RESOURCES FOR SENTIMENT ANALYSIS

A Seminar Report

Submitted in partial fulfillment of requirements for the degree of
Master of Technology (M. Tech)

by

Sagar Ahire
Roll No: 133050073

Under the guidance of
Prof. Pushpak Bhattacharyya

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
ABSTRACT

Sentiment Analysis refers to the problem of automatically determining the polarity of the opinion expressed by an author in a text. Two competing approaches to solve this problem are the classifier-based and resource-based approaches. The resource-based approach involves creation of dictionaries of lexical units of a language with their sentiment values and using these dictionaries for sentiment analysis of a text. This report covers the resource-based approach, including background work that led to the lexicons, a few case-studies covering the structure of the lexicons and the process of their creation, and recent developments, including those at IIT Bombay. The report concludes with several directions that can be taken in the future.
CONTENTS

1 INTRODUCTION 1
  1.1 Sentiment Analysis 1
  1.2 Motivation for Sentiment Resources 3
  1.3 Sentiment Lexical Resources 4
  1.4 Roadmap 6

2 BACKGROUND 7
  2.1 Sentiment Orientation Detection 7
  2.2 Subjectivity Detection 10

3 SENTIWORDNET 13
  3.1 Introduction 13
  3.2 Process of Creation 15
  3.3 Results 17

4 SO-CAL 18
  4.1 Introduction 18
  4.2 Features Used 18
  4.3 Results 21

5 WORDNET-AFFECT 25
  5.1 Introduction 25
  5.2 Structure 25
  5.3 Process of Creation 26

6 WORK AT IIT BOMBAY 28
  6.1 Indian-Language Sentiwordnets 28
  6.2 Wordnet-Linking for Cross-Language Sentiment Analysis 30

7 CONCLUSIONS AND FUTURE WORK 33
  7.1 Approaches of Building Sentiment Lexicons 33
  7.2 Choices for Representing Sentiment 34
  7.3 Future Work 35

BIBLIOGRAPHY 38
INTRODUCTION

Sentiment Analysis refers to the problem of determining the opinion expressed by an author towards a particular entity in a text. For instance, determining whether a movie review on a blog is positive or negative towards the movie is a sentiment analysis problem. Natural Language Processing techniques are apt for the task as this problem can be seen as either of the following, both of which are important (and competing) approaches in NLP:

- Learning the classification of a document using statistical techniques (i.e. corpus linguistics).
- Analyzing the meaning of the document using knowledge about its language itself.

While both of these approaches have their advantages and drawbacks, this report will concentrate on a technique which predominantly uses the second approach - use of lexical resources for Sentiment Analysis, which contain mappings of lexical units and their sentiment values.

This chapter is organized as follows. In Section 1.1, we examine the problem of sentiment analysis in short, including its applications. In Section 1.2, we examine the motivations of using sentiment lexicons for sentiment analysis. In Section 1.3, we give an introduction to lexical sentiment resources, including an account of the ways in which sentiment lexicons can be created. In Section 1.4, we give a roadmap to the rest of the report.

1.1 SENTIMENT ANALYSIS

Sentiment Analysis refers to the problem of inferring the sentiment expressed by an author from some text. This is a problem particularly suited to NLP as it is a special case of text analysis. Sentiment Analysis can be carried out at various levels depending on the granularity of the text.

1. **Word Level:** This involves ascertaining the sentiment carried by a word all by itself, i.e. without any contextual clues given by surrounding text. Thus, this can be viewed as determining the prior polarity of a word.

2. **Sentence Level:** This involves ascertaining the sentiment carried by a sentence. This gives some context, but additional processing is required for clues that go beyond the sentence (such as cross-sentence co-reference resolution)

3. **Document Level:** This involves ascertaining the sentiment carried by an entire document. While this makes all the information available (apart from world knowledge), difficulty is introduced by the possibility of multiple sentiments in the same document.
4. **Aspect Level**: This involves extracting the sentiments directed towards an entity or a particular aspect (feature) of an entity from a document (or a collection of documents). This is the most complex of all levels as it subsumes all the previous levels and adds further complexities [21].

Motivated by the complexity of aspect-level sentiment analysis, a definition of sentiment considers a sentiment as a 5-tuple as follows: [21]

\[
S = (e_l, a_{ij}, s_{ijkl}, h_k, t_l)
\]

where:
- \(e_l\) is the entity under consideration
- \(a_{ij}\) is a particular aspect or feature of the entity
- \(h_k\) is an sentiment holder
- \(t_l\) is the time at which the sentiment is held
- \(s_{ijkl}\) is the sentiment about aspect \(a_{ij}\) of entity \(e_l\) held by \(h_k\) at time \(t_l\).

For instance, consider the statement “Last week Deepali found out that the Nexus 4’s camera is very poor”. In this sentence, \(e_l = \text{Nexus 4}\), \(a_{ij} = \text{camera}\), \(h_k = \text{Deepali}\), \(t_l = \text{last week}\), \(s_{ijkl} = \text{poor}\), which, on sentiment analysis, should be found to be negative.

### 1.1.1 Applications

Sentiment Analysis is extremely vital, particularly in the times of the Internet as the sheer amount of opinionated text on blogs, social networking websites, etc. is rapidly increasing and there is no practical way all these opinions can be analyzed manually. However, analysis of opinions is extremely important as humans tend to depend on others’ opinions before taking decisions, ranging from what song to listen to which university to attend. Some of the important applications of Sentiment Analysis can be stated as follows [25]:

1. **Prospective Consumers**
   Sentiment Analysis can be useful for prospective consumers who are looking for information on which product is the best to buy among a set of choices. It is possible to aggregate the opinions of users of the products across various online media and distill the opinions down to a succinct summary or even a single value, making it easier for a buyer to choose a product.

2. **Market Research**
   Sentiment Analysis can also be very helpful at the other end of the spectrum – the manufacturers. A similar aggregation technique can be used for the manufacturer of a particular product to evaluate its reception in the market. Moreover, once the manufacturer ascertains the negative reactions for the product, it is possible for the manufacturer to work towards solving the grievances.

3. **Recommendation Systems**
   Sentiment Analysis can be incorporated into the recommendation systems built into certain websites that suggest products to buy depending
on the user’s past purchases. Products with more positive reactions can be recommended more often in hopes that the current user will like them too.

4. **SPAM and Flame Detection**
   SPAM e-mails are mass e-mails that are usually annoying at best and harmful at worst. Flames or “trolls” are messages on social networking sites or forums that are unnecessarily hateful and provocative. Sentiment Analysis can be adapted towards detecting and weeding out both of these.

5. **Question Answering**
   Question Answering applications can benefit from the addition of sentiment analysis, particularly when the question, the answer(s), or both, are potentially opinionated and thus need to be handled differently from strictly factual ones.

6. **Summarization**
   If a document is particularly opinionated, it is necessary to preserve the opinions while summarizing it. In addition, if a document has multiple opinions, all opinions need to be represented adequately and proportionately.

7. **Politics and Government Intelligence**
   Sentiment Analysis can be useful in predicting the outcome of elections and the reasons behind it. In addition, governments can use sentiment analysis to monitor the mood of the nation and initiate policies which may appease the vote-bank according to the data.

### 1.2 Motivation for Sentiment Resources

One of the dominant approaches to sentiment analysis is turning it into a standard machine learning/text classification problem by converting the document into a feature vector. The major allure behind this approach is that only the feature engineering stage is specific to sentiment analysis. Once that has been performed, any of the several commonly used and widely understood machine learning approaches can be used for the subsequent classification. In this way, sentiment analysis can be reduced to any of:

- A two-class or three-class classification problem with the classes as *positive*, *negative* and optionally *objective*.

- A multi-class classification problem with classes as a scale describing both the sentiment orientation and its strength (such as *strongly positive*, *mildly positive*, *neutral*, *mildly negative*, *strongly negative*).

- A regression problem which maps the sentiment of a document to a number.

For instance, Pang et al. [26] used a variety of features and trained the resultant vectors using the Naive Bayes, Maximum Entropy and Support Vector
Machine classifiers on a 2000-document movie review corpus. The highest accuracy was obtained for unigrams using SVM at 82.9% which, while impressive, was far below the state-of-the-art in standard 2-class text classification which could reach accuracies as high as over 95% [13].

The reason for this change in performance is that sentiment analysis is fundamentally different from text classification and is therefore not as suited for machine-learning techniques. The following are some fundamental challenges due to which classifier methods are not optimally suitable for sentiment analysis:

1. **Domain-specificity**
   Classifier-based methods work well when trained on a corpus of a particular domain, which is why text classification performs so well using classifier methods. However, this is primarily because the classifier learns several features that are domain-specific and may not hold in other domains or even cause sentiment drift. For instance, an SVM from Brooke [5] trained on movie reviews learnt that if the phrases `writer`, `director`, `plot`, `script` are mentioned then the review is likely to be negative, while `performances`, `ending` and `flaws` indicate a positive review. Moreover, there exist examples like “Go read the book” which is likely to be positive in a book review but negative in a movie review.

