the mutual dependence between two instances of random variables. If X and Y are random variables, the PMI between two possible instances X = x and Y = y is computed based on the following equation:

$$PMI(x,y) = \log \frac{Pr(X = x, Y = y)}{Pr(X = x)Pr(Y = y)} \tag{2.5}$$

This quantity is zero if x and y are independent, positive if they are positively correlated, and negative if they are negatively correlated.

Mutual information suffers from two theoretical problems: It assumes independent word variables, and longer documents are given higher weights in the estimation of the feature scores, which is in contrast to common evaluation measures that do not distinguish between long and short documents. Thus, some variant of mutual information have been proposed, like, weighted-PMI (Schneider, 2005) and normalized-PMI (Bouma, 2009; Hoang, Kim, and Kan, 2009).

Latent Semantic Analysis (LSA) was proposed in (Landauer and Dumais, 1996; Landauer, Foltz, and Laham, 1998) to extract semantic relations of words. LSA involves the following steps: First, a word by document matrix is created in which each cell contains the frequency of words in documents. Second, the raw matrix is modified using weighting models. The most popular weighting is TF-IDF (Term Frequency-Inverted Document Frequency). Third, Singular Value Decomposition (SVD) is performed on the matrix. SVD finds a reduced dimensional representation of the matrix that emphasizes the strongest relationships and throws away the noise. In other words, it makes the best possible reconstruction of the matrix with the least possible information. To do this, it throws out noise, which does not help, and emphasizes strong patterns and trends, which do help.

There are a few limitations that must be considered when deciding whether to use LSA. Some of these are:

- LSA assumes a Gaussian distribution and Frobenius norm which may not fit all problems. For example, words in documents seem to follow a Poisson distribution rather than a Gaussian distribution.

- LSA cannot handle polysemy (words with multiple meanings) effectively. It assumes that the same word means the same concept which causes problems for words, like *bank* that have multiple meanings depending on which contexts they appear in.

- LSA depends heavily on SVD which is computationally intensive and hard to update as new documents appear.

To address the LSA issues, a probabilistic version of LSA (Hofmann, 2001; Hofmann, 1999a) has been presented that is called Probabilistic Latent Semantic Analysis (PLSA). PLSA aims to extract topics from large collections of text such that topics are interpretable unlike the arbitrary dimensions of LSA. PLSA is the method in which:

- documents are represented as numeric vectors in the space of words,
- the order of words is lost but the co-occurrences of words may still provide useful insights about the topical content of a collection of documents,
- each document is a probability distribution over topics , and
- each topic is a probability distribution over words

There are a few limitations that should be considered when deciding whether to use PLSA. Some of these are:

- In PLSA, the observed variable *document* is an index into some training set. Thus, there is no natural way for the model to handle previously unseen documents.
- The number of parameters for PLSA grows linearly with the number of documents in the training set. The linear growth in parameters suggests that the model is prone to overfitting and empirically, overfitting is indeed a serious problem.

Various versions of PLSA have been proposed by existing research. For example, (Chien and Wu, 2008) extended MLE-style estimation of PLSA to MAP-style estimations; a hierarchical extension was proposed in (Hofmann, 1999b); (Ding, Li, and Peng, 2008) showed the equivalent between PLSA and another popular method, non-negative matrix factorization; and a high order of proof was shown in (Peng, 2009).

(Blei, Ng, and Jordan, 2003) has proposed Latent Dirichlet Allocation (LDA) that is a generative probabilistic model of a corpus. LDA overcomes both of the PLSA problems by treating the topic mixture weights as a k-parameter hidden random variable. The parameters in a k-topic LDA model do not grow with the size of the training corpus.

The PLSA model assumes that each word of a training document comes from a randomly chosen topic. The topics are themselves drawn from a document-specific distribution over topics. However, LDA assumes that each word of both the observed and unseen documents is generated by a randomly chosen topic which is drawn from a distribution with a randomly chosen parameter.

In the LDA model, the basic idea is that the documents are represented as random mixtures over latent topics, where a topic is characterized by a distribution over words. LDA is based on the exchangeability assumption and assumes that words are generated by topics and that those topics are *infinitely exchangeable* within a document. Infinitely exchangeable is defined based on De Finetti's Theorem[1] that is described as follows:

- A finite set of random variables $x_1, ..., x_N$ is said to be *exchangeable* if the joint distribution is invariant to permutation. If π is a permutation of the integers from 1 to N:

$$p(x_1, ..., x_N) = p(x_{\pi(1)}, ..., x_{\pi(N)}) \tag{2.6}$$

- An infinite sequence of random variables is *infinitely exchangeable* if every finite subsequence is exchangeable.

[1] De Finetti's Theorem: http://en.wikipedia.org/wiki/De_Finetti's_theorem

2.2 Indirect yes/no Question Answer Pairs Inference

This task aims to infer yes/no answers from indirect yes/no question-answer pairs (IQAPs). As mentioned before, an indirect question-answer pair is a polar (yes-no) question for which the corresponding answer does not contain an explicit yes or no answer and such answer should be inferred using context information.

In (Green and Carberry, 1999), the authors presented a computational model for interpreting and generating indirect answers to polar questions using a discourse-plan-based approach and a hybrid reasoning model. (de Marneffe, Manning, and Potts, 2010) worked on indirect yes/no question-answer pairs involving an adjective in question and an adjective in the answer.

(de Marneffe, Manning, and Potts, 2010) attempted to infer the yes/no answers using sentiment orientation (SO) of the adjectives appear in question and its corresponding answer. To compute the SO of the adjectives, they used an external source in which each of the reviews has an associated star rating: one star (most negative) to ten stars (most positive). They rescaled the rating categories by subtracting 5.5 from each, to center them at 0. This yields the scale R = (-4.5, -3.5, -2.5, -1.5, -0.5, 0.5, 1.5, 2.5, 3.5, 4.5) and achieved the SO of adjective in three following computational steps:

1. The probability of a word w given a rating category r is simply computed by: $Pr(w|r) = \mathit{count}(w, r)/\mathit{count}(r)$ where $\mathit{count}(w, r)$ is the number of tokens of word w in the reviews of rating category r, and let $\mathit{count}(r)$ be the total count for all words in rating category r.

2. For each rating, $Pr(r|w) = Pr(w|r)/\sum_{r' \epsilon R} Pr(w|r')$ and finally,

3. The expected rating value for a word w is $ER(w) = \sum_{r \epsilon R} rPr(r|w)$

where ER indicates expected rating or SO of the word.

(de Marneffe, Manning, and Potts, 2010) interpreted the answer based on the SOs extracted from the question-answer pair. For instance, if the SOs of the adjectives in an IQAP has different signs, then the answer conveys *no*. In case of the same sign, if the SO of the adjective in answer is greater than or equals to the SO of the adjective in question, then the answer conveys *yes*, and otherwise *no*.

They also used the method proposed in (Blair-Goldensohn et al., 2008) to compute the SO scores using WordNet instead of the external source. They showed that using WordNet produces 56% performance for inferring *yes* or *no* answers to IQAPs.

The limitation of their approach is that they assign a globally fixed SO score to each adjective. For example, the adjectives "*best*" and "*great*" are assigned the fixed SO scores of 1.08 and 1.1 respectively. This leads to ignore the context in which the adjectives appear (i.e. the IQAP). However, we will propose an approach in which the degree of certainty for the same answer can change in different IQAPs with respect to their context information. This dynamic degree of certainty not only depends on the adjective in the answer itself but also on the adjective in question that appears in the IQAP. So, we show that our method utilizes the context information better than the method proposed in (de Marneffe, Manning, and Potts, 2010).

2.3 Sentiment Orientation Prediction

The aim of polarity orientation is to label a subjective entity (word, sentence, document) as positive or negative (Pang and Lee, 2008; Liu, 2010). It is usually formulated as a binary classification task.

2.3.1 Review and Sentence Level

One of the major research topics in sentiment analysis is to automatically determine the polarity orientation of a given review as positive or negative. The review could be a movie review (Pang, Lee, and Vaithyanathan, 2002), product review, book review, or political review and the task is a binary classification task with positive and negative classes (Pang and Lee, 2008; Pang, Lee, and Vaithyanathan, 2002; Kim and Hovy, 2007; Read, 2005; Bansal, Cardie, and Lee, 2008).

An important assumption about review classification is that it assumes the review expresses opinions on a single topic and the opinions are from a single opinion holder. This assumption holds for most of the reviews because, usually, each review focuses on a single product and is written by a single user. However, it may not hold for forums and blog posts because in such environments the users may express opinions on multiple topics (e.g. products, books, etc).

There exist different approaches to review classification: classification based on text classification methods (usually supervised methods), and classification based on polarity score (unsupervised methods) (Pang and Lee, 2008; Liu, 2010):

(1) Supervised classification using text classification methods: This approach employs any existing supervised learning method to classify reviews into positive and negative classes. The common classification techniques that have been used for this task are Naive Bayesian classification, and Support Vector Machines (Pang, Lee, and Vaithyanathan, 2002). Pang et al. (2002) showed that a support vector machine (SVM) classifier with term unigram as its features and binary weights (absence (0) and presence (1)) is a strong baseline for sentiment classification on the movie review dataset. They compared Naive Bayes, maximum entropy, and support vector machines classifiers and showed that SVM outperforms the other classification methods. Different features have been used for this task (Pang, Lee, and Vaithyanathan, 2002; Dave, Lawrence, and Pennock, 2003; Abbasi, Chen, and Salem, 2008):

- **Terms and their frequency**: These features are individual words or word n-grams and their frequency counts (or other measures like TF-IDF). In some cases, word positions may also be considered. These features have been shown quite effective in sentiment classification.

- **Part of speech**: Many researches showed that adjectives are important indicators of opinions. Thus, adjectives have been treated as special features.

- **Sentiment words and phrases**: Opinion words are words that are commonly used to express positive or negative sentiments (e.g., *beautiful* and *amazing* are positive words, while *bad* and *terrible* are negative words). Although many sentiment words are adjectives and adverbs, nouns (e.g., *rubbish* and *crap*) and verbs (e.g., *hate* and *like*) can also indicate sentiments. There are also sentiment phrases and idioms that should be considered for the review classification task. Classification based on sentiment phrases uses the positive and negative phrases in reviews for classification.

- **Negations**: Negation words are important because their appearances often change the opinion orientation. For example, the sentence "*I don't like this camera*" is negative. However, not all occurrences of negation words change the opinion orientation. For example, "*not*" in "*not only ... but also ...*" does not change the orientation direction.

- **Syntactic dependency**: Words dependency based features generated from parsing or dependency trees are shown as important features for this task.

Abbasi et al. (2008) used genetic algorithms to do the review classification and proposed an algorithm called *Entropy Weighted Genetic Algo-*

rithm (EWGA). This algorithm makes use of information gain as a measure for feature selection. The EWGA algorithm achieved an accuracy of 91% on the movie review dataset (Pang, Lee, and Vaithyanathan, 2002). (Dasgupta and Ng, 2009) proposed a semi-supervised learning method for the classification task and achieved an accuracy of 76%.

Review classification (and in a more general term, sentiment analysis) is highly domain dependent and the accuracy of the algorithms differ across different domains (Pang and Lee, 2008; Liu, 2010). For example, (Turney, 2002) showed that the classification accuracy for reviews from *automobile* and *bank* domains (84% and 80% respectively) is higher than the classification accuracy for reviews from *movie* and *travel* domains (65.83% and 70.53% respectively). Transfer learning or domain adaptation has been shown effective for review classification. A classifier trained using reviews in one domain often performs poorly when it is applied or tested on reviews from another domain. The reason is that words may have different usage in different domains for expressing opinions. For example, the same word in one domain may mean positive, but in another domain may mean negative. Therefore, domain adaptation is needed.

(2) Unsupervised classification using score function: These approaches utilize a sentiment lexicon to extract a set of sentiment-bearing words and phrases from reviews. They then assign a sentiment score to each extracted word or phrase and generate an overall score for each review by summing up the sentiment scores of its word or phrase. The sign of the total score determines the class of the review (Dave, Lawrence, and Pennock, 2003).

Different sentences in a review may share different information about the polarity orientation of the review. A review could be a mixture of positive, negative, and neutral sentences, but usually it has a unique overall sentiment: positive or negative. Therefore, it is not necessary to use all the sentences to predict the overall sentiment of a review. (Becker and

Aharonson, 2010) showed that final sentences of reviews (instead of the whole review) can be used for review classification with no significant difference when we use the whole content of the review.

2.3.2 Aspect Level

In many sentiment analysis applications the objects[2] (Liu, 2007; Liu, 2010) in a review are considered as important evidences (Pang and Lee, 2008; Liu, 2010). For example one may just look for the opinions on a specific product, e.g., "*canon powershot sx210*". Each object can be assigned a set of *aspects* (e.g. considering *camera* as an object, its *picture quality* is an aspect) (Liu, 2007; Liu, 2010). So, one can study the subjective texts at the aspect level to generate detailed information about sentiments on different aspects of the objects.

In a typical review, the author writes both positive and negative aspects of the object, although the general sentiment on the entity may be positive or negative. However, review classification does not provide such information. To obtain these details, we need to go to the aspect level.

Aspect-level sentiment classification can be done in three steps as follows:

1. **Mark opinion words and phrases**: Given a sentence that contains one or more aspects, this step marks all sentiment words and phrases in the sentence. Each positive word is assigned the sentiment score of +1, and each negative word is assigned the sentiment score of -1.

2. **Handle sentiment shifters**: Sentiment shifters are the words and phrases that can shift or change sentiment orientations. Negation words like *not*, *never*, *none*, *nobody*, *nowhere*, *neither* and *cannot* are the common type of sentiment shifters. Furthermore, in English, *but* means contrary. We can handle *but* as follows: the opinion orientation

[2] An object could be a product, person, event, organization, or even a topic

before *but* and after *but* are opposite to each other if the opinion on one side cannot be determined. We should also note that, not every appearance of an opinion shifter changes the opinion orientation, e.g., "*not only ... but also ....*"

3. **Aggregating opinions**: This step applies a sentiment aggregation function to the resulting sentiment scores to determine the final orientation of the sentiment on each aspect in the sentence.

One main shortcoming of the above approach is that sentiment words or phrases obtained from a sentiment dictionary do not cover all types of expressions that convey sentiments. There are in fact many other possible sentiment bearing expressions.

2.3.3 Lexicon Level

One of the fundamental tasks in sentiment analysis is determining the polarity (sentiment orientation) of words. For example, the words "*excellent*" and "*amazing*" are positive-bearing words, while "*poor*" and "*terrible*" are negative-bearing words. Opinion words are stored in opinion lexicons and are used in the majority of sentiment analysis tasks, such as opinion retrieval (Ounis et al., 2006), opinion question answering (Dang and Owczarzak, 2008), opinion mining (Yi et al., 2003; Ding, Li, and Peng, 2008), and especially in the opinion classification task (Pang and Lee, 2008; Liu, 2010). Although most of the existing research worked on assigning a static (prior) polarity to each lexicon out of context, the polarity of some sentiment lexicons varies strongly with context. For example, the word *low* has a positive orientation in *low cost* but a negative orientation in *low salary.* We call these words like *low* ambiguous sentiment words. Based on consideration of this matter, current research can be divided into two categories; context-free sentiment prediction and context-dependent sentiment prediction which are explained in Sections 2.3.3.1 and 2.3.3.2, respectively.

2.3.3.1 Context-Free Sentiment Prediction

(Turney and Littman, 2003) proposed a method for automatically inferring the sentiment orientation of a word from its statistical association with a set of positive and negative seed words. To calculate the statistical association of a word with positive (negative) seed words, they used the number of hits returned by a search engine, with a query consisting of the word and one of the seed words (e.g., "*word* NEAR good", "*word* NEAR bad"). The proximity, NEAR, is to look for instances where the given word is physically close to the seed word in the returned document. The following seven positive and seven negative seed words are used as paradigms of positive and negative sentiment orientation:

- Good, nice, excellent, positive, fortunate, correct, and superior.
- Bad, nasty, poor, negative, unfortunate, wrong, and inferior.

Finally, they regarded the difference of two association strengths as a measure of sentiment orientation. The limitation of their work is that the seed words are carefully chosen instead of randomly selected and their approach may not work efficiently with new seed words, such as the following seed words:

- Right, worth, commission, classic, devote, super, confidence.
- Lost, burden, pick, raise, guilt, capital, blur.

With new seed words, the accuracy is reduced due to their sensitivity to context, in contrast to the original seed words. For example, the following ambiguous sentiment words *pick*, *raise*, and *capital* may seem surprising. These words are negative in some contexts, such as "pick on your brother", "raise a protest", and "capital offense", and are positive in others. Their approach is corpus-based approach which considers the co-occurrence of a word with one of the seed words.

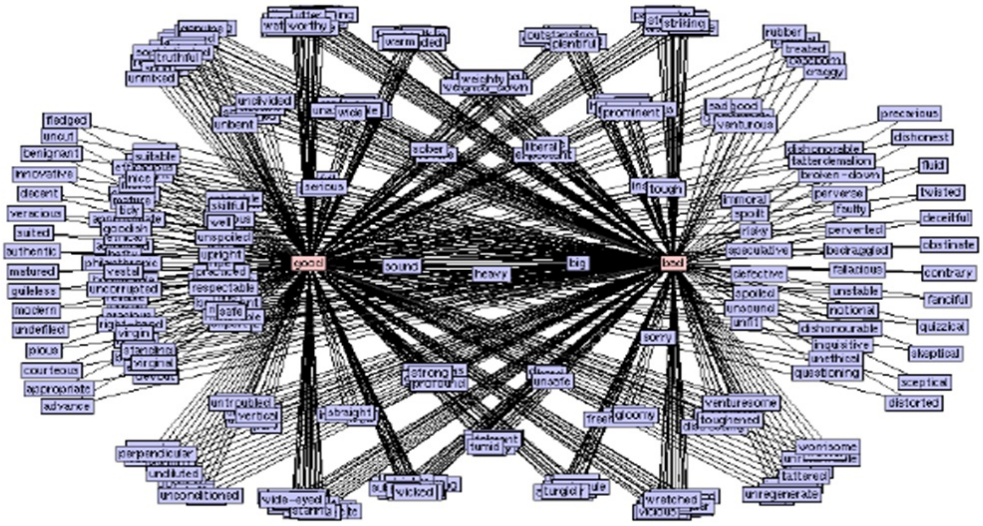


Figure 2.1: adapted from Kamps et al. (2004), the distance of a word with a set of bipolar adjectives (e.g., *good* and *bad*) is used to compute its SO

Another type of approach is proposed that is called dictionary-based approach which utilizes machine learning methods to construct opinion lexicons. The majority of dictionary-based methods use a small set of manually selected seed opinion words and dictionaries like WordNet[3]. (Kamps et al., 2004) presented a simple dictionary-based method for word sentiment detection. They constructed a lexical graph in which the nodes are adjectives and each edge connects two words that are synonyms based on Wordnet (Christiane, 1998).

They defined three kinds of factor based on the three sets of bipolar adjectives they employed to compute the sentiment orientation of an adjective; good/bad (as can be seen in the Figure 2.1), strong/weak, and active/passive. Then, they computed the three kinds of factors (Osgood, Suci, and Tannenbaum, 1957) as follows:

$$\begin{cases} Evaluative factor = \frac{d(w,bad)-d(w,good)}{d(good,bad)} \\ Potency factor = \frac{d(w,strong)-d(w,weak)}{d(strong,weak)} \\ Activity factor = \frac{d(w,active)-d(w,passive)}{d(active,passive)} \end{cases} \tag{2.7}$$

[3] WordNet: http://wordnet.princeton.edu/

where $d(w_i, w_j)$ between two words w_i and w_j is the length of a shortest path between w_i and w_j. Each equation is normalized by the distance between the two reference words (e.g., good/ bad). The reason is that "*good*" and "*bad*" are closely related in WordNet. There exists a 5-long sequence (*good*, *sound*, *heavy*, *big*, *bad*), shown in the Figure 2.1. Thus, we have $d(good, bad) = 4!$. Even though the adjectives "*good*" and "*bad*" have opposite meanings, they are still closely related by the synonymy relation. They used the adjectives of the dataset General Inquirer (Stone, 1997) and got the highest accuracy, 71.36%, with potency factor when scoring 0 as neutral. However, when treating [-0.25, 0.25] as neutral, the score for the evaluative factor is 76.72%, for the potency factor is 76.61%, and for the activity factor is 78.73%. One of their limitations is that the set of seed adjectives (e.g., good/ bad) employed with their approaches and the best set of seed adjectives which leads to highest accuracy are not clear. In addition, they considered only Sentiment Orientation (SO) of the adjectives and synonym relation. (Takamura, Inui, and Okumura, 2005) constructed a lexical graph by linking synonyms, antonyms, hypernyms. They also link two words if one word appears in the glossary of the other word. In addition, they used conjunctive expressions in corpus and connect two adjectives if the adjectives appear in a conjunctive form in the corpus. They regarded each word as an electron. Each electron has a spin and each spin has a direction taking one of two values: up or down. Two neighbouring spins tend to have the same orientation from an energetic point of view. Their hypothesis is that as neighbouring electrons tend to have the same spin direction while neighbouring words tend to have similar polarity. They posed the problem as an optimization problem and used the mean field method to find the best solution. They achieved 81.9% with 14 seed words (Turney and Littman, 2003) and the dataset General Inquirer lexicon (Stone et al., 1966). They mentioned several following limitations which their approaches cannot deal with:

1. Ambiguity of word senses. For example, one of the glossary entries of *costly* is "entailing great loss or sacrifice". The word *great* here means *large*, although it usually means *outstanding* and is positively oriented.

2. Lack of structural information. For example, *arrogance* means "overbearing pride evidenced by a superior manner toward the weak". Although *arrogance* is mistakenly predicted as positive due to the word *superior*, what superior here is *manner*.

3. The last one is idiomatic expressions. For example, although *brag* means "*show off*", neither of the terms "*show*" and "*off*" has the negative orientation. Idiomatic expressions often do not inherit the sentiment orientation from or to the words in the glossary.

To decrease the effect of the first limitation, (Hassan and Radev, 2010) considered the pair of word/part-of-speech in the nodes of graph, rather than only word. After constructing the graph, they predict the SO of a lexicon with the following steps. First, they used a random walk model to compute the polarity of a word at node i with unknown polarity. The walk model starts from the word and moves to a node j with a transition probability. The walk continues until hitting a word with a known polarity. The average time a random walk starting at i takes to hit the set of positive/negative nodes is an indicator of its polarity. Transition probability between any two nodes i and j can be computed by normalizing the weights of the edges out of node i and it is defined as the following equation:

$$P_{t+1|t}(j|i) = w_{ij} / \sum_k w_{ik} \tag{2.8}$$

where w_{ij} is the weight of the edge from node i to node j and k represents all nodes in the neighbourhood of i. $P_{t+1|t}(j|i)$ denotes the transition probability from node i at step t to node j at time step $t+1$. Then, they defined the hitting time based on the transition probability as follows:

$$h(i|S) = \begin{cases} 0, & i \epsilon S \\ \sum_{j \epsilon V} p_{ij} \times h(j|S) + 1, & otherwise \end{cases} \tag{2.9}$$

where $h(i|S)$ is the average number of steps a random walker, starting in state i not in S, will take to enter a state in S for the first time, and S is a subset of V which is all words in the graph. p_{ij} is the transition probability from i to j.

Second, for any given word i, they compute the hitting time $h(i|S+)$, and $h(i|S-)$ for the two seed sets (seven positive and seven negative) iteratively as described earlier. If $h(i|S+)$ is greater than $h(i|S-)$, the word is classified as negative, otherwise it is classified as positive. The ratio between the two hitting times could be used as an indication of how positive/negative the given word is.

Finally, they achieved accuracy 82.1% with 14 seeds and the dataset General Inquirer lexicon (Stone et al., 1966). Their accuracy is not significantly higher than spin model (Takamura, Inui, and Okumura, 2005). There are several following major shortcomings in the approaches presented in this section (dictionary-based and corpus-based approaches):

- All the above approaches do not consider context (in the graph for dictionary-based approach). That is each node takes only one static SO and there is not any node with dynamic polarity according to the context.

- These approaches do not work with ambiguous sentiment words with dynamic sentiment orientation. Some of the ambiguous words were removed in most of the existing research (Turney, 2002; Takamura, Inui, and Okumura, 2005; Hassan and Radev, 2010).

- All the above approaches need some words as seeds or external resources (e.g., reviews with known ratings).

2.3.3.2 Contextual Sentiment Prediction and Ambiguous Sentiment Words

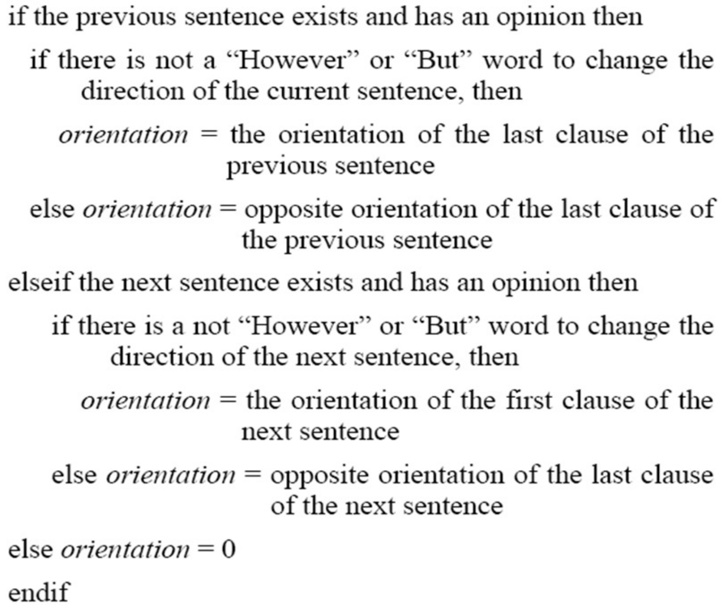
It is well known that there is no universally optimal sentiment lexicon since the polarity of words is sensitive to the topic domain (Pang and Lee, 2008; Liu, 2010; Tang, Tan, and Cheng, 2009). For example, "*unpredictable*" is negative in the electronics domain while being positive in the movie domain (Turney, 2002). To address this problem, (Hatzivassiloglou and McKeown, 1997) employed the synthetic or co-occurrence patterns in the text. They extracted conjunctions of adjectives from a given corpus and labelled each two conjoined adjectives as being of the same orientation, such as "*simple and well-received*" or different orientation, such as "*simplistic but well-received*". The result is a graph of adjectives connected by same-orientation or different-orientation links that they clustered into two subsets of adjectives by an optimization procedure on each connected component. They labelled as positive the cluster which has the highest average frequency of words. Their approach will probably works only with adjectives because there is nothing wrong with conjunctions of nouns or verbs with opposite polarities (e.g., "*war and peace*", "*rise and fall*", "*fat and beautiful*") (Hassan and Radev, 2010).

Indeed, sentiment lexicons adapted to the particular domain or topic have been shown to improve task performance in a number of applications, including opinion retrieval (Na et al., 2009; Jijkoun, de Rijke, and Weerkamp, 2010), and expression level sentiment classification (Choi and Cardie, 2009). Nevertheless, little attention has been paid to the further challenge that even in the same domain the same word may still indicate different polarities with respect to different aspects in context. (Lu et al., 2011) investigated the sentiment orientation of words (only adjectives and adverbs) that is domain specific (e.g. "*private*" is positive in hotel reviews; "*compatible*" is positive about printers) and dependent on the aspect in context (e.g. "*huge room*" vs. "*huge price*" for hotels; "*cheap ink*" vs.

"*cheap appearance*" for printers). They assigned a sentiment score to each aspect and opinion word combination (e.g. BATTERY: large: -1). They employed the following four heuristic constraints (evidences) and combine them in the objective function of an optimization framework:

1. Sentiment prior for the lexicon
2. Overall sentiment ratings in document-level containing the aspect-opinion
3. Similar sentiments which can be collected from synonyms in WordNet or from parsing the opinion collection with sentiment coherency assumption i.e. "*and*" rules as in linguistics heuristics, and
4. Opposite sentiments which are from antonyms in a thesaurus or "*but*" rules in linguistics heuristics.

For the experiments, they used two datasets, hotel reviews and customer feedback surveys on printers and the results demonstrate the advantage of combining all above constraints over using any single one. They employed features obtained from the sentence containing the opinion and rating of the review containing the sentence. (Ding, Li, and Peng, 2008) consider the features obtained from cross-reviews and cross-sentence. They employed the contextual information in other reviews of the same product. For example, in the following sentence "*the battery life is very long*", it is not clearly whether long means a positive or a negative opinion on the product feature "*battery life*". Their approach tries to see whether any other reviewer said that long is positive (or negative). For example, another reviewer wrote "*this camera takes great pictures and has a long battery life*". From this sentence, the context-dependent adjective long is positive for "*battery life*" because it is conjoined with the positive opinion word "*great*". In addition, they used the context of previous or next sentence (or clauses) to decide the orientation of the opinion word. The idea is



```
if the previous sentence exists and has an opinion then
  if there is not a "However" or "But" word to change the
      direction of the current sentence, then
    orientation = the orientation of the last clause of the
                  previous sentence
  else orientation = opposite orientation of the last clause of
                     the previous sentence
elseif the next sentence exists and has an opinion then
  if there is a not "However" or "But" word to change the
      direction of the next sentence, then
    orientation = the orientation of the first clause of the
                  next sentence
  else orientation = opposite orientation of the last clause
                     of the next sentence
else orientation = 0
endif
```

Figure 2.2: adapted from Ding et al. (2008), the context of previous or next sentence (or clauses) is used to decide the orientation of the opinion word

that people usually express the same opinion (positive or negative) across sentences (e.g., "*The picture quality is amazing. The battery life is long*") unless there is an indication of opinion change using words such as "*but*" and "*however*" (e.g., "*The picture quality is amazing. However, the battery life is short*"). They presented the algorithm based on cross-sentence as shown in Figure 2.2.

They showed that handling context dependent opinion words helps significantly for opinion sentence extraction and sentence orientation prediction, because many product aspects will be assigned the neutral orientation without context dependency handling. Although there are many novel approaches to extract the product aspects from reviews, there are only a few simple approaches to predict their sentiment orientation.

The ambiguous sentiment words can appear in any languages beside English. (Wu and Jin, 2010) proposed a knowledge-based method to determine the Sentiment Orientation (SO) of ambiguous sentiment adjectives (ASAs) within context in Chinese language. They claim that the SO of most ASAs can be determined by target nouns in noun-adjective phrases and they employed the modified nouns and three following steps to distinguish the SO of the adjectives. First, they classified the ASAs into two groups: positive-like adjectives and negative-like adjectives based on their observation, such as *large* and *small* are positive-like and negative-like adjectives, respectively. Second, they employed pattern-based and character-based approach to estimate the sentiment expectation of the modified nouns. Finally, they inferred the SO of the adjectives using the following equations:

$$C(a) = \begin{cases} 1, & \text{if a is positive\_like} \\ -1, & \text{if a is negative\_like} \end{cases} \tag{2.10}$$

$$C(n) = \begin{cases} 1, & \text{if n is positive expectation} \\ -1, & \text{if n is negative expectation} \end{cases} \tag{2.11}$$

$$SO(a) = C(a) \times C(n) \tag{2.12}$$

where $C(a)$ denotes the category of ASAs and $C(n)$ denotes the sentiment expectation of nouns. They developed their own dataset with 1338 positive and 1738 negative ASAs in Chinese language[4] and obtained the accuracy 78.52%. In addition, they showed that the disambiguation of 14 ASAs can obviously improve the performance of sentiment classification of product reviews proposed by (Wan, 2008). Their work has the following shortcomings:

- They only employed modified noun from context to do the disambiguation task.
- Although they attempt to predict the sentiment orientation of the adjectives, they assigned $+1$ to positives adjectives and -1 to negative ones without any strength or magnitude.

[4] It is downloadable at `http://semeval2.fbk.eu/semeval2.php?location=download&task_id=3&datatype=test`

Their approach may be language-dependent, because the main step of their approach employs pattern and character based approaches.

2.4 Emotion Analysis

Emotion Analysis or Affective Analysis is a study based on how to create computers that are able to recognize, interpret, and simulate human emotions. The terms *affect*, *mood*, and *emotion* are employed in the emotion analysis area and the exact differences between these terms can be shown with the following examples.

- *Pride can be thought of as feeling good about oneself* (Russell, 2003).

In the above example, the phrase "*feeling good*" is affect and the "*pride*" is an emotion. Mood is the affective (emotional) states that are about nothing specific or about everything in general. For example, when a person is in a depressive mood, the object might be the totality of self; and when a person is in an irritable mood, the object could be anything and anyone. Consequently, the cause of a mood may not always be easy to identify, for instance in the following example:

- *A person can wake up in a bad mood in the morning as a result of a confrontation the previous evening* (Ekkekakis, 2012).

In general, the terms *affect*, *mood*, and *emotion* are mostly used interchangeably, without any attempt at conceptual differentiation (Batson, Shaw, and Oleson, 1992).

Since emotions are the key issue in emotion analysis or affective analysis, it needs to generally define and classify emotions. The emotions are the reaction to the different situations we experience in our environment and they play an important role in the decision-making process and solving problems as well. Some examples of emotions and their mental and physical reactions are as follows:

Fear

- Mental: *It is a distressing emotion aroused by impending danger, evil, pain, etc*[5].
- Physical: *a heightened heartbeat, increased muscle tension.*

Happiness

- Mental: *It is a mental or emotional state of well-being characterized by positive or pleasant emotions ranging from contentment to intense joy*[6].
- Physical: *It is often felt as an expansive or swelling feeling in the chest and the sensation of lightness or buoyancy.*

Sadness

- Mental: *It is an emotional pain associated with, or characterized by feelings of disadvantage, loss, despair, helplessness and sorrow. These feelings of certain things are usually negative*[5].
- Physical: *feeling of tightness in the throat and eyes, and relaxation in the arms and legs.*

Shame

- Mental: *It is the painful feeling arising from the consciousness of something dishonorable, improper, and ridiculous, etc., done by oneself or another*[6].
- Physical: *It can be felt as heat in the upper chest and face.*

Desire

- Mental: *It is Desire is a sense of longing for a person or object or hoping for an outcome*[5].

[5] http://www.wikipedia.org/

[6] http://dictionary.reference.com/

- Physical: *It can be accompanied by a dry throat, heavy breathing, and increased heart rate.*

Some existing research presented lexical approaches and employed keyword-spotting techniques (Olveres et al., 1998; Strapparava and Mihalcea, 2007) for emotion analysis. (Aman and Szpakowicz, 2008) used a machine learning model that utilized corpus-based features and the following lexicons: Roget's Thesaurus (Jarmasz and Szpakowicz,) and WordNet-Affect (Strapparava and Valitutti, 2004). (Katz, Singleton, and Wicentowski, 2007) presented a supervised approach based on unigram model, and (Strapparava and Mihalcea, 2008) proposed the methods using LSA and Naïve Bayes to investigate the in news headlines. (Chaumartin, 2007) proposed a rule-based approach using WordNet-Affect (Strapparava and Valitutti, 2004) and SentiWordNet (Esuli and Sebastiani, 2006). The approach was applied to emotion analysis in news headlines.

(Neviarouskaya, Prendinger, and Ishizuka, 2010) presented a novel rule-based approach in which the rules were employed for semantically distinct verb classes. The approach involves three following stages:

1. classifies sentences according to the nine affect categories (Izard, 1971): *'anger', 'disgust', 'fear', 'guilt', 'interest', 'joy', 'sadness', 'shame', 'surprise',*
2. assigns the strength of the sentiment, and
3. determines the level of confidence for sentiment.

(Neviarouskaya, Prendinger, and Ishizuka, 2011) proposed another rule-based linguistic approach for affect recognition from text. Their proposed rule-based approach processed each sentence in stages, including symbolic cue processing, detection and transformation of abbreviations, sentence parsing and word/phrase/sentence-level analyses. Their approach can process sentences of different complexity, including simple, compound, complex and complex-compound sentences.

Earlier research showed that there exists a small set of basic (or fundamental) emotions which are central to other emotions (Ortony and Turner, 1990; Izard, 1971). Though there is little agreement about the number and types of basic emotions, some sets of basic emotions are central and generally accepted (Ortony and Turner, 1990). Some sets of emotions introduced in previous research are listed here:

- Anger, aversion, courage, dejection, desire, despair, fear, hate, hope, love, sadness (Arnold, 1960)
- Anger, disgust, fear, joy, sadness, surprise (Ekman, Friesen, and Ellsworth, 1982)
- Desire, happiness, interest, surprise, wonder, sorrow (Frijda, 1986)
- Rage, terror, anxiety, joy (Gray, 1982)
- Anger, contempt, disgust, distress, fear, guilt, interest, joy, shame, surprise (Izard, 1971)
- Fear, grief, love, rage (James, 1884)
- Anger, disgust, elation, fear, subjection, tender-emotion, wonder (McDougall, 1926)

Chapter 3

Predicting the Uncertainty of Sentiment Adjectives in Indirect Answers

Opinion question answering (QA) requires automatic and correct interpretation of an answer relative to its question. However, the ambiguity that often exists in the question-answer pairs causes complexity in interpreting the answers. This study aims to infer yes/no answers from indirect yes/no question-answer pairs (IQAPs) that are ambiguous due to the presence of *ambiguous sentiment adjectives*. We propose a method to measure the uncertainty of the answer in an IQAP relative to its question. In particular, to infer the *yes* or *no* response from an IQAP, our method employs antonyms, synonyms, word sense disambiguation as well as the semantic association between the sentiment adjectives that appear in the IQAP. Extensive experiments demonstrate the effectiveness of our method over the baseline.