2. **Lack of Context**
   During the feature extraction stage, a document is vectorized into a bit representation. This may preserve some information from the document at the expense of leaving out other, possibly vital information, mainly context. For instance, using unigram features, information about the order of words is entirely lost, and it is not feasible to use higher order n-grams to capture long-distance dependencies [8]. As a result, the following phrases all look very similar to a classifier even though the polarities are vastly different – “good”(+) , “not good”(−) , “not very good”(+) , “… do not think that this is any good”(−), etc. In addition, the document-level granularity further exacerbates the problem.

Use of a sentiment lexicon can solve these shortcomings of classifier-based approaches as they can be made domain-independent, and once the sentiment values for individual words are obtained, they can be further combined to arbitrary levels to incorporate context information without fearing a combinatorial explosion as is the case for classifier-based methods.

### 1.3 Sentiment Lexical Resources

Now that the use of sentiment lexicons has been motivated, the following section takes a brief look at the concept.

#### 1.3.1 Introduction

A lexical resource for sentiment analysis, also referred to as a *Sentiment Lexicon*, is a database of lexical units for a language along with their sentiment orientations. This can be expressed as a set of tuples of the form *(lexical
unit, sentiment). Here, the lexical units may be words, word senses, phrases, etc. On the other hand, the sentiment may be represented in several possible forms, some of which are:

- Fixed categorization into positive or negative,
- A finite number of graded sets such as strongly positive, mildly positive, neutral, mildly negative, strongly negative,
- A real value denoting sentiment strength in an interval such as \([-1, +1]\).

Once such a lexicon is available, it can be used appropriately to perform sentiment analysis on a document, either alone or in combination with classifier methods. For example, if a sentiment lexicon contains sentiment values in the range of \([-1, +1]\), a naive approach to sentiment analysis of a document would be to add up the sentiment values of all the words in the document and then conclude that the document is positive if the total sentiment is above 0, otherwise negative.

1.3.2 Ways to Create Sentiment Lexicons

There are two broad approaches to creation of sentiment lexicons – manual and automated.

1.3.2.1 Manual Creation of Sentiment Lexicons

Creation of a sentiment lexicon manually involves merely deciding on the structure of the sentiment lexicon, i.e. the values of the 2-tuple \((\text{lexical unit}, \text{sentiment})\), and then annotating a list of lexical units with their sentiment value. The list can be obtained from a dictionary or a corpus. In order to introduce robustness in the results, multiple annotators can be asked to perform the task and the inter-annotator agreement can be calculated. As a result, no computational or algorithmic complexity is involved.

The major advantage of this approach is that since the annotation is performed by humans, correctness is guaranteed barring an actual error in annotation. This is a desirable property as sentiment analysis using a correct resource is bound to perform better, and there are times when correctness requires innate human judgement while classifiers may get misled.

However, the problem with this approach is the immense time investment required. Consider the 2nd Edition of the Oxford English Dictionary, which has 291,500 words [1]. Taking a very conservative estimate of 90 seconds required for actually annotating the sentiment of the word and 30 seconds for post-processing per word to enter it in the database, this requires over 1,200 days working for 8 hours a day without any breaks. Due to this, the sizes of manual sentiment lexicons have been restricted to a few thousand words at most, adversely affecting coverage.

1.3.2.2 Automatic Creation of Sentiment Lexicons

The disadvantage of manual sentiment lexicons can be remedied by using automatic methods to create sentiment lexicons. While there are several methods
to create sentiment lexicons, one of the most popular is to create a set of starting seed words with known sentiment orientation, and then expand that seed set using an already existing lexical resource. While this is the most commonly used method, other ways of creating sentiment lexicons have also been explored, such as bootstrapping, which does not require a lexicon and learns patterns from a corpus instead. Several methods for creating automated sentiment lexicons have been described briefly in Chapter 2.

However, the advantages of the automatic approach in terms of the promise of high coverage are achieved only by a trade-off in accuracy of the lexicon as the methods used are far from perfect, as has also been explored in Chapter 2 and the subsequent chapters.

1.4 ROADMAP

The rest of the report is organized as follows:

- **Chapter 2** covers the initial work that paved the way for creation of some of the automated sentiment lexicons. It covers work on word sentiment orientation detection and subjectivity detection, opening the door to 3-way sentiment classification of words as *positive*, *negative* and *objective*.

- **Chapter 3** covers *Sentiwordnet*, a major resource created by automatically annotating each synset of Wordnet with a triplet of sentiment scores, one each for *positive*, *negative* and *objective*.

- **Chapter 4** covers *SO-CAL*, a system based on a manually created resource assigning a single sentiment score to each word. It is much smaller than Sentiwordnet but enjoys higher accuracy due to manual annotation.

- **Chapter 5** covers *Wordnet-Affect*, a semi-automatically generated resource that annotates each Wordnet synset with its sentiment and emotion information. It is aimed at solving the problem of *Affective Computing* which can be perceived as a superset of sentiment analysis.

- **Chapter 6** covers the *Hindi Sentiwordnet* made at IIT-Bombay by projecting the English Sentiwordnet to Hindi by using Wordnet linking.

- **Chapter 7** concludes, giving an insight into some pertinent issues regarding sentiment resources and also gives several possible ideas for future work.

**SUMMARY**

This chapter gave an introduction to the rest of the report. We saw an introduction to the sentiment analysis problem itself and motivated its need with a few of its applications. Then we moved on to motivating the need of sentiment lexicons by studying the drawbacks of the competing approach i.e. the classifier-based sentiment analysis approach. Then we finally saw what lexical sentiment resources are, and a short overview of the two competing ways to create such resources. Finally we ended with a look at what is in store in the rest of the report.
As a sentiment lexicon is a mapping from a lexical unit to a sentiment, creation of such a lexicon involves discovering the sentiment associated with each lexical unit in the language. While it might not be very difficult from an algorithmic perspective to perform this task manually, achieving any respectable amount of coverage is near-impossible considering the time investment required. The road to realizing most sentiment lexicons today was paved by research that aimed at automatically discovering the sentiment orientation of a word from some pre-existing resource. These approaches have then been subsequently used to create the lexicons themselves.

This chapter is organized as follows. In Section 2.1, we examine approaches to classify the sentiment of a word as either positive or negative using a variety of Machine Learning techniques. In Section 2.2, we then concentrate on the problem of detecting whether a word is subjective so that it can then be classified as positive or negative.

2.1 SENTIMENT ORIENTATION DETECTION

This section delves into the initial approaches used to automatically detect the sentiment orientation of a word.

2.1.1 Conjunction-Separated Adjectives

Hatzivassiloglou and McKeown [12] was one of the first works to tackle the problem of automatically determining the sentiment of words. It concentrated only on adjectives, as has since become common in most sentiment analysis research. The assumption used was that adjectives separated by certain conjunctions within a sentence have certain limits on their sentiment orientation, and so it would be possible to extract such pairs of adjectives.

For instance, consider the following sentences:

| The movie was shallow\(^(-)\) and boring\(^(-)\). |
| The movie was shallow\(^(-)\) but enjoyable\(^(+\).) |
| * The movie was shallow\(^(-)\) and enjoyable\(^(+\).) |

This example shows that if two adjectives are separated by and, they tend to have the same sentiment orientation. However, use of but flips the sentiment orientation. This is why the third sentence in the example appears incorrect – and is used to continue the same polarity, not flip it.

The conjunctions used included and, or, but, either-or, and neither-nor. Then, the following steps were used to classify the semantic orientation of a set of adjectives from a corpus:
• Pairs of adjectives separated by the aforementioned conjunctions were obtained by parsing sentences from the Wall Street Journal corpus [7]. In addition, their morphological content was also extracted.

• Then a log-linear model was run on these pairs to determine if the adjectives are of the same or different orientation.

• Then a clustering algorithm was used to combine words of the same orientation in the same cluster.

• The cluster with more terms is labelled as positive, owing to language being inherently biased towards positivity [4].

The final term set used contained 657 positive adjectives and 679 negative adjectives. The accuracy they achieved on this set was about 78%.

Results

2.1.2 Semantic Orientation using PMI

While similar to [12] in that it concentrated on adjectives too, Turney and Littman [34] chose a different approach to achieve the same goal. They started off with a small seed set of adjectives and then expanded it using web queries to find more adjectives with similar orientation. The assumption underlying the approach is that the sentiment orientation of an adjective is similar to that of its neighbours. This allows calculation of what has been referred to as semantic orientation by association, implying that words with a high association with positive words are positive, and similarly for negative words.

The two seed sets of adjectives used were:

\[
S_p = \{\text{good, nice, excellent, positive, fortunate, correct, superior}\}
\]
\[
S_n = \{\text{bad, nasty, poor, negative, unfortunate, wrong, inferior}\}
\]

where \(S_p\) denotes adjectives having positive orientation and \(S_n\) denotes adjectives having negative orientation.

This seed set was then expanded by issuing web queries to the Altavista search engine using the NEAR operator on the adjectives in the seed set. Once that was performed, the association was quantified using the Pointwise Mutual Information (PMI) measure as follows:

\[
\text{PMI}(w_1, w_2) = \log_2 \left( \frac{p(w_1 \& w_2)}{p(w_1)p(w_2)} \right) \quad (2.1)
\]

Here, \(p(w_1 \& w_2)\) is the probability that words \(w_1\) and \(w_2\) co-occur in the same document while \(p(w_1)\) and \(p(w_2)\) are the individual occurrence probabilities. Thus, the PMI is high when the two words co-occur in more documents. Thus PMI is a valid measure for measuring association.