3.1 Motivation and Problem Definition

In indirect yes/no question-answer pairs (IQAPs), the *yes* or *no* words do not explicitly appear in the indirect answers. However, *yes* or *no* responses can be inferred by interpreting the given information in IQAPs. It has been shown that 27% of answers to polar questions do not contain a direct *yes* or *no* word and 44% of them fail to convey a clear *yes* or *no* response (Hockey et al., 1997). The inherent uncertainty that exists in indirect answers needs to be captured to effectively interpret such answers (de Marneffe, Manning, and Potts, 2010). It is common that the answerers express their opinions in indirect manner using adjectives with different degree of strength (certainty), e.g. *terrible* has stronger strength than *bad*.

Existing research showed that adjectives are dominant elements to express opinions (Hatzivassiloglou and McKeown, 1997; Turney, 2002). In the review domain, although most of the adjectives have static sentiment orientation (SO), positive or negative, the SO of some adjectives vary with context. For example, the adjective *high* has *positive* SO in the phrase *high quality* and *negative* SO in *high cost*. The adjectives with dynamic SO in different contexts are called *Ambiguous Sentiment Adjectives* (ASAs) (Wu and Jin, 2010; Balahur and Montoyo, 2010). Recent works introduced a limited number of ASAs such as *young*, *many*, *high*, *thick*; they considered other adjectives, like *good* or *terrible*, as unambiguous (Wu and Jin, 2010).

In the IQAP domain, we observed that all the ASAs introduced in the review domains can also be ambiguous in this domain. Take the following IQAPs as examples:

E1) A: Is he ***qualified***? B: He is ***young***.

E2) A: Is he ***active***? B: He is ***young***.

The answers in E1 and E2 contain the ASA *young*. In E1, the answer conveys *no* and in E2 the answer conveys *yes* relative to the adjective used in the questions, i.e. *qualified* in E1 and *active* in E2.

Furthermore, our observation shows that all the adjectives can be potentially ambiguous in the IQAP domain. In the following examples E3 adapted from (de Marneffe, Manning, and Potts, 2010), the adjective *good* expresses weaker strength than *excellent*, and thus the asker infers that the answerer conveys *yes*:

E3) A: Do you think that's a ***good*** idea, that ...?

B: I think it's an ***excellent*** idea.

However, the adjective *good* takes a dynamic certainty with respect to the question and does not convey *yes* all the time, e.g. in E3, if we reverse the adjectives of question and answer then speakers infer that the answer conveys *no* (de Marneffe, Manning, and Potts, 2010). Thus, the adjective *good* which is always employed to express positive opinions in the review domain, can convey *yes* or *no* in different IQAPs (depending on the adjectives that appear in the question parts). We refer to such adjectives that can be employed in the answers of IQAPs and convey both *yes* and *no* in different answers as ambiguous sentiment adjectives (ASAs) in the IQAP domain.

In this chapter, we investigate IQAPs in which polar questions and their corresponding answers contain a sentiment adjective, such as *young*, *good*, *provocative* and *etc.* Therefore, the task is to automatically infer the answer of a given IQAP as *yes* or *no.*

The rest of this chapter is organized as follows. Section 3.2 explains our method for inferring IQAP answers. Section 3.3 reports the experimental results and evaluation of our method. Section 3.4 discusses the problems in inferring the answers of IQAPs. Finally, Section 3.5 summarizes the chapter.

3.2 Method

Our method has four main stages to infer the *yes* or *no* answers to IQAPs. First, we measure the certainty of the answer relative to its question for all IQAPs. Second, for each IQAP, we compute a threshold to evaluate the certainty of answer toward *yes* or *no* responses. Third, we infer the answers in each IQAP using the certainty of its answer and its obtained threshold. Finally, we present a refinement on the method by using synonyms. We explain these stages in the subsequent sections respectively.

3.2.1 Assigning Degree of Certainty to Answers

In this section, we aim to compute the certainty of an answer relative to its question in a given IQAP. Such certainty can be computed based on the association between the adjective of the question (SAQ) and the adjective of the answer (SAA). If the association between the SAQ and SAA is high, then the certainty of the answer relative to its question will be high.

Any similarity measure can be employed to estimate the association between SAQ and SAA. We here use two popular measures, Pointwise Mutual Information (PMI) and Latent Semantic Analysis (LSA) for this purpose. PMI between two words measures their mutual dependence and is defined as follows (Turney, 2002):

$$PMI(w_1, w_2) = \log_2 \left(\frac{P(w_1, w_2)}{P(w_1)P(w_2)} \right) \tag{3.1}$$

where $P(w_1, w_2)$ is the probability that w_1 and w_2 co-occur in the same context (e.g., a fixed window or sentence), and $P(w_1)$ and $P(w_2)$ are the probability of w_1 and w_2 in the entire corpus. Since PMI requires a large corpus to be effective, in our experiments, we employ a large corpus of 1.5M reviews explained in Section 3.3 to calculate PMI between the words, and consider co-occurrence of two words in less than five words distance between them.

LSA is another learning method for computing similarity between words (Landauer, Foltz, and Laham, 1998). It works based on analyzing relationships between a document and the words that it contains. LSA performs several steps to compute the association between two words. First, it forms a matrix with documents as rows and words as columns. Cells contain the number of times that a given word is used in a given document. Second, it employs Singular Value Decomposition (SVD) to represent the words and documents as vectors in a high dimensional semantic space. In a new matrix, words are represented as vectors. Finally, similarity of two words is computed as the cosine between their corresponding vectors in the semantic space. The value of cosine will be $+1$ for identical meanings, zero for unrelated meanings and -1 for opposite meanings.

We obtained LSA values using the TASA corpus[1] instead of the corpus that we employed for PMI. This is because LSA is computationally expensive with large corpora. In fact, this is because LSA uses a word-by-document matrix and the cost of computation increases substantially with a very large corpus.

3.2.2 Defining a Threshold

As we discussed above, we need to know whether the answer in an IQAP has enough certainty to convey a *yes* or *no* response. We compute a threshold for this purpose which can vary in different IQAPs. Since the antonym of a word belongs to the same scalar group (e.g., *hot* and *cold*) and has different sentiment orientation with the word (Hatzivassiloglou and McKeown, 1997), we can utilize the antonym of the SAQ to compute a threshold for each IQAP. Our intuition is that if the association strength between an SAA and its corresponding SAQ is greater than the association strength between the SAA and the antonym of its SAQ, the answer has enough degree of certainty to convey *yes*, and otherwise the answer is more likely

[1] LSA is obtained from: `http://cwl-projects.cogsci.rpi.edu/msr`

to be uncertain relative to the question and conveys a *no* response. For example, in E1, the association strength between *young* and *qualified* is smaller than the association between *young* and *unqualified*; therefore, the answer conveys *no*.

We find the antonym of an SAQ in two steps. First, we employ the IMS word sense disambiguation system (Zhong and Ng, 2010) to detect the sense of the SAQ. Then, we use WordNet to get the antonym of the SAQ based on its predicted sense. In WordNet, different senses of a word can have different antonyms.

3.2.3 Inferring Yes or No Answers

The following decision procedure employs two preceding steps to decide what a given answer conveys:

$$answer = \begin{cases} yes, & assoc(SAQ, SAA) > assoc(\sim SAQ, SAA) \\ no, & assoc(SAQ, SAA) < assoc(\sim SAQ, SAA) \\ uncertain, & otherwise \end{cases} \tag{3.2}$$

where $assoc(.,.)$ indicates our similarity measure (either *PMI* or *LSA*), and $\sim SAQ$ is antonym of the SAQ. Note that the appearance of a negation word in the answer to a question reverses the inferred answer, thus here it flips *yes* and *no* responses, but *uncertain* remains unchanged.

3.2.4 Refining Using Synset

In this section we propose to use the synonyms of the SAAs to supplement our method with more information about the SAAs. Since the synonym of a word is a word that has the same or nearly the same meaning as the original word, the SAA can be replaced by any of its synonyms with no major changes in its inferred original answer. In addition, different senses of an SAA may have different sets of synonyms (synsets) in WordNet. We can obtain the synset of SAAs using a word sense disambiguation system and WordNet in a similar way that we did for antonyms in Section 3.2.2. Having the synset of an SAA, we compute the association between SAQ and the synset of the SAA as follows:

$$assoc(SAQ, syn(SAA)) = \frac{1}{|synset|} \sum_{i=1}^{|synset|} assoc(SAQ, syn_i(SAA)) \quad (3.3)$$

where $syn(SAA)$ is the synset of the SAA, and $syn_i(SAA)$ is the i^{th} word in $syn(SAA)$. Equation 3.3 computes the association between SAQ and the synset of SAA by averaging the sum of the association between SAQ and each of the synonyms for SAA.

As we discussed before, the antonym of an SAQ can be used to decide about the certainty of the answer for an IQAP as *yes* or *no*. In Section 3.2.3 the antonym has been used with the SAA itself, i.e. $assoc(\sim SAQ, SAA)$. Here we use the synset of the SAA to predict the association between $\sim SAQ$ and the SAA more precisely. The following Equation can be used for this purpose:

$$assoc(\sim SAQ, syn(SAA)) = \frac{1}{|synset|} \sum_{i=1}^{|synset|} assoc(\sim SAQ, syn_i(SAA)) \quad (3.4)$$

We can use the above two equations to infer the *yes* or *no* response as follows:

$$answer = \begin{cases} yes, & assoc(SAQ, syn(SAA)) > assoc(\sim SAQ, syn(SAA)) \\ no, & assoc(SAQ, syn(SAA)) < assoc(\sim SAQ, syn(SAA)) \\ uncertain, & otherwise \end{cases} \quad (3.5)$$

3.3 Evaluation and Results

In this section we first explain the datasets that we used in this research, and then report the experiments conducted to evaluate our approach.

We used the dataset developed in (de Marneffe, Manning, and Potts, 2010) to evaluate our method. This dataset contains a set of IQAPs and their corresponding *yes* or *no* labels as its ground truth. It includes 125 IQAPs with two different sentiment adjectives in any question-answer pair as described in (de Marneffe, Manning, and Potts, 2010). They used two sources to gather the IQAPs: five different shows from online CNN interview transcripts, and the Switchboard Dialog Act corpus. They manually annotated the IQAP dataset for *yes* or *no* responses and identified the adjectives of the questions and answers. In all instances of IQAPs, the SAA is different from the SAQ.

We also used two datasets as development datasets to compute the association strength of word pairs based on PMI and LSA measures. To compute the co-occurrence information for PMI, we collected a large corpus of 1.5M reviews from Amazon product reviews for 25 different product types, such as *book*, *video*, and *music*. However, as we discussed in Section 3.2.1, LSA in contrast to PMI cannot handle large corpora (Lindsey et al., 2007). Therefore we employed the standard Touchstone Applied Science Associates (TASA) corpus to compute the association strength of word pairs using LSA. The TASA corpus is a collection of texts from textbooks, literature, works of fiction and nonfiction used in schools and the reading materials that a person is supposed to have been exposed to by his first year in college. This corpus contains more than 17M tokens corresponding to around 155K different types.

3.3.1 Experimental Results

In this section we report detailed results of our approach with different configurations. We compare the approach proposed in (de Marneffe, Manning, and Potts, 2010) as a baseline.

Given an IQAP, de Marneffe, Manning, and Potts (2010) assigned a sentiment orientation (SO, referred as expected rating value by authors) to both SAA and SAQ of the given IQAP, and then interpreted the answer based on the SOs. For instance, if the SOs of the SAA and SAQ have different signs, then the answer conveys *no*. In case of the same sign, if the SO of the SAA is greater than or equals to the SO of the SAQ, then the answer conveys *yes*, and otherwise *no*. They obtained an accuracy of 60% on the same IQAP dataset which is used in our experiments. They used an external source (a large corpus of reviews with ratings) to compute the SO of adjectives. Given an adjective, they computed the SO of the adjective as a function of the probability of rating given the adjective.

Their approach assigns a globally fixed SO score to each adjective. For example, the adjectives *best* and *great* are assigned the fixed SO scores of 1.08 and 1.1 respectively. This approach ignores the context in which the adjectives appear (i.e. the IQAP). However, in our approach the degree of certainty for the same answer may change in different IQAPs. This dynamic degree of certainty not only depends on the SAA itself but also on the SAQ that appears in the IQAP. So, our method utilizes the context information better than the method proposed in (de Marneffe, Manning, and Potts, 2010).

Table 3.1 shows the results of different approaches in terms of precision, recall and f-measure. The results in Table 3.1 are based on Equation 3.5 where we use antonyms (of SAQs) and synsets (of SAAs) to infer the yes or no answers. The first and second rows of Table 3.1 shows the result of our method when it uses PMI and LSA respectively. The last row shows the baseline results.

Similarity Measures	**Precision**	**Recall**	**F-Measure**
PMI-synset-antonym	67.14	65.45	66.28
LSA-synset-antonym	**73.97**	**75.98**	**74.96**
Baseline	60.00	60.00	60.00

Table 3.1: Performance of the approaches based on semantic similarity measures on IQAP inference task and their comparison with the sate-of-the-art approach

As it is clear from Table 3.1, when we use LSA our method achieves better performance than PMI. It was expected since PMI is known as a contextual similarity measure while LSA is known as a semantic similarity measure. So, LSA can better measure the semantic association between the adjectives which can definitely help the inference process. Our method using both PMI and LSA significantly outperforms the baseline method.

3.4 Analysis and Discussion

In this section, we discuss the effectiveness of our method from different perspectives. As we mentioned before, our method utilizes synset, antonym, and word sense disambiguation techniques. In this section, we dig into the IQAP problem and investigate the effectiveness of these techniques to tackle the IQAP problem. In Section 3.4.1, we analyze the role of synsets and antonyms, while in Section 3.4.2 we discuss the role of WSD.

3.4.1 Role of Synsets and Antonyms

In this section, we evaluate the effectiveness of synsets and antonyms for inferring yes or no answers. For this purpose, we repeat the experiments by ignoring synsets or antonyms respectively. Table 3.2 shows the results.

In Table 3.2, PMI-Antonym (LSA-Antonym) shows the results when we use Equation 3.2 to infer the answer of an IQAP based on the PMI (LSA) measure. In Equation 3.2, we only use SAA, SAQ, and $\sim SAQ$ and do not utilize the synsets (of SAAs). Table 3.2 shows that ignoring the synsets

Similarity Measures	Precision	Recall	F-Measure
PMI-antonym	58.94	56.18	57.53
LSA-antonym	62.23	60.88	61.55
PMI-synset	34.86	41.69	37.97
LSA-synset	67.35	56.86	61.66
PMI	32.20	34.38	33.25
LSA	66.70	54.95	60.26

Table 3.2: Experimental results on IQAP inference task using semantic similarity measures and without using synsets or antonyms

results in significant reduction in the final performance, i.e. from 66.28% (see Table 3.1) to 57.53% for PMI and 74.96% to 61.55% for LSA.

This result shows that synsets are highly effective for IQAP problem. We believe one of the reasons is about the fact that some SAAs and SAQs never (or rarely) co-occurred in our large corpus. This results in a very low association between them. However the synonyms of the SAAs may frequently occur with the SAQs. Therefore, the synonyms help us to more reliably predict the association between the SAQs and SAAs and consequently better infer the yes or no responses. Similar to PMI, LSA can benefit from synsets. In fact, as the result shows, LSA benefits more from the synsets than PMI. The reason is that our LSA measure uses a smaller corpus (TASA) than PMI. Therefore, it is more likely that an SAA does not appear in the LSA corpus than PMI corpus. In that sense, LSA should benefit more from the synsets than PMI.

To investigate the role of antonyms, we repeat the experiments without using them. In other words, for each IQAP, the answer is interpreted only based on the association between the SAQ and the synset of the SAA. Given an IQAP, if the similarity association between SAQ and the synset of the SAA is positive, then the inferred answer will be *yes*; if it is negative, the inferred answer will be *no*, and otherwise, it will be *uncertain*. The results of these experiments are shown as PMI-synset and LSA-synset in Table 3.2 for PMI and LSA respectively.

As it is clear from Table 3.2, the antonyms can also help to infer the correct answer comparing to PMI-synset-antonym or LSA-synset-antonym

Method	Precision	Recall	F-Measure
PMI	66.35	65.59	65.97
LSA	73.02	74.66	73.83

Table 3.3: Experimental results on IQAP inference task using semantic similarity measures and without using WSD

(See Table 3.1). The results of both PMI and LSA have significantly decreased when we do not use antonyms, from 66.28% to 37.97% for PMI and 74.96% to 61.66% for LSA. In addition, it is notable that the performance of PMI decreased more than LSA (from 66.28% to 37.97%).

Finally, we apply the proposed method without using synsets and antonyms. Here, for each IQAP, the answer is interpreted only based on the association between the SAQ and the SAA. Given an IQAP, if the similarity association between SAQ and the SAA is positive, then the inferred answer will be *yes*; if it is negative, the inferred answer will be *no*, and otherwise, it will be *uncertain*. The results of these experiments are shown in the last two rows of Table 3.2. As expected, we see the lowest performance when we do not utilize both synonyms and antonyms.

3.4.2 Role of Word Sense Disambiguation

In our method, we employed an automatic WSD system and obtained 66.28% and 74.96% performance using PMI and LSA respectively (see Table 3.1). Here, we study the impact of the WSD system on these results.

For this purpose, instead of the WSD system we only used the most common sense of the adjectives (the first sense in WordNet) and repeat the experiments. We took the most common sense as a replacement for the WSD system because it has been shown as a strong baseline in the WSD area. The results are shown in Table 3.3. As it is clear, using the most common sense of the adjectives slightly reduces the performance, from 66.28% to 65.97% for PMI and 74.96% to 73.83% for LSA.

It is notable that, in this experiment, the WSD system has assigned the most common sense to around 80% of the adjectives. In other word, only 20% of the adjectives assigned senses different than their most common senses. The efficiency of the WSD could have been more highlighted, if more IQAPs contain adjectives with senses different from their most common senses.

3.5 Summary

In this study, we examine the behaviour of adjectives in Indirect yes/no Question-Answer Pairs (IQAPs) domain. In particular, our task is to automatically detect whether the answer of a given IQAP conveys *yes* or *no.* We show that measuring the association between the adjectives in question and answer can be a main factor to infer a clear response from an IQAP. We utilize antonyms, synonyms and word sense disambiguation to tackle the IQAP problem and investigate the effectiveness of each of these techniques for this task.