Once the association is measured, it can be used to determine the sentiment orientation of a word using its association with the words in the seed set. This is performed using the following formula:

\[
O(w) = \sum_{w_1 \in S_p} \text{PMI}(w, w_1) - \sum_{w_1 \in S_n} \text{PMI}(w, w_1) \quad (2.2)
\]
As a result of this, if the word $w$ has a high PMI with positive words, its orientation is positive as the first term will have a higher value. Otherwise the second term will have a higher value and so its orientation will in turn have a negative value.

The term set used contained 1,614 positive terms and 1,982 negative terms. The accuracies obtained were 87% on the term set from [12] and 83% on this term set [9]. This is an improvement of 9% over the approach in [12].

2.1.3 Wordnet-based Expansion from Seed Set

Kamps et al. [16] used an approach that makes use of an already existing lexical resource, Wordnet [11] to infer the semantic orientation of adjectives. Wordnet is a lexical database in which words are grouped together by sense and these groupings, called synsets, are then linked to each other by various relations. The major contribution of this work was to visualize these Wordnet relations as a graph.

The relation considered was synonymy. Using the resultant graph consisting of words as nodes and the synonymy relation as edges, a distance metric called geodesic distance was defined to be the shortest path between two words. The assumption behind using synonymy and shortest path distance was that the more closely synonymous two words are to each other, the more similar their sentiment orientations are likely to be.

The seed set used consisted of only two words: good and bad. Then the following formula was used to find the sentiment orientation of any other word:

$$O(w) = \frac{d(w, \text{bad}) - d(w, \text{good})}{d(\text{good}, \text{bad})}$$  \hspace{1cm} (2.3)

Here, $d(w_1, w_2)$ is the geodesic distance, i.e. the shortest path distance between $w_1$ and $w_2$. This formulation is very similar to the one by [34] in Equation 2.2, except for the fact that while PMI has to be maximized, the geodesic distance has to be minimized, and so the terms in the numerator are interchanged, i.e. the metric associated with the positive term is subtracted from the one associated with the negative term.

The major problem with this approach was that it was useful only for the terms that are connected to either good or bad in the synonymy graph. This led to a term set of 663 terms. The accuracy obtained was about 67%.

2.1.4 Classification using Wordnet Glosses

The idea of using already existing lexical resources like Wordnet also inspired another key work in Esuli [9, Ch. 1]. This work overcomes the major disadvantage of the method in [16] and can now be used for every term in Wordnet. Moreover, it is also adaptable to any other dictionary or thesaurus. The idea involves using not the Wordnet relations, but the gloss of the terms in the dictionary to classify the sentiment orientation of a word. The gloss of a word in a dictionary refers to the description of a word and optionally a few example phrases showing the usage of the word. An example follows:

URL: http://wordnetweb.princeton.edu/perl/webwn?s=beautiful
The assumption behind using the gloss of a term for classifying its sentiment orientation was that words with similar glosses tend to have similar sentiment orientations.

The following are four possible ways of representing the gloss of a term:

- **TD**: Uses the Terms in a Wordnet synset and its Definition.
- **TD¬**: Uses TD with negation representation (for example, not good becomes ¬good).
- **TDS**: Uses the Terms, Definitions and Sample Phrases.
- **TDS¬**: Uses TDS with negation representation.

These glosses are then vectorized and then fed to a standard text classifier.

In order to generate the training set for the classifier, two starting seed sets \( T_{p0} \) and \( T_{n0} \) are chosen and then they are iteratively expanded for \( k \) iterations using the lexical relations of Wordnet to get the final training sets \( T_{pK} \) and \( T_{nK} \).

Experiments were performed using the term sets of \([12],[34]\) and \([16]\). It was found that the TDS¬ representation was the best performing representation yielding accuracies of 87\%, 83\% and 88\% on the three test sets respectively. This was a major improvement over the other three methods.

### 2.2 Subjectivity Detection

All the works considered in Section 2.1 concentrate on classifying a particular term as positive or negative. However, this already assumes that a particular term is subjective. The bigger problem that needs solving is detecting whether a term is subjective or objective, and only the subjective terms should further be classified as positive or negative. This is especially important as the problem of subjectivity detection itself is non-trivial. Moreover, if all the objective terms are identified as such and eliminated from further sentiment analysis it will eliminate the possibility of words being wrongly classified as positive or negative when they do not have any sentiment orientation of their own to begin with.

#### 2.2.1 Classification using Wordnet Gloses

The approach of Esuli [9, Ch. 2] is an extension of the approach used in [9, Ch. 1] to handle subjectivity detection. Thus it handles the bigger problem of determining whether a term is positive, negative, or objective and thus subsumes both the problems of subjectivity detection and sentiment detection. In order to adapt to this new problem, several changes were required to the original approach.

First and foremost, the approach now requires three training sets, and as a result, three seed sets \( T_{p0} \), \( T_{n0} \) and \( T_{o0} \). While \( T_{p0} \) was selected as \{good\}
and $\text{Tr}_n^0$ was selected as \{bad\} as per [16]. $\text{Tr}_n^0$ was selected as \{entity\}. The reason for this is that entity is the root node of Wordnet and thus can reach every node in the Wordnet hierarchy. Also, terms referring to entities usually are not known to be subjective.

Then during the iterative expansion of the seed sets to form the training sets, care is taken to ensure that if a term is in either $\text{Tr}_p^k$ or $\text{Tr}_n^k$ then it is not added to $\text{Tr}_o^k$.

The actual classification process again requires a change as the classification problem is now 3-way rather than the previous 2-way problem. There are three possible ways in which the positive vs. negative classification approaches can be modified for this new problem:

- **Approach 1**: This approach involves learning two binary classifiers: one classifying subjective vs. objective and the other classifying the subjective terms into positive or negative.

- **Approach 2**: This approach also involves learning two binary classifiers. However, these classifiers are positive vs. not positive and negative vs. not negative respectively. Then, the final classifications are decided as follows:

<table>
<thead>
<tr>
<th>POSITIVE</th>
<th>NOT POSITIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective</td>
<td>Negative</td>
</tr>
<tr>
<td>Positive</td>
<td>Objective</td>
</tr>
</tbody>
</table>

  Thus this approach regards objectivity as the absence of positivity or negativity rather than a class by itself.

- **Approach 3**: This approach simply involves learning a ternary classifier to classify positive vs. negative vs. objective.

The evaluation was performed on two fronts: **SO-Accuracy**, which was the accuracy on classifying subjective vs. objective, and **PNO-Accuracy**, which was the accuracy on classifying positive vs. negative vs. objective. Approach 2 was found to be the best performing approach with an average SO-Accuracy of 64% and a PNO-Accuracy of 61%. This shows that once subjectivity detection is performed, orientation detection is a very easy task as much of the noise in the term set is removed. However, subjectivity detection is in itself a very hard problem; much harder than orientation detection alone.

### 2.2.2 Bootstrapping

Riloff and Wiebe [28] approached the problem of subjectivity detection through bootstrapping. This process involves starting with a seed set consisting of known subjective terms and if a sentence is classified as subjective using the current term set, any other terms in the sentence are added to the term set. The assumption being that if the current set of terms detected a sentence as subjective, it may contain other subjective terms which can be discovered from the sentence.

The following were the steps used:

- First a seed set was created containing about 20 known subjective terms.
• An unannotated corpus was then fed to two classifiers that used this known set. These were sentence-level classifiers responsible for determining if a sentence is subjective or objective respectively.

• The sentences that were detected to be either subjective or objective were fed to an extraction pattern learner which used this information to learn about patterns that determine subjectivity and objectivity.

• This process was repeated for every sentence in the entire corpus.

The precision for this approach ranged from 71% to 85%. However, as the method was tested only on patterns bootstrapped from the corpus, recall was not evaluated.

SUMMARY

In this chapter, we observed the initial research concentrating on evaluating words to detect their sentiment orientation and subjectivity, which led to the modern automated lexical sentiment resources. The approaches used included use of linguistic phenomena such as the restrictions placed by specific conjunctions on sentiment and statistical approaches such as bootstrapping and semi-supervised machine learning. The methodology of each approach was briefly explained and the results thus obtained stated. We conclude that subjectivity detection is a crucial first step to building automated sentiment lexicons but this is a much harder problem than sentiment orientation detection alone.
The first lexicon this report will examine is Sentiwordnet. Sentiwordnet is an automatically generated lexicon from an existing resource – the Wordnet. As a result of automatic generation and use of such a large resource, Sentiwordnet has very high coverage. Moreover, its structure allows it to support not only a three-way positive vs. negative vs. objective classification, but also a graded sentiment label. All of these properties make Sentiwordnet an important lexicon for study.

This chapter is organized as follows. Section 3.1 gives an introduction to Sentiwordnet, including a brief introduction to Wordnet, the resource on which it is based. Section 3.2 then explains how Sentiwordnet was created. Section 3.3 evaluates the resultant lexicon.

3.1 INTRODUCTION

Sentiwordnet is essentially Wordnet annotated with sentiment information. As a result, Wordnet has been covered in brief before the coverage of the actual Sentiwordnet starts.

3.1.1 Wordnet

Wordnet [11] is an electronic lexical database for the English language. However, unlike most lexical resources, the fundamental unit of storage is not a word. Instead, the database is organized by word sense. The way this is implemented is by grouping together words of the same sense together in a single entity called a synset. An example synset is the 2-word synset \{brilliant, superb\}.¹

This gives rise to the structure of the Wordnet, referred to as the lexical matrix. This is a matrix with word forms as column entries and word meanings as row entries. This gives an indication of synonymy by multiple entries in the same row, and polysemy by multiple entries in the same column. The structure of the lexical matrix is shown below:

<table>
<thead>
<tr>
<th>Word Meanings</th>
<th>Word Forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>M₁</td>
<td>E₁₁, E₁₂, E₁₃, \ldots E₁ₙ</td>
</tr>
<tr>
<td>M₂</td>
<td>E₂₁, E₂₂, \ldots E₂ₙ</td>
</tr>
<tr>
<td>M₃</td>
<td>\vdots</td>
</tr>
<tr>
<td>Mₘ</td>
<td>Eₘ₁, Eₘ₂, \ldots Eₘₙ</td>
</tr>
</tbody>
</table>

¹ URL: [http://wordnetweb.princeton.edu/perl/webwn?w=brilliant](http://wordnetweb.princeton.edu/perl/webwn?w=brilliant)
Here, synset \( M_1 \) has two synonyms, \( F_1 \) and \( F_2 \), represented by entries \( E_{1,1} \) and \( E_{1,2} \) respectively, while word form \( F_2 \) is polysemous with meanings \( M_1 \) and \( M_2 \), represented by entries \( E_{1,2} \) and \( E_{2,2} \) respectively.

Wordnet synsets are linked to each other by semantic relations, which are relations between the meanings of synsets. Semantic relations can be thought of as ‘references’ from one synset to another. Some of the important semantic relations are:

- *Antonymy*, denoting opposite meaning,
- *Hypernymy*, denoting the superset relation,
- *Hyponymy*, denoting the subset relation,
- *Meronymy*, denoting the part-of relation,
- *Similar to*, denoting the similarity relation, etc.

The existence of such a well-organized lexical resource provides vital additional information in order to use in sentiment analysis besides the words themselves.

### 3.1.2 Sentiwordnet

Sentiwordnet, described first by Esuli and Sebastiani [10], is a sentiment lexicon which augments Wordnet with sentiment information. It does this by adding three sentiment scores to each synset in the Wordnet as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Pos}(s) )</td>
<td>The positive score of synset ( s )</td>
</tr>
<tr>
<td>( \text{Neg}(s) )</td>
<td>The negative score of synset ( s )</td>
</tr>
<tr>
<td>( \text{Obj}(s) )</td>
<td>The objective score of synset ( s )</td>
</tr>
</tbody>
</table>

where,

\[
0 \leq \text{Pos}(s), \text{Neg}(s), \text{Obj}(s) \leq 1
\]

\[
\text{Pos}(s) + \text{Neg}(s) + \text{Obj}(s) = 1
\]

For instance, the scores for the synset \( \text{beautiful}\#1 = \{\text{beautiful}\} \) are\(^2\):

- \( \text{Pos}('\text{beautiful}'\#1) = 0.75 \)
- \( \text{Neg}('\text{beautiful}'\#1) = 0.00 \)
- \( \text{Obj}('\text{beautiful}'\#1) = 0.25 \)

This can also be visualized as a triangle as shown in Figure 3.1:

This formulation of the sentiment values has the following salient features:

- The sentiment is now tied intimately to the *meaning* of a word rather than the word itself.
- A synset is now allowed to be both positive and negative, or neither positive nor negative.
- The sentiment evaluation is graded over a scale rather than a hard binary/ternary classification.

\(^2\) URL: [http://sentiwordnet.isti.cnr.it/search.php?q=beautiful](http://sentiwordnet.isti.cnr.it/search.php?q=beautiful)
However, the use of synsets over raw words means that in order for sentiment analysis to be performed on a document using Sentiwordnet, first Word-Sense Disambiguation needs to be performed on the document with reasonable accuracy.

3.2 Process of Creation

The process of creation of Sentiwordnet is an expansion of the approach used for the three-class sentiment classification from [9, Ch. 2] to handle graded sentiment values.

In essence, the creation algorithm created a training set by semi-supervised expansion of a seed set and then fed this training set to a team of ternary classifiers (which in turn consisted of two binary classifiers). The synsets were then assigned scores depending on the verdict of this set of ternary classifiers.

The following was the algorithm used in order to create Sentiwordnet:

1. Selection of Seed Set

The algorithm starts with a seed set $L_p$ and $L_n$ consisting of ‘paradigmatic’ positive and negative synsets respectively. These synsets were derived from the terms in the seed set used in [34]. The 14 terms gave rise to 47 positive and 58 negative synsets. Each synset was represented using the TDS$^{-}$ representation from [9, Ch. 1]

2. Creation of Training Set

This seed set was expanded for $k$ iterations using the following relations of Wordnet:

- Direct antonymy
- Similarity
- Derived from
- Pertains to
- Attribute
- Also see

These were the relations hypothesized to preserve (or in the case of direct antonymy, exactly invert) the associated sentiment. After $k$ iterations of expansion, this gave rise to the sets $T_{p}^{k}$ and $T_{n}^{k}$. 
The objective set $L_o = Tr_o^k$ was assumed to consist of all the synsets that did not belong to $Tr_p^k$ or $Tr_n^k$.

3. Selection of Learning Algorithms
Two learning algorithms were selected to form the basis for the classifiers in the set of ternary classifiers: The Rocchio Algorithm and Support Vector Machine (SVM). These particular algorithms were selected due to the difference in their treatment of prior probabilities – while SVM takes the prior distributions of the classes into account, Rocchio ignores them. As a result, the behaviour of the two algorithms is different.

4. Creation of Classifiers
A classifier can be defined as a combination of a learning algorithm and a training set. In addition to the two choices of learning algorithms, four different training sets were constructed with the number of iterations of expansion $k = 0, 2, 4, 6$. The size of the training set increased substantially with an increase in $k$. As a result, low values of $k$ yielded classifiers with low recall but high precision, while higher $k$ led to high recall but low precision.

As a result there were 8 ternary classifiers in total due to all combinations of the 2 learners and 4 training sets.

Similarly to [9, Ch. 2], each ternary classifier was actually made up of two binary classifiers, positive vs. not positive and negative vs. not negative, combined as follows:

<table>
<thead>
<tr>
<th>POSITIVE</th>
<th>NOT POSITIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEGATIVE</td>
<td>Objective</td>
</tr>
<tr>
<td>NOT NEGATIVE</td>
<td>Positive</td>
</tr>
</tbody>
</table>

5. Synset Scoring
Once the classifiers were ready, each synset from the Wordnet was vectorized and fed to the team of ternary classifiers as test input. Depending upon the output of the classifiers, each synset was assigned sentiment scores as follows:

$\text{Pos}(s) = \text{No of classifiers stating positive} / 8$
$\text{Neg}(s) = \text{No of classifiers stating negative} / 8$
$\text{Obj}(s) = \text{No of classifiers stating objective} / 8$

In this way, it was possible to annotate the entirety of the Wordnet with sentiment information using automated methods.

A summary of the sentiment of all of Sentiwordnet can be seen by averaging out all the values by part-of-speech:

<table>
<thead>
<tr>
<th>PART OF SPEECH</th>
<th>POSITIVE</th>
<th>NEGATIVE</th>
<th>OBJECTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJECTIVES</td>
<td>0.106</td>
<td>0.151</td>
<td>0.743</td>
</tr>
<tr>
<td>NOUNS</td>
<td>0.022</td>
<td>0.034</td>
<td>0.944</td>
</tr>
<tr>
<td>VERBS</td>
<td>0.026</td>
<td>0.034</td>
<td>0.940</td>
</tr>
<tr>
<td>ADVERBS</td>
<td>0.235</td>
<td>0.067</td>
<td>0.698</td>
</tr>
<tr>
<td>ALL</td>
<td>0.043</td>
<td>0.054</td>
<td>0.903</td>
</tr>
</tbody>
</table>
3.3 RESULTS

As Sentiwordnet consists of graded scores in \([0, 1]\) rather than simple labels, evaluation is no longer an easy task of comparing labels. In addition, the problem is further complicated by the fact that the sentiment scores are given to Wordnet synsets rather than words and there was no gold standard resource for sentiment annotations for synsets.

While the initial work on Sentiwordnet did not evaluate the results, the subsequent work Baccianella et al. \([2]\) solved the evaluation problem by using a special gold-standard manually annotated resource called Micro-WN (Op). Micro-WN (Op) consists of a carefully chosen 1,105-synset subset of Wordnet, manually annotated with three scores Pos(s), Neg(s) and Obj(s).