The work in this chapter has been presented in the 20th ACM Conference on Information and Knowledge Management, CIKM 2011 (Mohtarami et al., 2011).

Chapter 4

Sense Sentiment Similarity through Emotional Space

Semantic similarity measures have been employed in Chapter 3 to estimate the similarity between opinion words. However, this chapter shows that semantic similarity measures are less effective in capturing the similarity with respect to the sentiment. This makes a need of sentiment similarity measure. *Sentiment similarity* indicates the similarity between two words from their underlying sentiments. This chapter proposes an emotion-based approach to acquire sentiment similarity of word pairs with respect to their senses. Our approach is built on a model which maps from senses of words to vectors of twelve basic emotions. The emotional vectors are used to measure the sentiment similarity of word pairs. We show the utility of measuring sentiment similarity in two main natural language processing tasks, namely, indirect yes/no question answer pairs (IQAP) Inference and sentiment orientation (SO) prediction. Extensive experiments demonstrate that our approach can effectively capture the sentiment similarity of word pairs and utilize this information to address the above mentioned tasks.

4.1 Motivation and Problem Definition

This work focuses on the task of measuring sentiment similarity of word pairs. Sentiment similarity reflects the distance between words regarding their underlying sentiments. Many approaches have been proposed to capture the semantic similarity between the words to date; Latent Semantic Analysis (LSA), Point-wise Mutual Information (PMI), and WordNet-based similarity method are some examples of the semantic similarity measures.

These measures are good for relating semantically related words like "*car*" and "*automobile*", but are less effective in relating words with similar sentiment like "*excellent*" and "*superior*". For example, the following relations show the semantic similarity between some sentiment word pairs computed by LSA (Landauer, Foltz, and Laham, 1998) and their arranged relations.

$$\begin{aligned} &LSA(excellent, superior) = 0.40 \\ &\qquad < LSA(excellent, good) = 0.46 \\ &\qquad\qquad < LSA(good, bad) = 0.65 \end{aligned}$$

Clearly, the sentiment similarity between these words should be in the reversed order. In fact, although the terms "*excellent*", "*superior*" and "*good*" have the same sentiment orientation (positive), the intensity of sentiment in "*excellent*" is more similar to "*superior*" than "*good*". Thus, ideally, sentiment similarity of "*excellent*" and "*superior*" should be greater than "*excellent*" and "*good*" and as the terms "*good*" and "*bad*" are opposite in sentiment, their sentiment similarity should be zero.

To date, sentiment similarity has not received enough attention. In fact, the majority of existing works employed semantic similarity as a measure to compute sentiment similarity of word pairs (Kim and Hovy, 2004; Turney and Littman, 2003). In this study, we propose a principled approach to detect the sentiment similarity of word pairs with respect to their senses and their underlying sentiments. We introduce 12 basic emotions dedicated to sentiment similarity. Our method computes the sentiment similarity of word pairs based on the connection between their lexical semantics and basic emotions. We show that it effectively outperforms the semantic similarity measures that were used to predict sentiment similarity.

Furthermore, we show the utility of sentiment similarity prediction in two NLP tasks, namely, *Indirect yes/no Question Answer Pairs* (IQAPs) *Inference*, *Sentiment Orientation (SO) prediction.* We briefly explain the utility of sentiment similarity for these two tasks:

In IQAPs, as earlier explained in Section 3.1, the answer of a question-answer pair does not explicitly contain a clear *yes* or *no* word, but rather gives information which can be used to infer such an answer. Therefore, the task is to infer the *yes* or *no* answer for a given question-answer pair. Table 4.1 shows further examples of IQAPs with different degree of *yes* or *no.* In some cases, interpreting the answer is straightforward, e.g. E1, but in many cases the answerer shifts the topic slightly, e.g. E2 and E3. In these cases, the interference task is more difficult.

Clearly, the sentiment words of the question and answer of an IQAP are the pivots that determine the final answer as *yes* or *no.* We show that the sentiment similarity between the adjectives in the IQAPs can be used to effectively infer the *yes* or *no* answers. For example, in E1, though the adjective "*acceptable*" has weaker sentiment intensity than the adjective "*great*", the sentiment similarity between the two adjectives is sufficiently high to infer a *weak-yes* answer. However, if the answer contains an adjective with higher sentiment similarity with "*great*", e.g. "*excellent*", then the answer would be inferred as *strong-yes.* This is the same for other examples.

As the second application, we predict the sentiment orientation of words. Existing research utilized (a) word relations obtained from WordNet (Kim and Hovy, 2004; Hassan and Radev, 2010), (b) external resources like

Row	IQAP	Answer
E1	Q: Do you think that's a ***great*** idea? A: I think it's ***acceptable***.	*weak-yes*
E2	Q: Was she the ***best*** one on that old show? A: She was simply ***funny***.	*strong-yes*
E3	Q: He says he opposes amnesty, but .... Is he ***right***? A: He is a bit ***funny***.	*weak-no*
E4	Q: ... Is that ***true***? A: This is ***extraordinary*** and preposterous.	*strong-no*

Table 4.1: Examples of IQAPs

review rating (de Marneffe, Manning, and Potts, 2010), and (c) semantic similarity measures for this purpose (Turney and Littman, 2003; Kanayama and Nasukawa, 2006). We show that sentiment similarity is a more appropriate measure to achieve accurate sentiment orientation of words.

The sentiment similarity may also vary with respect to different senses of the words. For example, in E4, if we use the third sense of the adjective "*extraordinary*", i.e. "*unusual*", we can infer the correct answer, *no*. This is because the sentiment similarity between "*unusual*" and "*true*" is low. This is while the first sense (the most common sense) of "*extraordinary*" means "*bonzer*" that has sufficiently strong sentiment similarity with the adjective "*true*". Therefore the answer will be incorrectly interpreted as *yes* in the latter case.

In summary, the contributions of this chapter are follows:

- We propose an effective method to predict the sentiment similarity between word pairs at the sense level,
- We show that such sentiment similarity can better reflect the similarity between sentiment words than semantic similarity measures, and
- We show the utility of sentiment similarity in IQAP inference and SO prediction tasks.

The experiments in sentiment prediction show that our sentiment similarity method significantly outperforms two baselines by 6.85% and

Figure 4.1: Examples of affective emotional states; this figure illustrates that human have different feelings and reactions with respect to different emotions

18.1% improvements in F1. It also outperforms the best performing baseline for the IQAP task by 17.93% improvements in F1.

4.2 Method: Sense Sentiment Similarity

People often show their sentiment with various emotions, such as "*crying*" or "*laughing*". Although the emotions can be categorized into *positive* and *negative* sentiments, human have different feelings with respect to each emotion. For example, "*anger*" and "*fear*" have *negative* sentiments; however they reflect different feelings. Figure 4.1 illustrates different human reactions with respect to different emotions. The intensity of sentiment in each emotion is different from others.

Human behavior as a result of his emotions can be presented via the look on his face (e.g., Figure 4.1), the sound of his voice, or opinion words expressed in his writing/speaking. Since the opinion words carry a range of human emotions, they can be represented as a vector of emotional intensities. Emotion intensity values describe the intensity degree of emotions that can be varied from "*very weak*" to "*very strong*". For example, Table 4.2, adapted from Neviarouskaya, Prendinger, and Ishizuka (2009), shows several sample opinion words and their corresponding intensity values with respect to different emotions. For example, the verb "*regret*" has intensity values of 0.2 and 0.1 with respect to the "*guilt*" and "*sadness*" emotions respectively.

We propose to predict the sentiment similarity between the senses

Word	POS	Intensity Values
tremendous	adj.	surprise:1.0; joy:0.5; fear:0.1
success	noun	joy:0.9; interest:0.6; surprise:0.5
regret	verb	guilt:0.2; sadness:0.1

Table 4.2: Examples of words with emotional intensities with respect to the set of emotions: e = [*anger, disgust, fear, guilt, sadness, shame, interest, joy, surprise*]

of the words using the words' emotional vectors constructed from their intensities. We follow three steps to achieve this aim:

4.2.1 Designing Basic Emotional Categories

Previous research showed that there exists a small set of basic (or fundamental) emotions which are central to other emotions (Ortony and Turner, 1990; Izard, 1971). Though there is little agreement about the number and types of basic emotions, some sets of basic emotions are central and generally accepted (Ortony and Turner, 1990).

We use the emotional set studied in (Izard, 1971; Neviarouskaya, Prendinger, and Ishizuka, 2009) as its basic emotions appear in more number of emotional sets and have higher coverage than others. The basic emotions are: *anger, disgust, fear, guilt, sadness, shame, interest, joy, surprise.* We considered the first six emotions as negative emotions and the other three as positive. To have a balanced number of positive and negative emotions, we also employ three other positive basic emotions adapted from Ortony and Turner (1990): *desire, love, courage.*

We extend each basic emotion to an emotional category. For this purpose, we use the hierarchical synonyms of the basic emotions; we refer to these words as seeds. For each basic emotion, we pick its synonyms with the following constraints:

Desire	**Joy**	**Sadness**
cherished, 0.54	delight, 0.63	depressive,0.55
enthusiasm, 0.47	excitement, 0.6	sad, 0.54
ambition, 0.46	happy, 0.59	weepy, 0.54
honest, 0.46	glorious, 0.58	grief, 0.53
intimate, 0.45	pleasure, 0.57	loneliness, 0.51

Table 4.3: Examples of seed words in emotional categories and their semantic similarity values with their corresponding basic emotions

$$\begin{cases} \text{1. Relevant Seeds: Having the highest semantic similarity scores (computed by LSA) with the basic emotion, and} \\ \text{2. Balanced Matrix: The total occurrences of all the selected seeds for each category in our corpus remains balanced over the emotional categories} \end{cases} \tag{4.1}$$

As an example, Table 4.3 shows some selected seeds and their semantic similarity values with their corresponding basic emotions[1].

4.2.2 Constructing Emotional Vectors

In this step, we construct an emotional vector like $(I_1, I_2, ..., I_{12})$ for each word w where each I_k represents the intensity of k^{th} emotion in w. For instance, I_1 represents the intensity of "*anger*" and I_{12} indicates the intensity of "*courage*" emotion in the word w.

We employ the hypothesis that a word can be characterized by its neighbors (Turney and Littman, 2003). That is, the emotional vector of a word tends to correspond to the emotional vectors of its neighbors. Therefore, we use the sum of the co-occurrences of w with each seed in an emotional category to estimate the intensity value of w with the corresponding emotion as shown in Equation 4.2.

[1] Hierarchical synonyms can be obtained from thesaurus.com, and semantic similarity computed by LSA.

$$I_k = Intensity(w, cat_k) = \sum_{seed_j \in cat_k} co\_occur(w, seed_j) \tag{4.2}$$

where, I_k is the overall intensity value of w with the k^{th} emotional category cat_k, $seed_j$ is a seed word in cat_k, and $co\_occur(.,.)$ is the number of times that two words occur in the same window of text.

Note that employing co-occurrence is critical for words whose emotional meanings are part of common sense knowledge and not explicit (e.g., the terms "*mum*", "*ghost*", and "*war*"). The emotional intensity of such words can be detected based on their co-occurrence patterns with words with explicit emotional meanings, e.g. seeds.

In addition, a problem with the corpus-based co-occurrence of w and cat_k is that w may never (or rarely) co-occur with the seeds of an emotion. This results in a very weak intensity value of w in cat_k. We utilize synsets to tackle this issue. As the synset of a word has the same or nearly the same meaning as the original word, the word can be replaced by any of its synset with no major changes in its emotion. Therefore, we expect the synset to improve the predicted value for intensity of w in cat_k and hence better estimate sentiment similarity between words.

Furthermore, the major advantage of using synsets is that we can obtain different emotional vectors for each sense of a word and predict the sentiment similarity at the sense level. Note that, various senses of a word can have diverse meanings and emotions, and consequently different emotional vectors. Using synsets, the intensity value of w in an emotional category is computed by the sum of the intensity value of each word in the synset of w with the emotional category as presented in Equation 4.3.

$$I_k = \sum_{syn_i \in synset(w, sense(w))} Intensity(syn_i, cat_k) \tag{4.3}$$

where, $synset(w, sense(w))$ is the synset of a particular sense of w, and $Intensity(syn_i, cat_k)$ is computed using Equation 4.2.

4.2.3 Word Pair Sentiment Similarity

To compute the sentiment similarity between two words with respect to their senses, we use the correlation coefficient between their emotional vectors. Let **X** and **Y** be the emotional vectors of two words. Equation 4.4 computes their correlation:

$$corr(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)S_X S_Y} \tag{4.4}$$

where, $n = 12$ is number of emotional categories, $\bar{X}$, $\bar{Y}$ and S_X, S_Y are the mean and standard deviation values of **X** and **Y** respectively.

The above Equation measures the strength of a linear relationship between two vectors. This value varies between -1 (strong negative) and +1 (strong positive). The strong negative value between two vectors means they are completely dissimilar, and the strong positive value means the vectors have perfect similarity.

Given the correlation value between two words, the problem is that how large the correlation value should be such that we can consider the two words as similar in sentiment. We address this issue by utilizing the antonyms of the words. For this purpose, we take an approach similar to our work in Chapter 3. Since the antonym of a word belongs to the same scalar group (e.g., *hot* and *cold*) and has different sentiment orientation with the word, we consider two words, w_i and w_j as similar in sentiment iff they satisfy both of the following conditions:

1. $corr(w_i, w_j) > corr(w_i, \sim w_j)$,and
2. $corr(w_i, w_j) > corr(\sim w_i, w_j)$

where, $\sim w_i$ and $\sim w_j$ are antonyms of w_i and w_j respectively, and $corr(w_i, w_j)$ is the correlation between the emotional vectors of w_i and w_j obtained from Equation 4.4. Finally, we compute the sentiment similarity (SS) between two words as follows:

$$SS(w_i, w_j) = corr(w_i, w_j) - Max\{corr(w_i, \sim w_j), corr(\sim w_i, w_j)\} \tag{4.5}$$

A positive value of $SS(.,.)$ indicates that the words are sentimentally similar. The large value indicates strong similarity and small value shows weak similarity. Likewise, a negative value of $SS(.,.)$ shows the amount of dissimilarity between the words.

4.3 Applications

In this section we explain how sentiment similarity can be used to perform IQAP inference and predict the sentiment orientation of words respectively.

4.3.1 IQAP Inference

In IQAPs, the adjectives in the question and its corresponding answer are the main factors to infer *yes* or *no* answers. We employ the association between the adjectives in questions and their answers to interpret the indirect answers. Table 4.4 shows the algorithm we used for this purpose. Note that $SS(.,.)$ indicates sentiment similarity computed by our method (see Equation 4.5). As we discussed before, the positive SS between words means they are sentimentally similar which can vary from *weak* to *strong*, this leads to infer *weak-yes* or *strong-yes* response that conveys *yes*. However, negative SS indicates that the words are not sentimentally similar and results in *weak/strong-no* which leads to the *no* response.

4.3.2 Sentiment Orientation Prediction

We aim to compute more accurate sentiment orientation (SO) using our sentiment similarity method than any other semantic similarity measures.

Turney and Littman (2003) proposed a method in which the sentiment orientation of a given word is calculated from its contextual/semantic similarity with seven positive words like "*excellent*", minus its similarity with seven negative words like "*poor*" as shown in Table 4.5.

Inputs:
SAQ: The adjective in the question of the given IQAP.
SAA: The adjective in the answer of the given IQAP.

Output:
answer $\in \{yes, no, uncertain\}$

Algorithm:

```
if $SAQ$ or $SAA$ are missing from our corpus then
    answer = $Uncertain$;
else if $SS(SAQ, SAA) < 0$ then
        answer = $No$;
    else if $SS(SAQ, SAA) > 0$ then
            answer = $yes$;
```

Table 4.4: Decision procedure of employing sentiment similarity for IQAP inference task

Inputs:
$Pwords$: seven words with positive sentiment orientation
$Nwords$: seven words with negative sentiment orientation
$A(.,.)$: similarity function that measures the similarity between its arguments
w: a given word with unknown sentiment orientation

Output:
P: sentiment orientation of w

Algorithm:

```
$P = SO\_A(w) =$
    $\sum_{pword \in Pwords} A(w, pword) - \sum_{nword \in Nwords} A(w, nword)$
```

Table 4.5: Procedure to predict sentiment orientation (SO) of a word based on the similarity function $A(.,.)$

As a similarity function, $A(.,.)$, they employed point-wise mutual information (PMI) and LSA to compute the similarity between the words. We utilize the same approach, but instead of PMI or LSA we use our $SS(.,.)$ measure as the similarity function.

PMI has been earlier defined in Section 3.2. We reproduce its formula below for easy reference.

$$PMI(w_1, w_2) = \log_2 \left(\frac{P(w_1, w_2)}{P(w_1)P(w_2)} \right) \tag{4.6}$$

where $P(w_1, w_2)$ is the probability that w_1 and w_2 co-occur, and $P(w_1)$ and $P(w_2)$ are the probability of w_1 and w_2.

As discussed in Section 3.2, LSA performs several steps to compute the semantic similarity between two words. First, it forms a matrix with documents as rows and words as columns. Cells contain the number of times that a given word is used in a given document. Second, it attempts to reduce the high dimensional semantic space and compute the similarity of two words by the cosine between their corresponding vectors in the semantic space. LSA and to some extent PMI only utilize the semantic space and ignore the emotional space, whereas our SS measure effectively utilizes the emotional space.

4.4 Evaluation and Results

In this section we first explain the datasets used, and then report the experiments conducted to evaluate our approach.

4.4.1 Data and Settings

We used the review dataset developed by Maas et al. (2011) as the development dataset to compute the co-occurrences of word pairs. This dataset contains 50k movie reviews and 90k vocabulary. We consider a window of 10 words to compute co-occurrences.

We also employed the standard TASA corpus to compute the semantic similarity of word pairs for LSA. This corpus contains around 61K documents and 155K vocabulary. We believe that LSA with TASA produces better performance than our development dataset. This is because our corpus is smaller than TASA and it contains user generated text which is known to be grammatically week with many spelling errors and slangs. However, TASA is adapted from 6,333 textbooks and does not have the above issues.

For the evaluation purpose, we used two datasets: the MPQA (Wilson, Wiebe, and Hoffmann, 2005) and IQAPs (de Marneffe, Manning, and Potts, 2010) datasets. The MPQA dataset is used for SO prediction experiments, while the IQAP dataset is used for the IQAP experiments. For MPQA dataset, we ignore the neutral words and use the remaining 4000 opinion words with their sentiment orientations. The IQAPs dataset contains a 125 IQAPs and their corresponding *yes* or *no* labels as the ground truth as described in (de Marneffe, Manning, and Potts, 2010).