The evaluation was performed by ranking all the synsets by their positivity according to their Pos(s) scores, and similarly using Neg(s) for negativity. A similar ranking was computed for Sentiwordnet, and then the two rankings were compared using the p-Normalized Kendall \(\tau\) distance, given by

\[
\tau_p = \frac{n_d + p.n_u}{Z}
\]

(3.1)

where:
- \(n_d\) is the number of discordant, i.e. non-matching pairs,
- \(n_u\) is the number of unordered, i.e. tied pairs in the test set,
- \(p\) is the penalty for such unordered pairs set to \(p = 1/2\),
- \(Z\) is the normalization factor equal to number of examples in the gold standard.

The lower the value for \(\tau_p\), the better is the test resource. The values for Sentiwordnet were 0.281 and 0.231 for positive and negative rankings respectively, which is significantly better than the random choice mark of 0.5.

SUMMARY

This chapter covered a prominent lexical resource called Sentiwordnet. Sentiwordnet is an automatically generated high-coverage sentiment resource based on Wordnet. It makes extensive use of the information from Wordnet during the generation process; specifically the synset structure and the semantic relations across synsets. A semi-supervised expansion approach is used to create the training set, and then a team of 8 ternary classifiers varying in the learning algorithm and the training set is used to classify every Wordnet synset in order to obtain a set of 3 graded sentiment scores per synset, one each being the positive, negative and objective score.
In direct contrast to Sentiwordnet which is a high-coverage automatically generated lexical resource, the Sentiment Orientation CALculator (SO-CAL) system is based on a manually constructed low-coverage resource which is made up of raw words rather than synsets. The only major similarity between Sentiwordnet and SO-CAL is support for graded sentiment scores, but even the ways in which this is achieved differ widely. The salient feature of SO-CAL is that the information which is lost through not using the semantic relations of Wordnet is made up for in the sheer number of ‘features’ in which the words have been grouped. It is instructive to compare and contrast these two disparate approaches to sentiment lexicon creation.

This chapter is organized as follows. Section 4.1 gives an introduction to SO-CAL. Then Section 4.2 explains the resource in a bit more depth, giving an account of the various features used. Finally Section 4.3 evaluates SO-CAL in various ways.

### 4.1 Introduction

The latest and most improved version of the SO-CAL system has been documented in Taboada et al. [33]. SO-CAL uses as its basis a lexical sentiment resource consisting of about 5,000 words. (In comparison, Sentiwordnet has over 38,000 polar words and several other strictly objective words.) Each word in SO-CAL has a sentiment label which is an integer in \([-5, +5]\) apart from 0 as objective words are simply excluded. The strengths of SO-CAL lie in its accuracy, as it is manually annotated, and the use of detailed features that handle sentiment in various cases in ways conforming to linguistic phenomena.

### 4.2 Features Used

SO-CAL uses several ‘features’ to model different word categories and the effects they have on sentiment. In addition, a few special features operate outside the scope of the lexicon in order to affect the sentiment on the document level. This section gives an account of the various features of SO-CAL.

#### 4.2.1 Adjectives

Adjectives have generally been the focus of sentiment analysis as has been observed in Chapter 2 as per several works [12] [34] [16]. As such, adjectives are also the initial focus of SO-CAL. A manual dictionary of adjectives was created by manually tagging all adjectives in a 500-document multidomain review corpus, and the terms from the General Inquirer dictionary were annotated added to the list thus obtained. All sentiment annotation was on a scale of \([-5, +5]\) with objective words removed from the list. This led to a total of 2,252 adjectives being annotated.
For example, the word good has a sentiment label of +3. If a word has multiple senses, its sentiment value was obtained by averaging out the values for all the senses weighted by sense frequency. For example, inspired has a sentiment of +2 because of the conflicting senses of “heighten or intensify” which is a positive sense, “supply the inspiration for” which is an ambiguous sense (may relate to both homage or plagiarism) and “serve as the inciting cause of” which is a predominantly negative sense1.

4.2.2 Nouns, Verbs and Adverbs

Most sentiment analysis research concentrates on adjectives as they are hypothesized to carry sentiment more than other parts of speech. However, nouns, verbs and adverbs also carry a lot of important sentiment information as can be seen in the following examples:

```
NOUN: Stupendous trash(-) masquerading as a book...
VERB: I loved(+) the movie!
ADVERB: What a badly(-) produced track!
```

To remedy this lacuna, SO-CAL extended the approach used for adjectives to nouns and verbs. As a result, 1,142 nouns and 903 verbs were added to the sentiment lexicon. Adverbs were added by simply adding the -ly suffix to adjectives and then manually altering words whose sentiment wasn’t preserved, such as essentially.

In addition multi-word expressions were also added, leading to an addition to 152 multiwords in the lexicon. Thus, while the adjective funny has a sentiment of +2, the multiword act funny has a sentiment of −1.

4.2.3 Intensifiers and Downtoners

An Intensifier is a word which increases the intensity of the phrase to which it is applied, while a Downtoner is a word which decreases the intensity of the phrase to which it is applied. For instance the word extraordinarily in the phrase “extraordinarily good” is an intensifier while the word somewhat in the phrase “somewhat nice” is a downtoner.

Research work such as [17] has usually implemented intensifiers and downtoners by creating a dictionary of known intensifiers and downtoners and then using fixed addition and subtraction of the sentiment values respectively. However, this misses out on the wide range within intensifiers and downtoners. Moreover, if an intensifier is applied to an already intense sentiment, the intensification should be more than if it were applied to a lukewarm sentiment. Similar logic applies to downtoners. For instance, extremely is a much stronger intensifier than very. Moreover, the intensification contributed by extremely in the phrase “extremely fantastic” is more than in the phrase “extremely good”.

To remedy this, SO-CAL implements intensifiers and downtoners as percentage modifiers that act on the phrase which they are modifying. The percentage

---

1 URL: http://wordnetweb.princeton.edu/perl/webwn?s=inspired
values for intensifiers are positive while for downtoners the values are negative. For example, extraordinarily is an intensifier with value $+50\%$ while somewhat is a downtoner with value $-30\%$. Thus, the final sentiment of the phrase "extraordinarily good" is $3 + (3 \times 0.50) = +4.5$, while the sentiment of "somewhat good" is $3 - (3 \times 0.30) = +2.1$.

### 4.2.4 Negation

Negation is another vital phenomenon as due to negation a single word can invert the polarity of the entire sentence. This is especially vital for sentiment analysis using lexical resources because classifier-based methods find it very difficult to deal with negation, as seen in Section 1.2. Most research, such as [29] contends that it is intuitive to model negations as outright flips in polarity, referred to as switch negation, i.e. the sentiment score for "not good" will be $-3$. However, even though this approach seems reasonable at first glance, it fails to account for several situations. For example, the phrase "not excellent" will be more negative than "not good" ($-5$ vs $-3$) even though "not excellent" is actually partly positive. Similarly, "not atrocious" gets a higher positive score than "good".

Instead, SO-CAL implements negation as a numerical shift from the current sentiment by a fixed amount towards the opposite orientation. This approach is referred to as shift negation. Thus, if the current sentiment is positive, the negation subtracts a fixed number from the sentiment. Otherwise it adds a fixed number. The number has been set to 4. As a result, "not good" gets a sentiment value of $3 - 4 = -1$, which is negative as compared to "not excellent" with a sentiment value of $5 - 4 = +1$.

### 4.2.5 Irrealis Blocking

An irrealis is a word that indicates that the sentiment of the sentence to which the word is applied may not be reliable, because the event spoken about in the sentence has not actually occurred. For instance, in the sentence "You'd expect such a basic concept to be implemented correctly", the word expect is an irrealis marker because its addition indicates that the event was expected to happen but hasn’t actually happened. This makes the sentiment of the sentence unreliable.

The following are the irrealis markers identified by SO-CAL:

- Modals (could, would, etc.)
- Conditional Markers (if)
- Negative polarity items (any, anything)
- Certain verbs (expect, doubt)
- Questions
- Words enclosed in quotes

SO-CAL deals with irrealis markers by simply ignoring the sentiment content of the sentence in which it appears, i.e. the sentiment of the entire sentence.
is set to 0. This is because while an irrealis marker indicates that the sentiment of the sentence is not valid as it is, it does not give any further information.

4.2.6 Text-level Features

While SO-CAL has several features that operate on the word-level using the associated lexical sentiment resource, it also has other features that operate on the text-level, i.e., on the entire document. These features modify the sentiment of the document outside the scope of the resource. However, the values used ultimately come from the resource itself. This section shows that the way a sentiment lexicon is used is as important as the way it is created.

SO-CAL has the following text-level features:

1. Negation Weighting
   As has been observed in [4], humans are predisposed towards using positive language. As a result, it is not advisable to give equal importance to positive and negative sentiments as it would lead to a strong bias towards positivity. To deal with this problem, SO-CAL implements negation weighting by increasing each negative sentiment score by 50%. This compensates for the inherent bias towards positivity.

2. Repetition Weighting
   If a word is repeated multiple times in a sentence, it suggests that the speaker has a lack of conviction about what is being said and is instead using a word to indicate sentiment just because a word has to be used. Moreover, it has been hypothesized that an increase in frequency of a word leads to a reduction in its intensity (For an example, see [37]). To handle this, repetition weighting is introduced. The \( n^{th} \) occurrence of a word gets only \( 1/n \) of its sentiment score. As a result, the sentiment intensity reduces with increased repetition.