4.4.2 Experimental Results

4.4.2.1 IQAP Inference Evaluation

Table 4.6 shows the evaluation results for the task IQAPs. The first row presents the result obtained by the approach proposed by de Marneffe, Manning, and Potts (2010). This is our baseline and obtained an accuracy of 60% on the IQAP dataset. As explained in Chapter 2, their decision procedure is based on the individual sentiment orientation of the adjectives in question and its corresponding answer and does not consider the correlation between the two adjectives. However, our approach is able to directly infer *yes* or *no* responses using sentiment similarity between the adjectives and does not require computing sentiment orientation.

The second and third rows of Table 4.6 show the results of using

Method	Precision	Recall	F-Measure
Marneffe et al. (2010)	60.00	60.00	60.00
PMI	60.61	58.70	59.64
LSA	66.70	54.95	60.26
SS (w/o WSD)	75.03	77.85	76.41
SS (with WSD)	**76.69**	**79.75**	**78.19**

Table 4.6: Experimental results on IQAP inference task using sense sentiment similarity with and without WSD, and their comparison with semantic similarity measures and the state-of-the-art approach

PMI and LSA as the sentiment similarity (SS) measures in the algorithm explained in Table 4.4. The last rows, *SS (with WSD)* and *SS (w/o WSD)* indicate the results when we use our sentiment similarity measures with and without WSD respectively. *SS (w/o WSD)* is based on the first sense (most common sense) of the words, whereas *SS (with WSD)* utilizes the real sense of the words. We manually annotate the sense of the adjectives to investigate the importance of WSD in a perfect setting. The results show that they significantly improve the performance of the best performing baseline (LSA) by 16.15% and 17.93% F1 improvements. Furthermore, as it is clear in Table 4.6, using correct sense of the adjectives increases the performance from 76.41% to 78.19%. However, this difference is not significant because only 14% of the adjectives are assigned senses different from their first senses. The efficiency of the WSD would have been more prominent, if more IQAPs contain adjectives with senses different from their first senses.

4.4.2.2 Evaluation of Sentiment Orientation Prediction

Table 4.7 shows the results of word sentiment prediction. The results in the table are based on the algorithm in Table 4.5 where PMI, LSA and SS (our method) are used for calculating the similarity between two words respectively. As it is shown, LSA significantly outperforms PMI. It was expected since PMI is known as a contextual similarity measure which is based on co-occurrence of word pairs. Furthermore, our development

Method	Precision	Recall	F-Measure
SO-PMI	56.20	56.36	55.01
SO-LSA	66.31	66.89	66.26
SO-SS	**73.07**	**73.89**	**73.11**

Table 4.7: Experimental results on SO prediction task using sense sentiment similarity and its comparison with semantic similarity measures

dataset is relatively small and this leads to poor co-occurrence information. The SS method utilizes the first sense of the words here and significantly outperforms the two baselines. It outperforms PMI and LSA by 18.1% and 6.85% respectively. The SS method, in contrast to PMI, does not require big development dataset to perform well.

4.5 Analysis and Discussion

In this section, we explore the role of using singular value decomposition (SVD) and different emotional categories. In addition, we study the effect of synsets and antonyms of words for predicting their sentiment similarity. We investigate these factors on the sentiment prediction task.

Role of using SVD: To study the role of SVD, we construct an *emotional matrix* using the emotional vectors of words and their antonyms with respect to their senses.

$$\begin{array}{l|cccc|}
 & anger & disgust & \dots & courage \\
w_1, sense(w_1) & I_1 & I_2 & \dots & I_{12} \\
w_2, sense(w_2) & I_1^{'} & I_2^{'} & \dots & I_{12}^{'} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
w_i, sense(w_i) & I_1^{''} & I_2^{''} & \dots & I_{12}^{''}
\end{array}$$

Our *SS* measure works based on the co-occurrence between words and emotional categories. Thus, some inappropriate words may add some noise to the vectors and emotional matrix. Running SVD allows us to collapse the matrix into a smaller dimensional space where highly correlated items are captured as a single feature. In other words, it makes the

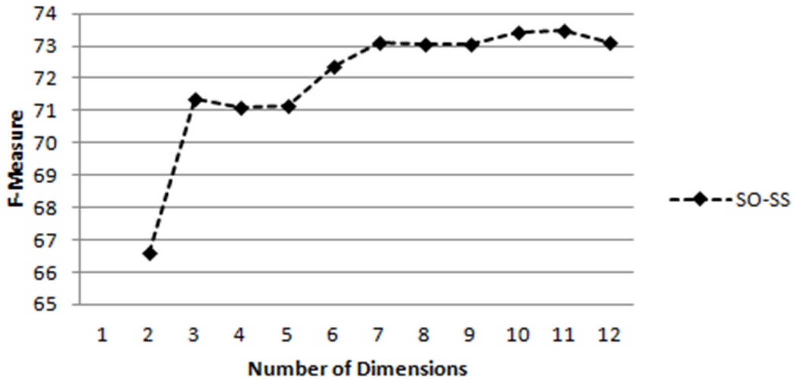


Figure 4.2: Dimensions reduction; this figure shows the experimental results on the sentiment prediction task using SVD with different dimensional reductions. The experiment using 12 emotions means it has done without dimensional reduction

best possible reconstruction of the matrix with the least possible information and can potentially reduce the noise coming from the co-occurrence information. It can also emphasize the strong patterns and trends.

We repeat the experiments on the sentiment prediction task using SVD with different dimensional reductions. Figure 4.2 shows the results. As it is shows, higher performances can be achieved with greater dimensions. The highest performance occurs in the dimension 11 which is 73.50%. The results also show that the dimensions lower than three results in great reductions in the performance, whereas there are no big performance reductions in the greater dimensions. We believe this is because of the use of synsets that can highly resist against the co-occurrence noise in the data.

Role of emotional categories: As explained in Section 4.2.1, we construct emotional categories from hierarchical synonyms of the basic emotions (we referred to them as seeds). Here, we repeat the experiments on the sentiment prediction task by three sets of emotional categories to illustrate the importance of the two constraints explained in Equation 4.1.

Figure 4.3 shows that if we use "*all hierarchical synonyms*" as seeds, the performance of sentiment prediction is poor. The reason is that some irrelevant seeds may enter into the emotional categories solely due to their distance in the hierarchical synonyms.

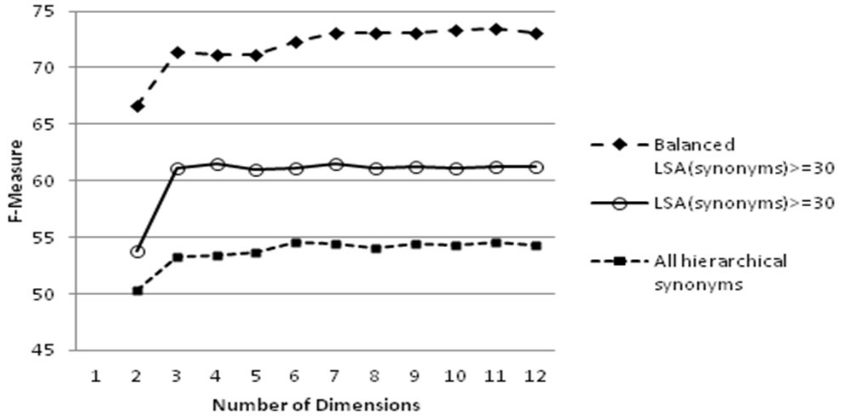


Figure 4.3: Selection of emotional categories; this figure shows the experimental results on the sentiment prediction task using different sets of emotional categories

To only utilize the relevant seeds of each emotional category, we considered the first constraint of Equation 4.1 which is that the selected seeds should be semantically close to the basic emotions. Therefore, we construct the second emotional set employing the hierarchical synonyms which have high semantic similarities (LSA) with the basic emotions, i.e. "$LSA(synonyms) \geq 30$". Here we set 30 as the threshold. This is because we aim to keep a sufficient number of seeds in each category and at the same time preserve the semantic similarities between seeds and their corresponding emotion categories. As Figure 4.3 shows, this constraint improves the performance of sentiment prediction over all the dimensions.

The emotional vectors may also being biased toward the category that has the highest number of occurrences of seeds in the development corpus. Thus, the second constraint of Equation 4.1 requires the categories to be balanced with respect to their seeds occurrences (frequencies) in the development corpus. We balanced the second set in such a way that the sum of the frequencies of all the seeds in each category remains the same among the categories (balanced matrix). We also manually removed a few ambiguous or irrelevant seeds from each category. For example, the emotional category "*interest*" has mainly two sets of synonyms related to *interestingness* and *finance*; however we only consider the *interestingness*

Strategies	**Precision**	**Recall**	**F-Measure**
w/o Antonyms and Synsets	67.79	68.47	67.57
with Synsets	71.47	72.25	71.43
with Antonyms	68.34	69.04	68.12
with Antonyms and Synsets	**73.07**	**73.89**	**73.11**

Table 4.8: Role of using synsets and antonyms; Experimental results on SO prediction task using sense semantic similarity without using synsets or antonyms

set as it reflects the target sentiment. As Figure 4.3 shows, using two constraints results in the best performance in any dimension. This experiment indicates that an accurate result can be obtained, if only relevant seeds that results in a balanced matrix are selected.

Role of using synsets and antonyms of words: We show the important role of antonyms and synsets of words which we explained in Section 4.2.2. For this purpose, we repeat the experiment for SO prediction by computing sentiment similarity of word pairs without using the synonyms and antonyms.

Table 4.8 shows the results. As it is clear, the highest performance can be achieved when antonyms and synonyms are used, while the lowest performance is obtained without using them. Table 4.8 also shows that using only synsets is more effective than using only antonyms. This could be because of the higher probability of the existence of synonyms than antonyms for a word.

4.6 Summary

In this chapter, we propose an effective method to compute sentiment similarity from a connection between semantic space and emotional space. We show the effectiveness of our method in two NLP tasks namely, indirect question-answer pair inference and sentiment orientation prediction. Our experiments show that sentiment similarity measure is an essential prerequisite to obtain reasonable performances in the above tasks. We show that sentiment similarity significantly outperforms two popular semantic similarity measures, namely, PMI and LSA.

The work in this chapter has been presented in the 26th Conference on Artificial Intelligence, AAAI 2012 (Mohtarami et al., 2012).

Chapter 5

Probabilistic Sense Sentiment Similarity through Hidden Emotions

Sentiment Similarity of word pairs reflects the distance between the words regarding their underlying sentiments. Similar to Chapter 4, this chapter aims to infer the sentiment similarity between word pairs with respect to their senses. To achieve this aim, in contrast to Chapter 4 which proposed the use of a fixed set of basic emotions, we now propose a probabilistic emotion-based approach that is built on the hidden emotional models in which the number and types of the basic emotions are considered as unknown. The hidden emotional models aim to predict a vector of hidden emotions for each sense of the words. The resultant hidden emotional vectors are then employed to infer the sentiment similarity of word pairs. We apply the proposed approach to address two main NLP tasks, namely, *Indirect yes/no Question Answer Pairs* inference and *Sentiment Orientation* prediction. Extensive experiments demonstrate the effectiveness of the proposed approach.

Word	**Emotional Vector**	**SO**
e = [*anger, disgust, sadness, fear, guilt, interest, joy, shame, surprise*]		
Rude	['0.2','0.4',0,0,0,0,0,0,0]	-0.6
doleful	[0, 0,'0.4',0,0,0,0,0,0]	-0.4
smashed	[0,0,'0.8','0.6',0,0,0,0,0]	-1.4
shamefully	[0,0,0,0,0,0,0,'0.7',0]	-0.7
deceive	[0,'0.4','0.5',0,0,0,0,0,0]	-0.9

Table 5.1: Sample of emotional vectors with respect to the following set of emotions: e = [*anger, disgust, sadness, fear, guilt, interest, joy, shame, surprise*]

5.1 Motivation and Problem Definition

This chapter attempts to predict sense sentiment similarity that aims to infer the similarity between word pairs with respect to their senses and underlying sentiments through hidden emotions.

As we discussed in Chapter 4, existing works employed semantic similarity measures to estimate sentiment similarity of word pairs (Kim and Hovy, 2004; Turney and Littman, 2003). However, it has been shown that although the semantic similarity measures are good for relating semantically related words like "*car*" and "*automobile*" (Islam and Inkpen, 2008), they are less effective in capturing sentiment similarity. For example, using Latent Semantic Analysis (Landauer, Foltz, and Laham, 1998), the semantic similarity of "*excellent*" and "*good*" is greater than the similarity between "*excellent*" and "*superior*". However, the intensity of sentiment in "*excellent*" is more similar to "*superior*" than "*good*". That is, sentiment similarity of "*excellent*" and "*superior*" should be greater than "*excellent*" and "*good*".

As we discussed above, semantic similarity measures are less effective in inferring sentiment similarity between word pairs. In addition, considering just the total sentiment of words (as positive or negative) is also not sufficient to accurately infer sentiment similarity between word senses. The reason is that, although the opinion words can be categorized into *positive* and *negative* sentiments with different sentiment intensity values, they carry different human emotions. In fact, a sentiment word can be represented as a vector of emotions with *intensity* values from "*very weak*" to "*very strong*". For example, Table 5.1 shows several sentiment words and their corresponding emotion vectors based the following set of emotions: e = [*anger, disgust, sadness, fear, guilt, interest, joy, shame, surprise*]. Given the above emotions, "*deceive*" has 0.4 and 0.5 intensity values with respect to the emotions "*disgust*" and "*sadness*" with an overall -0.9 (i.e. -0.4-0.5) value for sentiment orientation (Neviarouskaya, Prendinger, and Ishizuka, 2007; Neviarouskaya, Prendinger, and Ishizuka, 2009).

The difficulty of the sentiment similarity prediction task is evident when terms carry different types of emotions. For instance, all the words in Table 5.1 have negative sentiment orientation, but, they carry different emotions with different emotional vectors. For example, "*rude*" reflects the emotions "*anger*" and "*disgust*", while the word "*doleful*" only reflects the emotion "*sadness*". As such, the word "*doleful*" is closer to the words "*smashed*" and "*deceive*" involving the emotion "*sadness*" than others.

Using only semantic similarity measures or considering the overall sentiment orientation of words are not suitable to infer sentiment similarity of words. This chapter shows that hidden emotional vectors of the words can be effectively utilized to predict the sentiment similarity between them.

To achieve the aims of this chapter, we propose a probabilistic approach employing the hidden emotional model in which the semantic and emotional spaces are combined to predict the hidden emotional vectors of the words. These emotional vectors are then employed to infer Probabilistic Sense Sentiment Similarity (PSSS) between the words. Furthermore, we show that PSSS can be effectively utilized to address *Indirect yes/no Question Answer Pairs* (IQAPs) *Inference* and *Sentiment Orientation* (SO) *prediction* tasks.

In IQAPs, the answer of the question does not explicitly contain a clear *yes* or *no*, but rather gives information to infer such an answer.

That is, the IQAPs inference task aims to interpret the information in the answer of a given IQAP and infer the *yes* or *no* response. The second task (SO prediction) aims to determine the sentiment orientation of individual words. A more detailed information on these tasks have been presented in Section 4.1.

In summary, the contributions of this chapter are follows:

- We propose an effective approach to predict the sentiment similarity between word pairs through hidden emotions at the sense level,
- We show that the sentiment similarity computed using emotional vectors is more accurate than using the SO of the words,
- We show that such sentiment similarity can be utilized to get accurate SO for each sense of the words, and
- Our hidden emotional model can infer the types and number of hidden emotions in a corpus.

5.2 Sentiment Similarity through Hidden Emotions

Previous research showed that there exists a small set of basic (or fundamental) emotions which are central to other emotions (Ortony and Turner, 1990; Izard, 1971). based on previous research, in Chapter 4, we employed twelve basic emotions that are central and generally accepted: *anger, disgust, fear, guilt, sadness, shame, interest, joy, surprise, desire, love, courage.* However, in previous research, there is little agreement about the number and types of basic emotions. Thus, we will now assume that the number and types of basic emotions are hidden and not pre-defined and propose two emotional models to extract the hidden emotions of word senses to infer their sentiment similarity.

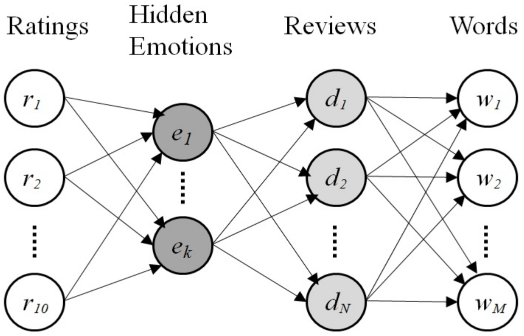
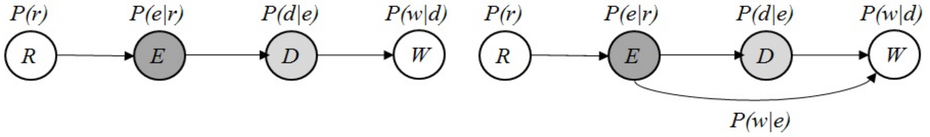


Figure 5.1: The structure of Probabilistic Sense Sentiment Similarity (PSSS)



(a) Series model

(b) Bridged model

Figure 5.2: Hidden emotional model

5.2.1 Hidden Emotional Model

Online review portals provide rating mechanisms (in terms of stars, e.g. 5- or 10-star rating) to allow users to attach ratings to their reviews. A rating indicates the summarized opinion of a user who ranks a product or service based on his feelings. Though positive or negative ratings are assigned to the reviews, there are various feelings and emotions behind such ratings with respect to the content of the reviews.

Figure 5.1 shows the intermediate layer of hidden emotions behind the ratings (sentiments) assigned to the documents (reviews) containing the words. This figure indicates the general structure of our Probabilistic Sense Sentiment Similarity (PSSS) model. It shows that hidden emotions (e_i) link the rating (r_j) and the documents (d_k). In this section, we aim to employ ratings and the relations among ratings, documents, and words to extract the hidden emotions.

Figure 5.2 illustrates a simple graphical model of Figure 5.1. As Figures 5.2 shows, the rating r from a set of ratings $R = r_1, ..., r_p$ is assigned to a hidden emotion set $E = e_1, ..., e_k$. A document d from a set of documents $D = d_1, ..., d_N$ with vocabulary set $W = w_1, ..., w_M$ is associated with the hidden emotion set.