4.3 Results

SO-CAL has been evaluated comprehensively on various fronts. Each and every choice taken while creating SO-CAL has been examined and the configuration presented has emerged as the best option out of all current work on the topic. A summary of the results is presented in this section.

4.3.1 Evaluation of the Lexicon

This was a test to determine if the structure of the lexicon, including nouns, verbs, adverbs and multiwords with a graded sentiment score was really the best performing version. Four different configurations of the lexicon were tested:

- **Simple**: Only +2 or −2 values with switch negation and +1 or −1 intensification.
- **Only-Adj**: Only adjectives.
- **One-Word**: Only single words, no multiword expressions.
• **FULL:** The entire lexicon as outlined here.

The tests were performed on a total of 5,100 documents out of which 4,700 were entirely new test documents (the other 400 were used during lexicon creation). The results are as follows:

<table>
<thead>
<tr>
<th>Dictionary Setting</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>75.11</td>
</tr>
<tr>
<td>Only-Adj</td>
<td>73.93</td>
</tr>
<tr>
<td>One-Word</td>
<td>78.23</td>
</tr>
<tr>
<td>Full</td>
<td>78.74</td>
</tr>
</tbody>
</table>

This shows that the graded labels and all the parts of speech are definitely required and multiwords are helpful.

### 4.3.2 Evaluation of SO-CAL’s Features

This was a test to determine if all the features used in SO-CAL such as shift negation, irrealis blocking, etc. were indeed necessary. The following combinations of configurations for the features were tested on the same 5,100 document corpus:

1. Only all words
2. All words + shift negation
3. All words + shift negation + intensification
4. All words + shift negation + intensification + irrealis blocking
5. All words + shift negation + intensification + irrealis blocking + negation weighting
6. All words + shift negation + intensification + irrealis blocking + negation weighting + repetition weighting
7. All words + switch negation + intensification + irrealis blocking + negation weighting + repetition weighting

The results are as follows (see above list for the Configuration Number field):

<table>
<thead>
<tr>
<th>Configuration Number</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>66.04</td>
</tr>
<tr>
<td>2</td>
<td>68.35</td>
</tr>
<tr>
<td>3</td>
<td>71.35</td>
</tr>
<tr>
<td>4</td>
<td>72.66</td>
</tr>
<tr>
<td>5</td>
<td>77.32</td>
</tr>
<tr>
<td>6</td>
<td>78.74</td>
</tr>
<tr>
<td>7</td>
<td>78.37</td>
</tr>
</tbody>
</table>

These results show that the full feature set outlined in this chapter corresponding to configuration number 6 (All words + shift negation + intensification + irrealis blocking + negation weighting + repetition weighting) is the best performing feature set.
4.3.3 Evaluation across domains

This was a test to evaluate SO-CAL’s performance across domains. This was to validate the hypothesis that lexicon-based methods can scale better across domains as compared to classifier-based methods. These tests were performed on a 400-document subset of the 5,100-document corpus evenly distributed across 8 domains. The subset used was a part of the previously unseen subset of the corpus. The results are as follows:

<table>
<thead>
<tr>
<th>Domain</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Books</td>
<td>74</td>
</tr>
<tr>
<td>Cars</td>
<td>78</td>
</tr>
<tr>
<td>Computers</td>
<td>90</td>
</tr>
<tr>
<td>Cookware</td>
<td>78</td>
</tr>
<tr>
<td>Hotels</td>
<td>76</td>
</tr>
<tr>
<td>Movies</td>
<td>78</td>
</tr>
<tr>
<td>Music</td>
<td>82</td>
</tr>
<tr>
<td>Phones</td>
<td>84</td>
</tr>
<tr>
<td>Overall</td>
<td>80</td>
</tr>
</tbody>
</table>

This shows that SO-CAL performs reasonably well across domains without too many fluctuations in performance.

4.3.4 Evaluation against other lexicons

This was a test to evaluate the lexicon itself, using other lexicons as a comparison. This test was necessary to validate the words in the lexicons themselves, and more importantly, the choice of using a manual lexicon over an automatically generated one. The following were the lexicons used for the comparison:

- The Google-generated PMI-based dictionary, from [32]
- The Maryland dictionary consisting of 70,000+ words, from [22]
- The manually constructed General Inquirer, documented in [24]
- The automatically constructed Subjectivity Lexicon, from [35]
- Sentiwordnet, from [10]
- The SO-CAL lexicon

The tests were performed on the 5,100-document corpus. The following are the results:

<table>
<thead>
<tr>
<th>Lexicon</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>62.98</td>
</tr>
<tr>
<td>Maryland</td>
<td>62.65</td>
</tr>
<tr>
<td>General Inquirer</td>
<td>68.02</td>
</tr>
<tr>
<td>Sentiwordnet</td>
<td>65.02</td>
</tr>
<tr>
<td>Subjectivity Lexicon</td>
<td>72.04</td>
</tr>
<tr>
<td>SO-CAL</td>
<td>78.74</td>
</tr>
</tbody>
</table>
These results show that SO-CAL performs better than any other dictionary considered, in spite of being rather small at only 5,000 words (Sentiwordnet has over 30,000 polar words and the Maryland dictionary has over 70,000 words). These results indicate that accuracy is more important than mere coverage.

SUMMARY

This chapter covered a sentiment analysis system called SO-CAL based on a sentiment lexicon. The lexicon was created by annotating words manually. The words, about 5,000 in all, include adjectives, nouns, verbs, adverbs and multiwords. Each word is graded as an integer on a scale of \([-5, +5]\). In addition to just the words, several other features were added to capture linguistic phenomena such as negation, intensification, irrealis markers, etc. Evaluation tests performed using SO-CAL show that the structure of the lexicon and the feature set is the best performing one out of all considered possibilities. Moreover, SO-CAL scales well across domains and performs better than any other state-of-the-art sentiment lexicon including automatically generated ones with much higher word counts.
Wordnet-Affect is an automatically generated sentiment resource backed by Wordnet, not much differently from Sentiwordnet (Chapter 3). However, what makes Wordnet-Affect different is the way the sentiment of each synset is represented. Rather than just a sentiment lexicon, Wordnet-Affect is an affective lexicon, i.e. it associates each synset with information relating to its ‘affect content’ or ‘emotional content’. As a result, Wordnet-Affect has much more information about a word than just a sentiment label and thus can be used for a wider range of applications. In addition, representing affect information in a lexicon added additional complexities to the automatic generation process.

5.1 Introduction

Wordnet-Affect was presented by Strapparava and Valitutti [31]. It has been originally designed for affective computing i.e. giving computing machines the ability to perceive, process and express emotions (Picard [27]). It adds affective information to a select set of synsets of Wordnet. Such a resource is necessary due to the following reasons:

- A haphazard process of listing down emotion words in general affect literature has led to the inclusion of not only emotions but also moods, cognitive states, etc. as emotion words.
- There is agreement on only three main affective features – valence, arousal and motivational intensity. These are not enough to uniquely classify every emotion.
- Most affective computing approaches work on the term level and are ignorant of word senses.

In addition, Wordnet-Affect is also necessary for sentiment analysis. This is because each emotion has a valence associated with it, which is basically the sentiment of the emotion. As a result, affective computing solves a broader problem than sentiment analysis. However, sentiment analysis can also be helped by the identification of the emotion category behind the sentiment. For instance, sadness can be handled differently from shame even though both have negative sentiments.

5.2 Structure

Wordnet-Affect associates affective information with each synset. This information can be considered to be a 2-tuple of (a-label, valency). Here, the a-label describes the type of emotion while the valency describes the sentiment associated with it.

a-labels are organized in a hierarchical manner in order to classify all affective concepts into groups. Each path in the hierarchy starts with a special
node called root and ends with a synset describing the entire hierarchy as a leaf node. The valency of each synset can have one of four possible values: positive, negative, neutral, ambiguous.

A part of the Wordnet-Affect hierarchy is shown in Figure 5.1.

In addition, synsets with different parts of speech for the same concept are linked using the POS-R relation, standing for 'Part-of-Speech Reference'. For instance, the verb cheer, the adjective cheerful, the phrase cheer up and the noun cheerfulness are all linked by POS-R.

5.3 PROCESS OF CREATION

Like Sentiwordnet, Wordnet-Affect progressed through two stages of creation – the creation of the seed set, followed by automatic expansion using Wordnet relations. However, in the case of Wordnet-Affect, the process of creation of the seed set was non-trivial as the amount of information that needed to be represented was much more than just a single triplet of numbers as in Sentiwordnet.

The two stages in Wordnet-Affect’s creation are explained below.

1. CORE RESOURCE (MANUAL)

In order to function as a seed for expansion using Wordnet, a core resource called Affect was created. Affect consisted of 1,903 terms. Each term was manually annotated with affect information and the a-label hierarchy was built. Then the information from these terms was projected on to Wordnet synsets to create a combination of Wordnet and Affect, forming the base for Wordnet-Affect.