Considering Figure 5.1, we represent the entire text collection as a set of (w, d, r) in which each observation (w, d, r) is associated with a set of unobserved emotions. If we assume that the observed tuples are independently generated, the whole data set is generated based on the joint probability of the observation tuples (w, d, r) as the follows:

$$D = \prod_r \prod_d \prod_w P(w, d, r)^{n(w,d,r)} = \prod_r \prod_d \prod_w P(w, d, r)^{n(w,d)n(d,r)} \tag{5.1}$$

where, $P(w, d, r)$ is the joint probability of the tuple (w, d, r), and $n(w, d, r)$ is the frequency of w in document d of rating r (note that $n(w, d)$ is the term frequency of w in d and $n(d, r)$ is one if r is assigned to d, and 0 otherwise).

There are two ways to infer the joint probability $P(w, d, r)$ with respect to the Figure 5.2(a) and 5.2(b) in which the lines indicates the dependency between the elements r, e, d, and w.

The first model, Figure 5.2(a), assumes that *the word w is dependent on d and independent of e* (we refer this assumption as *A1*). We call the model constructed based on this assumption as *Series Hidden Emotional Model* (SHEM). The joint probability for this model is defined as follows considering the hidden emotion e:

- Regarding class probability of the hidden emotion e to be assigned to the observation (w, d, r):

$$P(w, d, r) = \sum_e P(w, d, r|e)P(e) = \sum_e P(w, d|e)P(r|e)P(e)$$

- Regarding the assumption *A1* and Bayes' Rule:

$$= \sum_e P(w|d, e)P(d, e)P(r|e) = \sum_e P(w|d)P(d|e)P(e)P(r|e)$$

$$= P(w|d) \sum_{e} P(d|e)P(e)P(r|e) \tag{5.2}$$

In reality, a word w can inherit properties (e.g., emotions) from the document d that contains w. Thus, we can assume that w is implicitly dependant on e. To account for this, we present the second emotional model which is called *Bridged Hidden Emotional Model* (BHEM) and shown in Figure 5.2(b). Our assumption, *A2*, in the BHEM model is as follows: *w is dependent on both d and e.* The joint probability for this model is defined as follows considering hidden emotion e:

- Regarding class probability of the hidden emotion e to be assigned to the observation (w, d, r):

$$P(w, d, r) = \sum_{e} P(w, d, r|e)P(e) = \sum_{e} P(w, d|e)P(r|e)P(e)$$

- Regarding the assumption *A2* and Bayes' Rule:

$$= \sum_{e} P(w|d, e)P(d, e)P(r|e) = \sum_{e} P(d, e|w)P(w)P(r|e)$$

- Regarding the assumption *A2* and conditional independency:

$$= \sum_{e} P(d|w)P(e|w)P(w)P(r|e)$$

$$= P(d|w) \sum_{e} P(w|e)P(e)P(r|e) \tag{5.3}$$

In the bridged model, the joint probability does not depend on the probability $P(d|e)$ and the probabilities $P(w|e)$, $P(e)$ and $P(r|e)$ are unknown, while in the SHEM model, the joint probability does not depend on $P(w|e)$, and probabilities $P(d|e)$, $P(e)$, and $P(r|e)$ are unknown.

We employ Maximum Likelihood to learn the unknown probabilities and infer the possible hidden emotions. The log-likelihood of the whole data set D of Equation 5.1 can be defined as:

$$L = \sum_{r} \sum_{d} \sum_{w} n(w, d)n(d, r) \log P(w, d, r) \tag{5.4}$$

Replacing $P(w, d, r)$ by the values computed using series and bridged models in Equations 5.2 and 5.3 results in:

$$L_1 = \sum_r \sum_d \sum_w n(w, d)n(d, r) \log\left[P(w|d) \sum_e P(d|e)P(e)P(r|e)\right] \quad (5.5)$$

$$L_2 = \sum_r \sum_d \sum_w n(w, d)n(d, r) \log\left[P(d|w) \sum_e P(w|e)P(e)P(r|e)\right] \quad (5.6)$$

The above optimization problems are hard to compute due to the log of sum. Thus, *Expectation-maximization* (EM) is usually employed. EM consists of two following steps:

1. E-step: Calculate expectation (posterior probabilities) for hidden variables given the observations by using the current estimates of the parameters, and

2. M-step: Update parameters such that the data log-likelihood (log L) increases using the posterior probabilities in the E-step.

The steps of EM can be computed regarding SHEM and BHEM models. First, we derive the EM equations for SHEM by utilizing the assumption *A1* and Bayes Rule as follows:

E-step:

$$P(e|w, d, r) = \frac{P(r|e)P(e)P(d|e)}{\sum_e P(r|e)P(e)P(d|e)} \quad (5.7)$$

M-step:

$$P(r|e) = \frac{\sum_d \sum_w n(w, d)n(d, r)P(e|w, d, r)}{\sum_r \sum_d \sum_w n(w, d)n(d, r)P(e|w, d, r)} \quad (5.8)$$

$$P(d|e) = \frac{\sum_r \sum_w n(w, d)n(d, r)P(e|w, d, r)}{\sum_d \sum_r \sum_w n(w, d)n(d, r)P(e|w, d, r)} \quad (5.9)$$

$$P(e) = \frac{\sum_r \sum_d \sum_w n(w, d)n(d, r)P(e|w, d, r)}{\sum_e \sum_d \sum_r \sum_w n(w, d)n(d, r)P(e|w, d, r)} \quad (5.10)$$

Second, EM of BHEM employs assumptions *A2* and Bayes Rule and is defined as follows:

E-step:

$$P(e|w,d,r) = \frac{P(r|e)P(e)P(w|e)}{\sum_e P(r|e)P(e)P(w|e)} \tag{5.11}$$

M-step:

$$\begin{aligned} P(r|e) &= \frac{\sum_d \sum_w n(w,d)n(d,r)P(e|w,d,r)}{\sum_r \sum_d \sum_w n(w,d)n(d,r)P(e|w,d,r)} \\ &= \frac{\sum_w n(w,r)P(e|w,d,r)}{\sum_r \sum_w n(w,r)P(e|w,d,r)} \end{aligned} \tag{5.12}$$

$$\begin{aligned} P(w|e) &= \frac{\sum_r \sum_d n(w,d)n(d,r)P(e|w,d,r)}{\sum_w \sum_r \sum_d n(w,d)n(d,r)P(e|w,d,r)} \\ &= \frac{\sum_r n(w,r)P(e|w,d,r)}{\sum_w \sum_r n(w,r)P(e|w,d,r)} \end{aligned} \tag{5.13}$$

$$\begin{aligned} P(e) &= \frac{\sum_r \sum_d \sum_w n(w,d)n(d,r)P(e|w,d,r)}{\sum_e \sum_d \sum_r \sum_w n(w,d)n(d,r)P(e|w,d,r)} \\ &= \frac{\sum_r \sum_w n(w,r)P(e|w,d,r)}{\sum_e \sum_r \sum_w n(w,r)P(e|w,d,r)} \end{aligned} \tag{5.14}$$

Note that in Equation 5.11, the probability $P(e|w,d,r)$ does not depend on the document d. Also, in Equations 5.12-5.14 we remove the dependency on document d using the following Equation:

$$\sum_d n(w,d)n(d,r) = n(w,r) \tag{5.15}$$

where $n(w,r)$ is the occurrence of w in all the documents in the rating r.

The EM steps computed by the bridged model do not depend on the variable document d, and discard d from the model. The reason is that w bypasses d to directly associate with the hidden emotion e in Figure 5.2(b).

Finally, we construct the emotional vectors using the algorithm presented in Table 5.2. The algorithm uses document-rating, term-document

```
Inputs:
  Series Model: Document-Rating $D \times R$, Term-Document $W \times D$
  Bridged Model: Term-Rating $W \times R$

Output:
Emotional vectors $\{e_1, e_2, ..., e_k\}$ for $w$

Algorithm:

1. Enriching hidden emotional model:
     Series Model: Update Term-Document $W \times D$
     Bridged Model: Update Term-Rating $W \times R$

2. Initialize unknown probabilities:
     Series Model: Initialize $P(d|e)$, $P(r|e)$, and $P(e)$, randomly
     Bridged Model: Initialize $P(w|e)$, $P(r|e)$, and $P(e)$

3. while $L$ has not converged to a pre-specified value do

4. E-step;
     Series Model: estimate the value of $P(e|w,d,r)$ in Equation 5.7
     Bridged Model: estimate the value of $P(e|w,d,r)$ in Equation 5.11

5. M-step;
     Series Model: estimate the values of $P(r|e)$, $P(d|e)$, and $P(e)$ in Equations 5.8-5.10, respectively
     Bridged Model: estimate the values of $P(r|e)$, $P(w|e)$, and $P(e)$ in Equations 5.12-5.14, respectively

6. end while

7. If series hidden emotional model is used then

8. Infer word emotional vector: estimate $P(w|e)$ in Equation 5.16.

9. End if
```

Table 5.2: Algorithm to Construct emotional vectors via $P(w|e)$

and term-rating matrices to infer the unknown probabilities. This algorithm can be used with both bridged or series models. Our goal is to infer the emotional vector for each word w that can be obtained by the probability $P(w|e)$. Note that, this probability can be simply computed for the SHEM model using $P(d|e)$ as follows:

$$P(w|e) = \sum_{d} P(w|d)P(d|e) \tag{5.16}$$

5.2.1.1 Enriching Hidden Emotional Models

In our hidden model, the term-document, document-rating and term-rating matrices are employed as inputs to infer the emotional vectors. The matrices just present the knowledge about the frequency of a word in documents or documents in ratings.

Suppose we have prior information about the semantic similarity between some words before using the hidden model. For example, there are two words w_1 and w_2 in the matrices that are synonyms (thus their emotional vectors should be similar). The question is how this knowledge can be transferred to our model. One simple way is using some post-processing after getting the emotional vectors of w_1 and w_2, e.g. by averaging their emotional vectors. However, this approach is less effective, since the knowledge about the synonyms w_1 and w_2 has not yet been transferred to the hidden model and this knowledge has not been employed in the learning step of the model. To utilize the word similarity knowledge, we use the following enriched matrix in which each cell shows the semantic relation between the two words in the corresponding row and column. If we do not have any knowledge about two words or they are not sentimentally co-related, their corresponding cell will be zero. To compute the semantic similarity between each two words, we utilize the synset of the words as follows:

$$w_iw_j = P(syn(w_i)|syn(w_j)) = \frac{1}{|syn(w_i)|} \sum_{i}^{|syn(w_i)|} \frac{1}{|syn(w_j)|} \sum_{j}^{|syn(w_j)|} P(w_i|w_j) \tag{5.17}$$

where, $syn(w)$ is the synset of word w. Let $count(w_i, w_j)$ be the co-occurrence of the words w_i and w_j, and let $count(w_j)$ be the total word count. The probability of w_i given w_j will then be as follows: $P(w_i|w_j) = count(w_i, w_j)/count(w_j)$.

The reason we employ the co-occurrence of the words is as follows. First, we employ the hypothesis that a word can be characterized by its neighbors (Turney and Littman, 2003). That is, the emotional vector of a word tends to correspond to the emotional vectors of its neighbors. Second, each entry of the input matrices of our hidden model is based on the frequency of a word in the whole length of a document or rating. However, this scale is large and may add some noise to our hidden model. The co-occurrence of words in a small window can make our model more accurate.

In addition, the reason we employ the synset of the words is as follows. First, as the synset of a word has the same or nearly the same meaning as the original word, the word can be replaced by any of its synset with no major changes in its emotion. Second, the major advantage of using synset is that we can obtain different emotional vectors for each sense of a word and predict the sentiment similarity at the sense level. Note that, various senses of a word can have diverse meanings and emotions, and consequently different emotional vectors. If two words w_i and w_j are synonyms, their corresponding entry in the enriched matrix will be one.

To improve our hidden model, the enriched matrix $W \times W$ is multiplied to the inputs of the model $W \times D$ or $W \times R$ such that the sense of words can be added to the matrices. The learning step of EM is done using the updated inputs. In this case, the correlated words can inherit the properties of each other. For example, if w_i does not occur in a document or rating involving another word (i.e., w_j), the word w_i can be indirectly

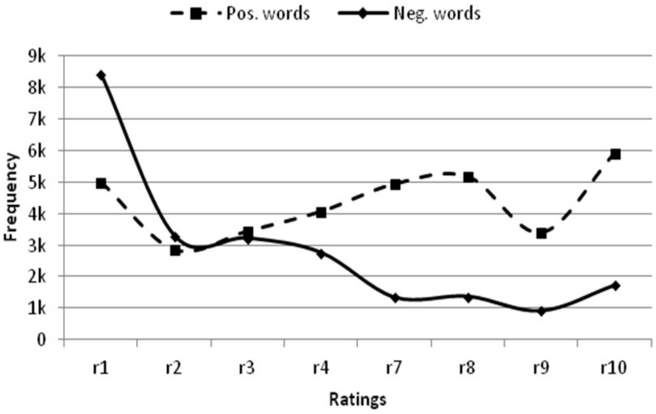


Figure 5.3: Nonuniform distribution of opinion words through ratings. Here, r1-r4 and r7-r10 are respectively negative and positive ratings. We exclude the ratings 5 and 6 that are more neutral

associated to the document through the word w_j. However, the distribution of the opinion words in documents and ratings is not uniform. This may decrease the effectiveness of the enriched matrix.

The nonuniform distribution of opinion words has been also reported by Amiri and Chua (2012) who showed that positive words are frequently used in negative reviews. We also observed the same pattern in the development dataset. Figure 5.3 shows the overall occurrence of some positive and negative seeds in various ratings. As shown, in spite of the negative words, the positive words may frequently occur in both positive and negative documents. Such distribution of positive words can mislead the enriched model.

To address this issue, we measure the confidence of an opinion word in the enriched matrix as follows.

$$Confidence_w = \frac{ABS[(TF_w^- \times DF_w^-) - (TF_w^+ \times DF_w^+)]}{(TF_w^- \times DF_w^-) + (TF_w^+ \times DF_w^+)} \tag{5.18}$$

where, $TF_w^-(TF_w^+)$ is the frequency of w in the ratings 1 to 4 (7 to 10), and $DF_w^-(DF_w^+)$ is the total number of documents with rating 1 to 4 (7 to 10) that contain w. The confidence value of w varies from 0 to 1, and it increases if:

- There is a large difference between the occurrences of w in positive and negative ratings.

- There is a large number of reviews involving w in the relative ratings.

To improve the efficiency of enriched matrix, the columns corresponding to each word in the matrix are multiplied by its confidence value.

5.2.2 Predicting Sentiment Similarity

So far, we computed the emotional vectors of the words with respect to their senses using the proposed series hidden emotional model. To infer the sentiment similarity of words, we compare each emotion of a word with corresponding emotion of another. To achieve this aim, we use the correlation coefficient between the emotional vectors of two words to compute the sentiment similarity between them regarding their senses. Let X and Y be the emotional vectors of two words. Equation 5.19 computes their correlation:

$$corr(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)S_X S_Y} \tag{5.19}$$

where, n is number of emotional categories, $\bar{X}, \bar{Y}$ and S_X, S_Y are the mean and standard deviation values of X and Y respectively.

The problem is that how large the correlation value should be to consider two words as similar in sentiment. We address this issue by utilizing the antonyms of the words as explained in Chapter 3. Since the word and its antonyms have opposite sentiment orientation, we consider two words, w_i and w_j as similar in sentiment iff they satisfy both of the following conditions:

1. $corr(w_i, w_j) > corr(w_i, \sim w_j), and$
2. $corr(w_i, w_j) > corr(\sim w_i, w_j)$

where, $\sim w_i(\sim w_j)$ are antonyms of $w_i(w_j)$ respectively, and $corr(w_i, w_j)$ is the correlation between the emotional vectors obtained from Equation 5.19.

Inputs:
SAQ: The adjective in the question of given IQAP.
SAA: The adjective in the answer of given IQAP.

Output:
answer $\in \{yes, no, uncertain\}$

Algorithm:

```
if $SAQ$ or $SAA$ are missing from our corpus then
    answer = $Uncertain$;
else if $PSSS(SAQ, SAA) < 0$ then
        answer = $No$;
    else if $PSSS(SAQ, SAA) > 0$ then
        answer = $yes$;
```

Table 5.3: Decision procedure of employing Probabilistic Sense Sentiment Similarity (PSSS) to address IQAP inference task

Finally, we compute the probabilistic sense sentiment similarity ($PSSS$) between two words as follows:

$$PSSS(w_i, w_j) = corr(w_i, w_j) - Max\{corr(w_i, \sim w_j), corr(\sim w_i, w_j)\} \tag{5.20}$$

A positive value of $PSSS(.,.)$ indicates that the words are sentimentally similar and negative value shows the amount of dissimilarity between the words.

5.3 Applications

We explain our approach in utilizing sentiment similarity between words to perform IQAP inference and SO prediction tasks respectively.

In IQAPs, we employ the sentiment similarity between the adjectives in questions and answers to interpret the indirect answers. For easy reading, we reproduce the algorithm in Table 4.4 as Table 5.3 for this purpose. $PSSS(.,.)$ indicates probabilistic sense sentiment similarity computed by Equation 5.20. A positive $PSSS$ means the words are sentimentally similar and thus the answer is yes. However, negative $PSSS$ leads to a no response.

In SO-prediction task, we attempt to show that sentiment similarity

Inputs:
Pwords: seven words with positive SO
Nwords: seven words with negative SO
$A(.,.)$: similarity function, and w: a given word with unknown SO

Output:
P: sentiment orientation of w

Algorithm:
1. $P = SO\_A(w) =$

$$\sum_{pword \in Pwords} A(w, pword) - \sum_{nword \in Nwords} A(w, nword)$$

Table 5.4: SO based on the similarity function $A(.,.)$

along with a simple algorithm is able to accurately predict sentiment orientation (SO). To achieve this aim, sentiment similarity is computed from Equation 5.20 and the algorithm presented by Turney and Littman (2003) is used which we show again in Table 5.4 for easy reading. Just as in Table 4.5, the similarity function $A(.,.)$ in Table 5.4 is implemented using our $PSSS(.,.)$ instead of PMI employed by Turney and Littman (2003).

5.4 Evaluation and Results

5.4.1 Data and Settings

We used the review dataset employed by Maas et al. (2011) as the development dataset that contains movie reviews with star rating from one star (most negative) to 10 stars (most positive). We exclude the ratings 5 and 6 that are more neutral. We used this dataset to compute all the input matrices in Table 5.2 as well as the enriched matrix. The development dataset contains 50k movie reviews and 90k vocabulary.