2. EXPANSION (SEMI-AUTOMATIC)

Once the base resource for Wordnet-Affect was created, automatic expansion was performed using the following semantic relations of Wordnet:
• Direct antonymy
• Similarity
• Derived from
• Pertains to
• Attribute
• Also see

These relations were hypothesized to preserve the affect information associated with the synsets. (Direct antonymy was assumed to preserve the a-label but invert the valency.) Expansion was also performed for other relations but the result had to be filtered manually. For example, the hypernym of the synset cheer is the synset attribute, which does not preserve the affect information associated with cheer.

SUMMARY

This chapter covered Wordnet-Affect which, like Sentiwordnet, is an automatically generated resource that uses Wordnet for expansion. While it contains information related to affect, it can also be used for affect-oriented sentiment analysis due to the presence of sentiment information in the form of valence. The affect information is organized in the form of a hierarchy of a-labels. The process of building Wordnet-Affect consisted of creation of a large manual resource that established the a-label hierarchy, followed by automatic expansion using the semantic relations of Wordnet.
Work at IIT-Bombay across all of NLP has concentrated heavily on adapting existing NLP techniques to multilingual settings for Indian languages. There is a lack of NLP research for Indian languages which is a problem as Indian languages present a unique set of challenges for NLP (including, for example, Marathi’s tendency to combine words as in घराणाच्या स्वामीर or the rich morphology of Dravidian languages). Sentiment analysis is no different.

This chapter covers two key developments at IIT-Bombay towards sentiment analysis resources. This concentrates on adapting sentiment analysis techniques and existing resources to multiple languages and covers both sentiment lexicons and auxiliary resources which can help sentiment analysis.

The organization of this chapter is as follows. Section 6.1 covers Indian-language Sentiwordnets, specifically the Hindi Sentiwordnet which was created by projecting the sentiment scores of the English Sentiwordnet on Hindi synsets. Section 6.2 covers use of linked wordnets to help cross-language sentiment analysis, including an approach to automate the process of wordnet linking.

### 6.1 Indian-Language Sentiwordnets

As a first step towards Indian language sentiment analysis, Joshi et al. [14] touched upon three possible approaches to perform sentiment analysis in Hindi. These, while specified for Hindi, are applicable to any language in general, and resource-constrained Indian languages in particular. They are:

- Performing in-language sentiment analysis using classifier methods and a Hindi corpus,
- Performing machine translation of a Hindi document to perform sentiment analysis on an English corpus,
- Creating a sentiment resource in Hindi to use for sentiment analysis.

Towards creation of an Indian-language sentiment resource, the Hindi Sentiwordnet\(^1\) was created in Joshi et al. [14]. This section examines it and compares it to the other two approaches for Indian language sentiment analysis.

#### 6.1.1 Introduction

There are two possible options for developing a sentiment resource in Hindi:

- Create a new resource from scratch, or
- Convert an existing sentiment resource to Hindi.

\(^1\) Available for download at [http://www.cfilt.iitb.ac.in/Sentiment_Analysis_Resources.html](http://www.cfilt.iitb.ac.in/Sentiment_Analysis_Resources.html)
While the first approach increases the likelihood of the resultant resource actually working for the new language, adapting an existing resource from another language is an attractive approach due to several reasons. Primarily, this is because the difficult work of actually coming up with the lexicon has already been done. Moreover, the way the lexicon has been formulated (three sentiment scores in [0, 1] in the case of Sentiwordnet) is already tried and tested and is known to work. As a result, the only concern is adaptation of the lexicon to another language.

The Hindi Sentiwordnet was, thus, created by converting the English Sentiwordnet to Hindi. This was made possible because the synsets of the English Wordnet have direct links to corresponding Hindi synsets in the Hindi Wordnet, as a result of Indowordnet (Bhattacharyya [3]). Indowordnet is an ongoing effort at IIT-Bombay that aims at creating a linked wordnet structure for 16 Indian languages, which is, in turn, being linked to the Princeton English Wordnet.

6.1.2 Process of Creation

The Hindi Sentiwordnet was created by linking each synset in Sentiwordnet (ver. 1.1) to its corresponding Hindi synset from Indowordnet. This process was carried out in the following steps:

1. Extract a synset $S_e$ from Sentiwordnet along with its corresponding sentiment scores.

2. Using Indowordnet, find the corresponding linked synset $S_h$ for Hindi.

3. If $S_h$ exists, create an entry in the Hindi Sentiwordnet for $S_h$ with sentiment scores same as $S_e$.

This led to creation of Hindi Sentiwordnet with about 16,000 synsets. For instance, the single-word synset for अनुकूलीय has sentiment values of $\text{Pos}(s) = 0.5$, $\text{Neg}(s) = 0.125$ and $\text{Obj}(s) = 0.375$.

6.1.3 Evaluation

The Hindi Sentiwordnet was evaluated for a sentiment analysis task alongside the other two approaches of in-language sentiment analysis and machine translation. The Hindi corpus used was a 250-document manually annotated movie review corpus. In addition, the 2,000-document movie review corpus from [26] was used for the machine translation approach.

The following are the best-performing accuracies for the three approaches:

<table>
<thead>
<tr>
<th>Approach</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-language Sentiment Analysis</td>
<td>78.14</td>
</tr>
<tr>
<td>Hindi Sentiwordnet</td>
<td>60.31</td>
</tr>
<tr>
<td>Machine Translation</td>
<td>65.96</td>
</tr>
</tbody>
</table>

The accuracy for Hindi Sentiwordnet is disappointingly low as compared to the other approaches, particularly standard in-language classifier-based sentiment analysis. However, the following are possible reasons for this:
6.2 Wordnet-linking for Cross-Language Sentiment Analysis

As seen in Section 6.1, in order to build a sentiment resource using Wordnet Projection, it is necessary to have wordnets in two languages linked with each other. This can also help corpus-based approaches as sense-annotated corpora in one language can be used to automatically sense-annotate corpora in another language. This is particularly useful as use of sense features is known to improve performance [15].

Khapra et al. [18] cover Wordnet linking, including a short survey of approaches used. In particular, a special linked wordnet structure called Multidict has been covered. In addition, a probabilistic cross-linking approach is proposed to automate the process of wordnet linking which is otherwise time and effort intensive. This section covers these two approaches in brief.

6.2.1 Multidict

The Multidict is a synset-based multilingual dictionary proposed by Mohanty et al. [23] working at IIT Bombay. This is a novel structure which allows linking of multiple wordnets by creating an entry for each synset and storing Wordnet entries for every individual wordnet corresponding to the synset. However, the novelty of this approach comes from the fact that within each synset link, the corresponding words are also linked. This allows Multidict to function as a bilingual or multilingual dictionary.

The advantage of such a structure for corpus-based sentiment analysis is that not only does synset projection become much easier, such a resource also
allows for use of a corpus in one language for sentiment analysis in another language, even if none of the corpora is sense-marked. This is made possible due to the word-level linking and is helpful for sentiment analysis in languages where resources are not readily available.

In addition, the advantage of Multidict for lexicon-based sentiment analysis is that projection can be carried out not only for synset-level resources, but also for word-level resources, again due to Multidict’s word-level linking. As a result, no explicit machine translation would be required to translate lexical sentiment resources from one language to another.

The disadvantage of this approach is that it is extremely time consuming. Not only must every synset in at least two wordnets be linked, the words in the synsets must also be explicitly linked. This approach also faces several linguistic challenges.

6.2.2 Probabilistic Cross-Linking of Wordnets

In order to automate the process of Wordnet linking on the word level, Khapra et al. [18] proposed a probabilistic approach to Wordnet linking. This approach assumes that, given a sense-level linked Wordnet between source language $L_1$ (say, Hindi) and target language $L_2$ (say, Marathi), every word in $L_2$’s synset is cross-linked to any word in $L_1$ with some probability.

This probability can be computed for a word $W$ of language $L_2$ corresponding to linked synset $S$ using two corpora in languages $L_1$ and $L_2$ out of which $L_1$ has sense annotation. The formulation is given by:

$$E[c(S, W)] = \sum_{h_i \in \text{crosslinks}} P(h_i | W, S) \ast c(S, h_i)$$

Here, $E[c(S, W)]$ is the expected count of the $L_2$ word $W$ in the sense $S$, $P(h_i | W, S)$ is the probability of the $L_1$ word $h_i$ given that the sense $S$ is used, and $c(S, h_i)$ is the count of the number of times the $L_1$ word $h_i$ is used for sense $S$.

The $L_1$ word $h_i$ is assumed to be independent of the $L_2$ word $W$. Thus the following assumption can be made:

$$P(h_i | W, S) = P(h_i | S)$$

As a result, the equation now reduces to

$$E[c(S, W)] = \sum_{h_i \in \text{crosslinks}} P(h_i | S) \ast c(S, h_i)$$

This can easily be found from the two corpora.

As a result, word-level linking can be obtained without creating a Multidict, thus saving a lot of time and cost.

SUMMARY

This chapter introduced the Hindi SentiWordNet developed at IIT-Bombay using projection from the English SentiWordNet through Indowordnet links. It
then introduced efforts on word-level Wordnet linking in IIT-Bombay, including a novel manual approach using the Multidict, and an automated probabilistic approach using corpora.
CONCLUSIONS AND FUTURE WORK

This chapter concludes, summing up the major takeaways of the report including some observations. The chapter is organized as follows. Section 7.1 dwells on the major ways in sentiment lexicons can be created and the pros and cons of each. Section 7.2 concentrates on a major design choice – how should the sentiment be represented. Finally Section 7.3 puts forward ideas on possible future work in this area.