We also used two datasets for the evaluation purpose: the MPQA (Wilson, Wiebe, and Hoffmann, 2005) and IQAPs (de Marneffe, Manning, and Potts, 2010) datasets. The MPQA dataset is used for SO prediction

Method	**Precision**	**Recall**	**F-Measure**
PMI	56.20	56.36	55.01
ER	65.68	65.68	63.27
PSSS-SHEM	68.51	69.19	67.96
PSSS-BHEM	**69.39**	**70.07**	**68.68**

Table 5.5: Experimental results on SO prediction task using series and bridged hidden emotional models, and their comparison with the other approaches

experiments, while the IQAP dataset is used for the IQAP experiments. We ignored the neutral words in MPQA dataset and used the remaining 4k opinion words. Also, the IQAPs dataset (de Marneffe, Manning, and Potts, 2010) contains 125 IQAPs and their corresponding *yes* or *no* labels as the ground truth.

5.4.2 Experimental Results

To evaluate our PSSS model, we perform experiments on the SO prediction and IQAPs inference tasks. Here, we consider six emotions for both bridged and series models. We will study the effect of emotion numbers in Section 5.5.1. Also, we set a threshold of 0.3 for the confidence value in Equation 5.18, i.e. we set the confidence values smaller than the threshold to 0. We explain the effect of this parameter in Section 5.5.3.

5.4.2.1 Evaluation of SO Prediction

We evaluate the performance of our PSSS models in the SO prediction task using the algorithm explained in Table 5.4 by setting our PSSS as the similarity function (A). The results on SO prediction are presented in Table 5.5. The first and second rows present the results of our baselines, PMI (Turney and Littman, 2003) and Expected Rating (ER) (Potts, 2011) of words respectively.

PMI extracts the semantic similarity between words using their co-occurrences. As Table 5.5 shows, it leads to poor performance. This is mainly due to the relatively small size of the development dataset which affects the quality of the co-occurrence information used by the PMI.

ER computes the expected rating of a word based on the distribution of the word across rating categories. The value of ER indicates the SO of the word. As shown in the two last rows of the table, the results of PSSS approach are higher than PMI and ER. The reason is that PSSS is based on the combination between sentiment space (through using ratings, and matrices $W \times R$ in BHEM, $D \times R$ in SHEM) and semantic space (through the input $W \times D$ in SHEM and enriched matrix $W \times W$ in both hidden models). However, the PMI employs only the semantic space (i.e., the co-occurrence of the words) and ER uses occurrence of the words in rating categories.

Furthermore, the PSSS model achieves higher performance with BHEM rather than SHEM. This is because the emotional vectors of the words are directly computed from the EM steps of BHEM. However, the emotional vectors of SHEM are computed after finishing the EM steps using Equation 5.16. This causes the SHEM model to estimate the number and type of the hidden emotions with a lower performance as compared to BHEM, although the performances of SHEM and BHEM are comparable as will be explained in Section 5.5.1.

5.4.2.2 Evaluation of IQAPs Inference

To apply our PSSS on IQAPs inference task, we use it as the sentiment similarity measure in the algorithm explained in Table 5.3. The results are presented in Table 5.6. The first and second rows are baselines. The first row is the result obtained by de Marneffe, Manning, and Potts (2010) approach. They computed SO of the adjectives based on the expected ratings (ER), and then employed the SO to infer *yes* or *no.* However, our experiments show that the approach based on sentiment similarity constructed using emotional vectors is more accurate than only comparing the SOs to infer indirect answers.

Method	**Precision**	**Recall**	**F-Measure**
Marneffe et al. (2010)	60.00	60.00	60.00
PMI	60.61	58.70	59.64
PSSS-SHEM	62.55	61.75	61.71
PSSS-BHEM (w/o WSD)	65.90	66.11	63.74
SS-BHEM (with WSD)	**66.95**	**67.15**	**65.66**

Table 5.6: Experimental results on IQAP inference task using series and bridged hidden emotional models, and their comparison with the other approaches

The second row of Table 5.6 shows the results of using a popular semantic similarity measure, PMI, as the sentiment similarity (SS) measure in Table 5.3. The result shows that PMI is less effective in capturing the sentiment similarity.

Our PSSS approach directly infers *yes* or *no* responses using SS between the adjectives and does not require computing SO of the adjectives. In Table 5.6, *PSSS-SHEM* and *PSSS-BHEM* indicate the results when we use our PSSS with SHEM and BHEM respectively. Table 5.6 shows the effectiveness of our sentiment similarity measure. Both models improve the performance over the baselines, while the bridged model leads to higher performance than the series model.

Furthermore, we employ Word Sense Disambiguation (WSD) to disambiguate the adjectives in the question and its corresponding answer. For example, *Q: ... Is that* ***true****?* A: *This is* ***extraordinary*** *and preposterous.* In the answer, the correct sense of the *extraordinary* is *unusual* and as such answer *no* can be correctly inferred. In the table, (*w/o WSD*) is based on the first sense (most common sense) of the words, whereas (*with WSD*) utilizes the real sense of the words. As Table 5.6 shows, WSD increases the performance. WSD could have higher effect, if more IQAPs contain adjectives with senses different from the first sense.

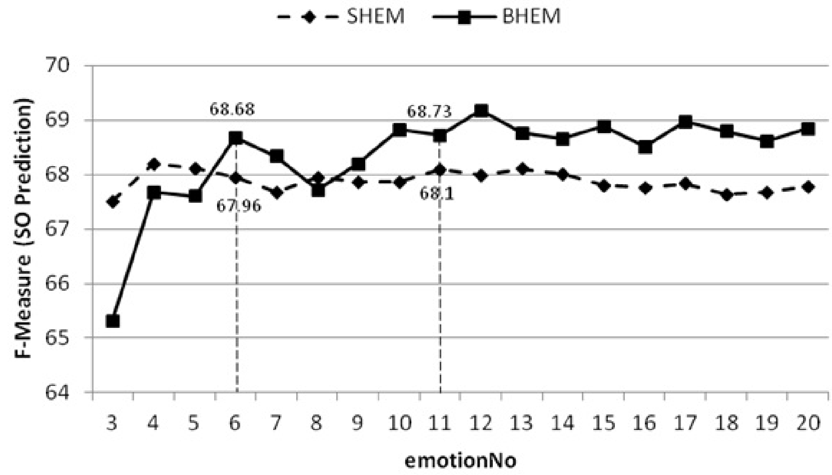


Figure 5.4: Performance of BHEM and SHEM on SO prediction through different number of emotions

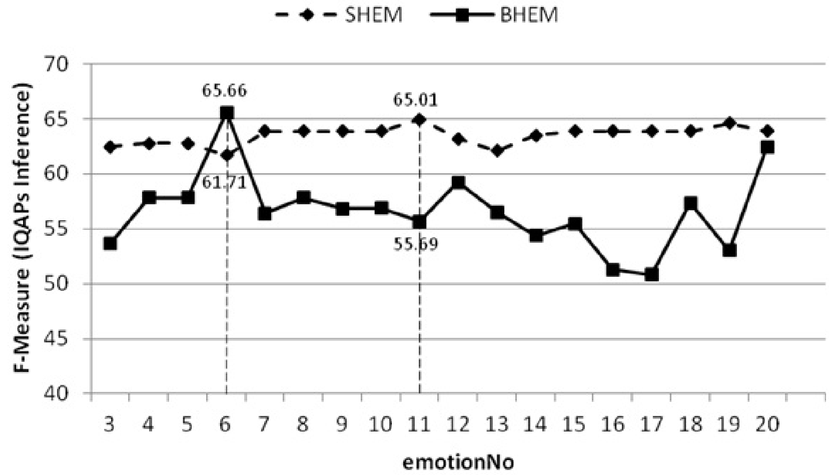


Figure 5.5: Performance of BHEM and SHEM on IQAPs inference through different number of emotions

5.5 Analysis and Discussions

5.5.1 Number and Types of Emotions

In our PSSS approach, there is no limitation on the number and types of emotions as we assumed emotions are hidden. In this Section, we perform experiments to predict the number and type of hidden emotions.

Figures 5.4 and 5.5 show the results of the hidden models (SHEM and BHEM) on SO prediction and IQAPs inference tasks respectively with different number of emotions. As the Figures show, in both tasks, SHEM achieved high performances with 11 emotions. However, BHEM achieved high performances with six emotions. Now, the question is which emotion number should be considered? To answer this question, we further study the results as follows.

First, for SHEM, there is no significant difference between the performances with six and 11 emotions in the SO prediction task. This is the same for BHEM. Also, the performances of SHEM on the IQAP inference task with six and 11 emotions are comparable. However, there is a significant difference between the performances of BHEM in six and 11 emotions. So, we consider the dimension in which both hidden emotional models present a reasonable performance over both tasks. This dimension is six here.

Second, as shown in the Figures 5.4 and 5.5, in contrast to BHEM, the performance of SHEM does not considerably change with different number of emotions over both tasks. This is because, in SHEM, the emotional vectors of the words are derived from the emotional vectors of the documents after the EM steps, see Equation 5.16. However, in BHEM, the emotional vectors are directly obtained from the EM steps. Thus, the bridged model is more sensitive than series model to the number of emotions. This could indicate that the bridged model is more accurate than the series model to estimate the number of emotions.

Emotion#1	**Emotion#2**	**Emotion#3**
excellent (1)	unimpressive (1)	disreputable (1)
magnificently (1)	humorlessly (1)	villian (1)
blessed (1)	paltry (1)	onslaught (1)
sublime (1)	humiliating (1)	ugly (1)
affirmation (1)	uncreative (1)	old (1)
tremendous (2)	lackluster (1)	disrupt (1)

Table 5.7: The top six words for three emotions obtained from BHEM. The numbers in parentheses show the sense of the words

Therefore, based on the above discussion, the estimated number of emotions is six in our development dataset. This number may vary using different development datasets.

In addition to the number of emotions, their types can also be interpreted using our approach. To achieve this aim, we sort the words based on their probability values, $P(w|e)$, with respect to each emotion. Then, the type of the emotions can be interpreted by observing the top k words in each emotion. For example, Table 5.7 shows the top 6 words for three out of six emotions obtained for BHEM. The numbers in parentheses show the sense of the words. The corresponding emotions for these categories can be interpreted as "*wonderful*", "*boring*" and "*disreputable*", respectively.

We also observed that, in SHEM with eleven emotion number, some of the emotion categories have similar top k words such that they can be merged to represent the same emotion. Thus, it indicates that the BHEM is better than SHEM to estimate the number of emotions than SHEM.

5.5.2 Effect of Synsets and Antonyms

We show the important effect of synsets and antonyms in computing the sentiment similarity of words. For this purpose, we repeat the experiment for SO prediction by computing sentiment similarity of word pairs with and without using synonyms and antonyms. Figure 5.6 shows the results obtained from BHEM. As the Figure shows, the highest performance can be achieved when synonyms and antonyms are used, while the lowest

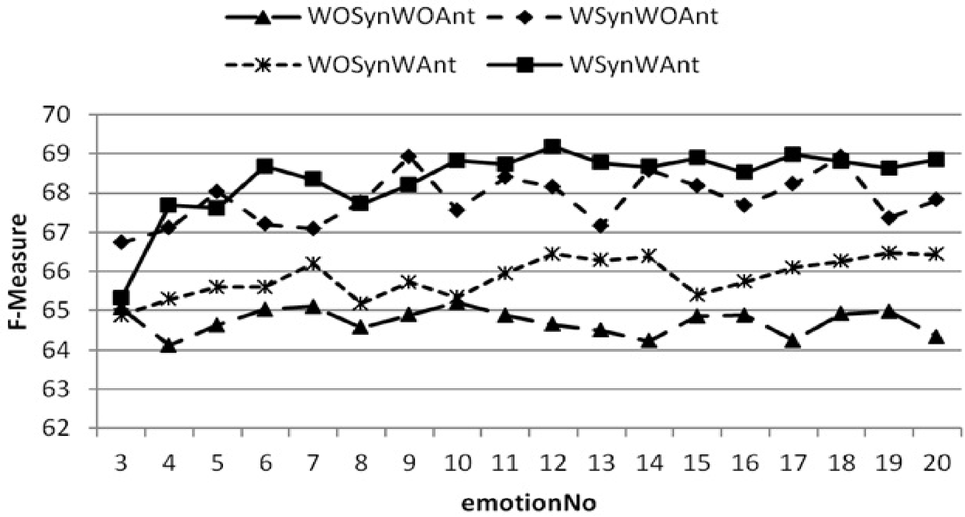


Figure 5.6: Effect of synonyms and antonyms in SO prediction task with different emotion numbers in BHEM

performance is obtained without using them. Note that, when the synonyms are not used, the entries of the enriched matrix are computed using $P(w_i|w_j)$ instead of $P(syn(w_i)|syn(w_j))$ in the Equation 5.17. Also, when the antonyms are not used, the $Max(,)$ in Equation 5.20 is 0 and PSSS is computed using only correlation between words.

The results show that synonyms can improve the performance. As Figure 5.6 shows, the two highest performances are obtained when we use synonyms and the two lowest performances are achieved when we don't use synonyms. This indicates that the synsets of the words can improve the quality of the enriched matrix. The results also show that the antonyms can improve the result (compare WOSynWAnt with WOSynWOAnt). However, synonyms lead to greater improvement than antonyms (compare WSynWOAnt with WOSynWAnt).

5.5.3 Effect of Confidence Value

In Section 5.2.1.1, we defined a confidence value for each word to improve the quality of the enriched matrix. To illustrate the utility of the confidence value, we repeat the experiment for SO prediction by BHEM using all the

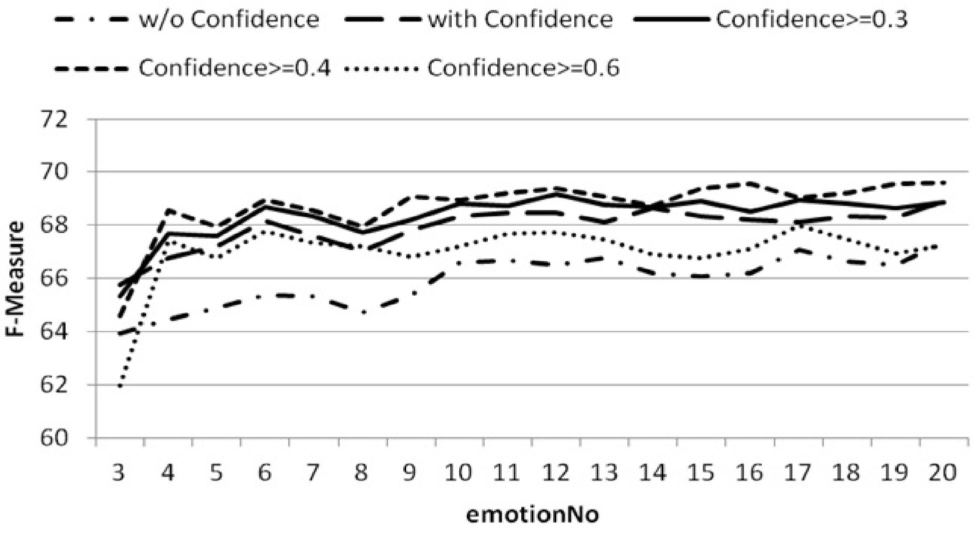


Figure 5.7: Effect of confidence values in SO prediction with different emotion numbers in BHEM

words appearing in the enriched matrix with different confidence thresholds. The results are shown in Figure 5.7, "w/o confidence" shows the results when we don't use the confidence values, while "with confidence" shows the results when use the confidence values. Also, "$confidence > x$" indicates the results when we set all the confidence value smaller than x to 0. The thresholding helps to eliminate the effect of low confidence words.

As Figure 5.7 shows, "w/o confidence" leads to the lowest performance, while "with confidence" improves the performance with different numbers of emotions. The thresholding is also effective. For example, a threshold like 0.3 or 0.4 improves the performance. However, if a large value (e.g., 0.6) is selected as threshold, the performance decreases. This is because a large threshold filters a large number of words from the enriched model that decreases the effect of the enriched matrix.

5.5.4 Convergence Analysis

The PSSS approach is based on the EM algorithm for the BHEM (or SHEM) presented in Table 5.2. This algorithm performs a predefined number of iterations or until convergence. To study the convergence of the al-

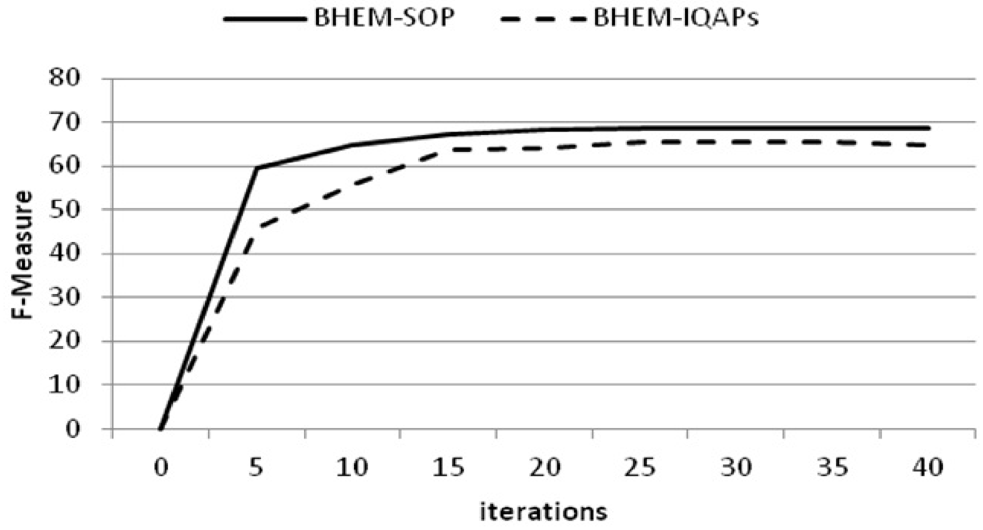


Figure 5.8: Convergence of BHEM

gorithm, we repeat our experiments for SO prediction and IQAPs inference tasks using BHEM with different numbers of iterations. Figure 5.8 shows that after the first 15 iterations the performance does not change dramatically and is nearly constant when more than 30 iterations are performed. This shows that our algorithm will converge in less than 30 iterations for BHEM. We observed the same pattern in SHEM.

5.5.5 Bridged Vs. Series Model

The bridged and series models are both based on the hidden emotions that were developed to predict the sense sentiment similarity. Although their best results on the SO prediction and IQAPs inference tasks are comparable, they have some significant differences as follows:

- BHEM is considerably faster than SHEM. The reason is that, the input matrix of BHEM (i.e., $W \times R$) is significantly smaller than the input matrix of SHEM (i.e., $W \times D$).
- In BHEM, the emotional vectors are directly computed from the EM steps. However, the emotional vector of a word in SHEM is computed using the emotional vectors of the documents containing the word.

This adds noise to the emotional vectors of the words.

- BHEM gives more accurate estimation over types and number of emotions versus SHEM. The reason is explained in Section 5.5.1.