7.1 APPROACHES OF BUILDING SENTIMENT LEXICONS

Three major approaches to building sentiment lexicons were encountered over the course of this report. We now examine them in a nutshell.

7.1.1 Manual

The manual approach to creation of sentiment lexicons concentrates on using purely manual efforts to annotate a set of words with their sentiment. The words to be annotated could come from any source such as a corpus or an existing dictionary, but apart from that, no external influences or automated mechanisms are used. SO-CAL, covered in Chapter 4, is an example of a lexicon created using a manual approach.

The major advantage of this approach is that it promises high accuracy as there is no statistical algorithm which could go wrong. In order to further bolster this method, multiple annotators can be asked to annotate the same words and the inter-annotator agreement can be calculated.

The major disadvantage of this approach is the time investment required to come up with a sizeable resource (an estimate was worked out in Section 1.3). As a result, the lengths of manually created resources are severely restricted.

7.1.2 Automatic

The automatic approach first creates a seed set of known sentiment-bearing words and then may use either a pre-existing resource such as Wordnet or a corpus to automatically expand the seed set to create a larger set. This set can then either function as the final lexicon, or it can be used as a training set for a statistical mechanism to collect further data. Sentiwordnet is an example of an automatically generated sentiment resource.

The advantages of this approach are that the time investment required is very less in comparison to manual approaches. The only work that needs to be put in is running the algorithm once and, as a result, the most time intensive part of the process is coming up with a suitable algorithm in the first place. In addition, this approach allows use of existing resources to create the lexicon, thus allowing reuse of both the resource and techniques used to deal with it.
The major disadvantage of this approach is that there are several ways in which the statistical algorithm used to generate the resource may be misled, leading to incorrect entries in the resource. For example, the synset for the adverb *beastly*#1 = {beastly} has the following values in Sentiwordnet: Pos(s) = 0.25, Neg(s) = 0, Obj(s) = 0.75.\(^1\)

### 7.1.3 Projection

The projection approach takes an existing sentiment resource in one language and maps it to another language.

This approach is very useful for creating resources in non-English languages without repeating the entire process of resource creation. Moreover, the base resource can be either manually or automatically generated, or any combination thereof, providing for immense flexibility.

The major disadvantage of this approach, as outlined in Section 6.1, is that there needs to be some form of linking between the two languages. For instance, while using synsets, the Wordnets of the two languages should be linked, which is an immense effort. Moreover, there may be some concepts which exist in one of the two languages but do not map to any concept in the other due to differences in culture, geography, etc. Lastly, a major roadblock is that if the projection process is in itself statistical (for example, using machine translation for individual words), the entries in the resultant lexicon may be inaccurate.

### 7.2 Choices for Representing Sentiment

Apart from the choice of the lexical unit to use (which hasn’t seen much work till date), the major design choice in a sentiment lexicon is how to represent the sentiment itself. A few variations have been observed in most major work and this report itself has covered some of them. A summary follows.

#### 7.2.1 Only Labels

All of the work covered in Chapter 2 concentrated on methods that could eventually be used to create sentiment resources with words being annotated by only hard labels of *positive*, *negative* or optionally *objective*. While this is algorithmically easier, a lot of sentiment information is lost in the process due to lack of any scale within the sentiment categories. Furthermore, the apparent success of more detailed resources like Sentiwordnet and SO-CAL established that use of mere labels is a poor approach to sentiment representation.

#### 7.2.2 Graded Score

Most sentiment resources today use a graded representation for sentiment. For instance, Sentiwordnet assigns a set of three values, each representing the extent to which a Wordnet synset is *positive*, *negative* and *objective* respectively.

---

1 URL: [http://sentiwordnet.isti.cnr.it/search.php?q=beastly](http://sentiwordnet.isti.cnr.it/search.php?q=beastly)
On the other hand, SO-CAL assigns a single value which gives an indication on a sentiment scale.

The major obvious advantage of creating a resource with graded sentiment scores is that if the scores are not required, they can always be collapsed into a label scheme. For instance, in the case of Sentiwordnet, each synset can be assigned a label representing the maximum sentiment score. However, the reverse is obviously not true. Moreover, this representation is truer to the wide variety of intensity found in language.

7.2.3 Affect Information

Another approach, followed by Wordnet-Affect, is encoding the emotional information of a synset in the lexicon in addition to raw sentiment. This provides information about the source of the sentiment in addition to the sentiment itself which may be useful in certain settings.

7.3 Future Work

This section details some possible ideas for future work on sentiment resources. It includes entirely new ideas which have not been spotted in literature yet, and adaptation of currently existing work to new settings.

7.3.1 Automatic Resources for Higher Lexical Units

While a lot of research has concentrated on various representations of sentiment, the possibility of using different lexical units in sentiment lexicons has not been explored yet. This is an unfortunate lacuna, because given that sentiment analysis is a high-level NLP problem combining elements of pragmatics and discourse, information from all the core NLP layers can be assumed to be freely available. This idea has also been inspired by work in Machine Translation, which started at word-level alignments [6] but has now moved to phrase [20], factor (word + features) [19] and tree [36] levels.

Taking this into account, more research needs to be performed on sentiment resources which assign sentiment values to either a phrase, a factor, or a tree. In turn, a phrase may be either a linguistic entity or a sequence of words (both of which are approaches used in Machine Translation). A factor could be composed of a wide variety of word features. A tree could be hierarchical in nature. Initial efforts on a tree-based sentiment resource have been made by Socher et al. [30].

Lexicon creation can be performed either fully automatically, or in a two-stage process – first all the entries in the lexicon can be extracted automatically, and then each of them can be annotated manually.

A problem that can be foreseen with this approach is that sentiment analysis of a document would need a pre-processing step in order to represent the document in a form suitable for the resource. For example, for tree-based sentiment analysis, the entire document will have to be parsed.
7.3.2  Manual Resources on Sense Level

While SO-CAL is an exceedingly thorough resource, implementation of one aspect seems to be unsatisfactory – handling of words with multiple senses. There may be some merit in attempting to create a fully manual resource at the sense level. Not only will this improve the granularity of information available in the resource and combine synonymous words together, it will also improve the quality of the annotations themselves, as the human annotator is likely to perform better when presented with a single unambiguous sense to annotate.

In addition, if this resource becomes a reality, it would facilitate a fair comparison with Sentiwordnet, which may be a major step in settling the manual vs. automatic resource debate once and for all.

7.3.3  Sentiwordnet Projection for Other Indian Languages

The Hindi Sentiwordnet that was created using the projection technique had two pre-requisites – existence of an English sentiment resource (fulfilled by Sentiwordnet) and a linking between the two languages (partially fulfilled by Indowordnet). Due to Indowordnet, these conditions are also fulfilled for a large number of other Indian languages, and therefore, it would be wise to create Sentiwordnets for these other languages too.

7.3.4  Manual Sentiment Lexicons for Indian Languages

There also seems to be merit in creating manually generated resources for Indian languages in order to compare the performance of manual vs. automatic approaches.

7.3.5  Techniques for Incorporating Slang

In the ‘Internet Age’, sentiment analysis is assuming a lot of importance in the social network arena. However, there are no sentiment resources that concentrate directly or indirectly on ‘netspeak’ or slang which is used very commonly on the internet. A sentiment resource which exclusively contains slang expressions and their sentiments would be very helpful in solving the problem. In addition, such a resource would also be beneficial for classifier approaches because they fail at identifying slang effectively too. If a classifier detects a slang expression, the feature value could be replaced in some way by the sentiment annotation from the resource.

This presents a unique challenge as slang is ever-changing, and therefore, a manual approach is unfeasible. Instead, a ‘dynamic lexicon’ might be necessary. A possible approach is to crawl the internet looking for popular slang terms and bootstrap the sentiment out of them. Another possibility is to have a ‘time-versioned’ lexicon to keep track of how sentiments associated with slang terms change over time on the internet. (An example is “LOL” – when the term was first used its sentiment was overwhelmingly positive but today the sentiment is close to neutral due to abuse of the term.) This, when combined with information on when the document was created, will give added insight
on the author’s sentiments when the document was written as opposed to the reader’s perception when the document is being read.

SUMMARY

This chapter concludes the report, first by concentrating on two major issues in creation of sentiment lexicons and finally suggesting several ideas for future work. The first issue was the method used to create the sentiment lexicon, which can be either of manual annotation, automatic generation, or cross-language projection. The second issue was representation of sentiment, which can be performed using either only the labels, graded scores, or additional information about the source of the sentiment such as affect. Finally, the ideas pitched forward for possible work included creation of sentiment lexicons using higher levels of NLP, creation of manual sentiment lexicons at the sense level, creating projected resources and manual resources for Indian languages, and incorporation of internet slang in sentiment lexicons.


[22] Saif Mohammad, Cody Dunne, and Bonnie Dorr. Generating high-coverage semantic orientation lexicons from overtly marked words and a