5.6 Summary

We propose a probabilistic approach to infer the sentiment similarity between word senses with respect to automatically learned hidden emotions. We propose to utilize the correlations between reviews, ratings, and words to learn the hidden emotions. We show the effectiveness of our method in two NLP tasks. Experiments show that our sentiment similarity models lead to effective emotional vector construction and significantly outperform semantic similarity measure for the two NLP task.

The Series Hidden Emotional Model (SHEM) in this chapter will be presented in the 27th Conference on Artificial Intelligence, AAAI 2013 (Mohtarami, Lan, and Tan, 2013a), and the Bridged Hidden Emotional Model (BHEM) and comparing it with the SHEM will be presented in the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013 (Mohtarami, Lan, and Tan, 2013b).

Chapter 6

Conclusion and Future Direction

This thesis indicates that although semantic similarity measures can effectively capture the similarity between two entities (e.g., words, phrases or sentences) with respect to their meanings, they are less effective capturing the sentiment similarity between the entities. This thesis is the first major attempt to predict sense sentiment similarity and investigate its impact on improving Indirect yes/no Question Answer Pairs (IQAP) inference and Sentiment Orientation (SO) prediction. We explain the major benefits and contributions of this thesis as follows:

Predicting the Uncertainty of Sentiment Adjectives in Indirect Answers

- In Chapter 3, we investigate the IQAP inference to interpret an answer relative to its question as *yes* or *no* response. To address the task, we employ the similarity measures and show that the similarity between the opinion words (e.g.,adjectives) in the questions and their answers can be the main factor to infer the clear response from an indirect answer. Based on our proposed method, the degree of certainty for the same answer may change in different IQAPs that leads to producing different answers. In addition, we presented the concept of ambiguous sentiment adjectives in IQAPs and attempt to address them with respect to the context.

- Our method involved the following main stages: First, the certainty of the answer is measured with respect to its corresponding question for an IQAPs. Second, a threshold is computed for each IQAP with respect to the antonyms. Finally, the obtained certainty value is evaluated based on its computed corresponding threshold to distinguish if the answer is certain enough with respect to its question to infer *yes* answer or not. Extensive experiments demonstrated the effectiveness of our method over the baseline. In addition, we investigated the role of antonyms, synonyms and word sense disambiguation to tackle the IQAP task.

- In Chapter 3, semantic similarity measures have been employed to compute the similarity between questions and answers. However, since the semantic similarity measures ignore the sentiment to predict similarity of entities, we need another similarity measure that can more accurately capture the similarity with respect to the sentiment than semantic similarities. This leads to our next contributions.

Sense Sentiment Similarity through Emotional Space

- In Chapter 4, we show that although semantic similarity measures are capable of extracting indirect semantic relations between entities and compute their semantic similarities, these methods are not suitable measures to infer the sentimental distance between the entities.

- We propose sense sentiment similarity measure to compute the similarity between words regarding their sentiments and senses. Furthermore, we showed the utility of sense sentiment similarity in two main natural language processing tasks, namely, IQAP Inference and SO prediction.

- Our approach is built on a model which maps from senses of words to vectors of twelve basic emotions. The emotional vectors were used to measure the sentiment similarity of word pairs. Extensive experiments demonstrated the effectiveness of our approach to capture the sentiment similarity of word pairs and to address the IQAP inference and SO-prediction tasks. We showed that sentiment similarity significantly outperforms two popular semantic similarity measures, namely, PMI and LSA.

- According to previous research, there exists a small set of basic emotions which are central to other emotions. Thus, we employ the following set of basic human emotions (Izard, 1971; Ortony and Turner, 1990; Neviarouskaya, Prendinger, and Ishizuka, 2009): *anger, disgust, fear, guilt, sadness, shame, interest, joy, surprise, desire, love, courage*. However, there is little agreement over the number and types of the basic emotions. This leads to our next contributions.

Probabilistic Sense Sentiment Similarity through Hidden Emotions

- In Chapter 5, we suppose that the number and types of the emotions are not clear, that is the emotions are hidden. Then, we propose a probabilistic approach based on the hidden emotional models and Expected Maximization (EM) algorithm to predict the emotional vectors and infer sense sentiment similarity.

- We interpret the number and types of the hidden emotions through the proposed hidden emotional models in which the relations between the words, ratings and reviews are employed.

- Via IQAPs inference task, we show that the best way to predict sense sentiment similarity of words is employing their emotional vectors and show that it is more accurate than only comparing the overall sentiments of the words.

- Via SO prediction task, we show that employing sense sentiment similarity measure along with a simple algorithm can achieve a comparable performance with the state-of-the-art approach to predict sentiment orientation.

6.1 Future Direction

This thesis proposed the approaches based on human basic emotions. Thus, one promising future direction is to extend our exploration on emotion or affective analysis of text (especially, in microblogs like Twitter[1], Facebook[2] and etc), and another type of natural language (i.e., speech). Thus, several future opportunities are envisioned to go beyond the research of this thesis.

Micro-blogs Emotion analysis

- We would like to apply our proposed emotional vectors of the word senses to analyze the emotions of micro-blogs. In micro-blogs like Twitter, there is a limit on the size of the text. Thus, the words, emoticons and abbreviations are key factors to detect their emotional vectors. Since we have already proposed the effective approaches to infer the emotional vectors of the words, the approaches can be extended on predicting the emotional vectors of the emoticons, abbreviations, phrases, sentences and finally whole text of the micro-blogs.

Speech emotion recognition

- We would like to explore the use of the proposed hidden emotional models (in Chapter 5) to recognize the speaker's emotions from a speech utterance. The emotions can be considered as hidden beyond the speech and then the relation between the elements of the speech

[1] `www.twitter.com`

[2] `www.facebook.com`

(e.g., pitch or the energy) can be employed to propose a speech hidden emotional model for emotion recognition.

List of publications arising from this thesis

Mohtarami, Mitra, Man Lan, and Chew Lim Tan. 2013a. From semantic to emotional space in probabilistic sense sentiment analysis. In *the 27th AAAI Conference on Artificial Intelligence.*

Mohtarami, Mitra, Man Lan, and Chew Lim Tan. 2013b. Probabilistic sense sentiment similarity through hidden emotions. In *the 51st Annual Meeting of the Association for Computational Linguistics.*

Mohtarami, Mitra, Hadi Amiri, Man Lan, Thanh Phu Tran, and Chew Lim Tan. 2012. Sense sentiment similarity: an analysis. In *the 26th AAAI Conference on Artificial Intelligence.*

Mohtarami, Mitra, Hadi Amiri, Man Lan, and Chew Lim Tan. 2011. Predicting the uncertainty of sentiment adjectives in indirect answers. In *the 20th ACM International Conference on Information and Knowledge Management*, CIKM'11, pages 2485-2488.

References

Abbasi, Ahmed, Hsinchun Chen, and Arab Salem. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):12.

Aman, Saima and Stan Szpakowicz. 2008. Using roget's thesaurus for fine-grained emotion recognition. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pages 296–302.

Amiri, Hadi and Tat-Seng Chua. 2012. Mining slang and urban opinion words and phrases from cqa services: an optimization approach. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 193–202. ACM.

Arnold, Magda B. 1960. *Emotion and personality.* Columbia University Press.

Balahur, Alexandra and Andrés Montoyo. 2010. Opal: Applying opinion mining techniques for the disambiguation of sentiment ambiguous adjectives in semeval-2 task 18. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 444–447. Association for Computational Linguistics.

Bansal, Mohit, Claire Cardie, and Lillian Lee. 2008. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. *Proceedings of COLING, Companion Volume, Posters*, pages 13–16.

Batson, C. Daniel, Laura L. Shaw, and Kathryn C. Oleson. 1992. Differentiating affect, mood, and emotion: Toward functionally based conceptual distinctions. *Sage Publications, Inc.*

Becker, Israela and Vered Aharonson. 2010. Last but definitely not least: on the role of the last sentence in automatic polarity-classification.

In *Proceedings of the ACL 2010 Conference Short Papers*, pages 331–335. Association for Computational Linguistics.

Bethard, Steven, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. 2004. Automatic extraction of opinion propositions and their holders. In *AAAI Spring Symposium on Exploring Attitude and Affect in Text*.

Blair-Goldensohn, Sasha, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A Reis, and Jeff Reynar. 2008. Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP in the Information Explosion Era*.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Bouma, Gerlof. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Chaumartin, François-Régis. 2007. Upar7: A knowledge-based system for headline sentiment tagging. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 422–425. Association for Computational Linguistics.

Chien, Jen-Tzung and Meng-Sung Wu. 2008. Adaptive bayesian latent semantic analysis. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):198–207.

Choi, Yejin and Claire Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 590–598. Association for Computational Linguistics.

Christiane, Fellbaum. 1998. Wordnet: an electronic lexical database. *Cambrige, MIT Press, Language, Speech, and Communication.*

Dang, Hoa Trang and Karolina Owczarzak. 2008. Overview of the tac 2008 opinion question answering and summarization tasks. In *Proceedings of the 1st Text Analysis Conference.*

Dasgupta, Sajib and Vincent Ng. 2009. Mine the easy, classify the hard: a semi-supervised approach to automatic sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 701–709. Association for Computational Linguistics.

Dave, Kushal, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*, pages 519–528. ACM.

de Marneffe, Marie-Catherine, Christopher D. Manning, and Christopher Potts. 2010. Was it good? it was provocative. learning the meaning of scalar adjectives. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 167–176. Association for Computational Linguistics.

Ding, Chris, Tao Li, and Wei Peng. 2008. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics and Data Analysis*, 52(8):3913–3927.

Ekkekakis, Panteleimon. 2012. Affect, mood, and emotion. In *G. Tenenbaum, R.C. Eklund, and A. Kamata (Eds.), Measurement in Sport and Exercise Psychology*, pages 321–332.

Ekman, Paul, Wallace V. Friesen, and Phoebe Ellsworth. 1982. *What emotion categories or dimensions can observers judge from facial behavior?* New York: Cambridge University Press.

Esuli, Andrea and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422.

Ferrucci, David, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79.

Frijda, Nico H. 1986. *The emotions.* Cambridge University Press.

Gray, Jeffrey A. 1982. *The neuropsychology of anxiety.* Oxford: Oxford University Press.

Green, Nancy and Sandra Carberry. 1999. Interpreting and generating indirect answers. *Computational Linguistics*, 25(3):389–435.

Hassan, Ahmed and Dragomir Radev. 2010. Identifying text polarity using random walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 395–403. Association for Computational Linguistics.

Hatzivassiloglou, Vasileios and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics*, pages 174–181. Association for Computational Linguistics.

Hoang, Hung Huu, Su Nam Kim, and Min-Yen Kan. 2009. A re-examination of lexical association measures. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 31–39. Association for Computational Linguistics.

Hockey, Beth Ann, Deborah Rossen-Knill, Beverly Spejewski, Matthew Stone, and Stephen Isard. 1997. Can you predict responses to yes/no questions? yes, no, and stuff. In *Proceedings of the Eurospeech*, volume 97.

Hofmann, Thomas. 1999a. The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data. In *International Joint Conference on Artificial Intelligence*, volume 16, pages 682–687. Citeseer.

Hofmann, Thomas. 1999b. Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc.

Hofmann, Thomas. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196.

Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177. ACM.

Islam, Aminul and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(2):10.

Izard, Carroll E. 1971. *The face of emotion*, volume 23. Appleton-Century-Crofts New York.

James, William. 1884. What is an emotion? *Mind*, (34):188–205.

Jarmasz, Mario and Stan Szpakowicz. Roget's thesaurus: A lexical resource to treasure. *CoRR*, abs/1204.0258.

Jiang, Jay J. and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, cmp-lg/9709008.

Jijkoun, Valentin, Maarten de Rijke, and Wouter Weerkamp. 2010. Generating focused topic-specific sentiment lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 585–594. Association for Computational Linguistics.

Jurafsky, D. and J.H. Martin. 2009. *Speech and language processing: An introduction to natural language processing , computational linguistics, and speech recognition.* Prentice Hall.

Kamps, Jaap, M.J. Marx, Robert J. Mokken, and Maarten De Rijke. 2004. Using wordnet to measure semantic orientations of adjectives. In *LREC*, pages 1115–1118.

Kanayama, Hiroshi and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363. Association for Computational Linguistics.

Katz, Phil, Matthew Singleton, and Richard Wicentowski. 2007. Swat-mp: the semeval-2007 systems for task 5 and task 14. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 308–313. Association for Computational Linguistics.

Kim, Soo-Min and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, page 1367. Association for Computational Linguistics.

Kim, Soo-Min and Eduard Hovy. 2007. Crystal: Analyzing predictive opinions on the web. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1056–1064.

Landauer, Thomas K. and Susan T. Dumais. 1996. How come you know so much? from practical problem to theory. *Basic and Applied Memory: Memory in context*, pages 105–126.

Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284.

Leacock, Claudia and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An Electronic Lexical Database*, 49(2):265–283.

Lesk, Michael. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26. ACM.

Lin, Dekang. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, volume 1, pages 296–304.

Lindsey, Robert, Vladislav D. Veksler, Alex Grintsvayg, and Wayne D. Gray. 2007. Be wary of what your computer reads: the effects of corpus selection on measuring semantic relatedness. In *8th International Conference of Cognitive Modeling, ICCM*.

Liu, Bing. 2007. *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Verlag.

Liu, Bing. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 2:568.

Lu, Yue, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *Proceedings of the 20th International Conference on World Wide Web*, pages 347–356. ACM.

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 142–150. Association for Computational Linguistics.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

Manning, Christopher D. and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.

McDougall, William. 1926. *An introduction to social psychology.* Boston: Luce.

Mohtarami, Mitra, Hadi Amiri, Man Lan, and Chew Lim Tan. 2011. Predicting the uncertainty of sentiment adjectives in indirect answers. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2485–2488. ACM.

Mohtarami, Mitra, Hadi Amiri, Man Lan, Thanh Phu Tran, and Chew Lim Tan. 2012. Sense sentiment similarity: an analysis. In *the 26th AAAI Conference on Artificial Intelligence.*

Mohtarami, Mitra, Man Lan, and Chew Lim Tan. 2013a. From semantic to emotional space in probabilistic sense sentiment analysis. In *the 27th AAAI Conference on Artificial Intelligence.*

Mohtarami, Mitra, Man Lan, and Chew Lim Tan. 2013b. Probabilistic sense sentiment similarity through hidden emotions. In *the 51st Annual Meeting of the Association for Computational Linguistics.*

Na, Seung-Hoon, Yeha Lee, Sang-Hyob Nam, and Jong-Hyeok Lee. 2009. Improving opinion retrieval based on query-specific sentiment lexicon. In *Advances in Information Retrieval.* Springer, pages 734–738.

Neviarouskaya, Alena, Helmut Prendinger, and Mitsuru Ishizuka. 2007. Textual affect sensing for sociable and expressive online communication. In *Affective Computing and Intelligent Interaction.* Springer, pages 218–229.

Neviarouskaya, Alena, Helmut Prendinger, and Mitsuru Ishizuka. 2009. Sentiful: Generating a reliable lexicon for sentiment analysis. In *Affective Computing and Intelligent Interaction, ACII*, pages 1–6. IEEE.

Neviarouskaya, Alena, Helmut Prendinger, and Mitsuru Ishizuka. 2010. Recognition of affect, judgment, and appreciation in text. In *Pro-*

ceedings of the 23rd International Conference on Computational Linguistics, pages 806–814. Association for Computational Linguistics.

Neviarouskaya, Alena, Helmut Prendinger, and Mitsuru Ishizuka. 2011. Affect analysis model: novel rule-based approach to affect sensing from text. *Natural Language Engineering*, 17(1):95.

Olveres, Jimena, Mark Billinghurst, Jesus Savage, and Alistair Holden. 1998. Intelligent, expressive avatars. *Proceedings of WECC*, 98:47–55.

Ortony, Andrew and Terence J. Turner. 1990. What's basic about basic emotions. *Psychological Review*, 97(3):315–331.

Osgood, Charles Egerton, George John Suci, and Percy H. Tannenbaum. 1957. *The measurement of meaning*, volume 47. Urbana: University of Illinois Press.

Ounis, Iadh, Craig Macdonald, Maarten de Rijke, Gilad Mishne, and Ian Soboroff. 2006. Overview of the trec 2006 blog track. In *TREC*.

Pang, Bo and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing*, pages 79–86. Association for Computational Linguistics.

Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL*, pages 38–41. Association for Computational Linguistics.

Peng, Wei. 2009. Equivalence between nonnegative tensor factorization and tensorial probabilistic latent semantic analysis. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 668–669.

Potts, Christopher. 2011. On the negativity of negation. In *Proceedings of SALT*, volume 20, pages 636–659.

Read, Jonathon. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pages 43–48. Association for Computational Linguistics.

Russell, James A. 2003. Core affect and the psychological construction of emotion. *Psychological Review*, 110(1):145.

Schneider, Karl-Michael. 2005. Weighted average pointwise mutual information for feature selection in text categorization. In *Knowledge Discovery in Databases: PKDD 2005*. Springer, pages 252–263.

Stone, Philip J. 1997. Thematic text analysis: New agendas for analyzing text content. *Text Analysis for the Social Sciences*, pages 35–54.

Stone, Philip J., Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The general inquirer: A computer approach to content analysis*. MIT Press.

Strapparava, Carlo and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics.

Strapparava, Carlo and Rada Mihalcea. 2008. Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, pages 1556–1560. ACM.

Strapparava, Carlo and Alessandro Valitutti. 2004. Wordnet-affect: an affective extension of wordnet. In *Proceedings of LREC*, volume 4, pages 1083–1086.

Takamura, Hiroya, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 133–140. Association for Computational Linguistics.

Tang, Huifeng, Songbo Tan, and Xueqi Cheng. 2009. A survey on sentiment detection of reviews. *Expert Systems with Applications*, 36(7):10760–10773.

Turney, Peter and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*.

Turney, Peter D. 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*, pages 417–424. Association for Computational Linguistics.

Wan, Xiaojun. 2008. Using bilingual knowledge and ensemble techniques for unsupervised chinese sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 553–561. Association for Computational Linguistics.

Wiebe, Janyce, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Wiebe, Janyce M., Rebecca F. Bruce, and Thomas P. O'Hara. 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*, pages 246–253. Association for Computational Linguistics.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354. Association for Computational Linguistics.

Wu, Yunfang and Peng Jin. 2010. Semeval-2010 task 18: Disambiguating sentiment ambiguous adjectives. In *Proceedings of the 5th Interna-*

tional Workshop on Semantic Evaluation, pages 81–85. Association for Computational Linguistics.

Wu, Zhibiao and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

Yi, Jeonghee, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *the 3rd IEEE International Conference on Data Mining, ICDM*, pages 427–434. IEEE.

Zhong, Zhi and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83. Association for Computational Linguistics.