A Survey of Sentiment Lexicons

Sagar Ahire
Computer Science and Engineering
IIT Bombay
sagarahire@cse.iitb.ac.in

Abstract

This is a survey paper that introduces sentiment lexicons and explains the state of the art in the field of sentiment lexicons. Different kinds of lexicons are covered, varying in aspects such as coverage, methods of creation, lexical unit and granularity. It aims at giving a representative sampling of the field of sentiment lexicons.

1 Introduction

Sentiment Analysis is one of the most important applications of Natural Language Processing. It refers to the study of extraction of opinions from text. There are two approaches to Sentiment Analysis – the *classifier-based approach*, which treats Sentiment Analysis as a special case of text classification and uses standard Machine Learning techniques to solve the problem and the *lexicon-based approach*, which uses sentiment lexicons – dictionaries of words with labels specifying their sentiments – to identify the sentiment of text. Both approaches have their advantages and drawbacks.

This report presents the lexicon-based approach to sentiment analysis. To that end, it describes the current state-of-the-art in sentiment lexicons.

While the classifier-based approach is the dominant approach towards sentiment analysis in literature, its performance is not up to the mark. For instance, [9] 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% [7].

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 [3] 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 [4]. 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.

2 Sentiment Lexicons

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

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.

2.1 Approaches for Creation of Sentiment Lexicons

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

2.1.1 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 `(lexical unit, 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 estinate 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.

2.1.2 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.

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.

3 SO-CAL

The *Sentiment Orientation CALculator (SO-CAL)* system is based on a manually constructed low-coverage resource which is made up of raw words. There is no sense information associated with a word. 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.

The latest and most improved version of the SO-CAL system has been documented in [12]. 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.

3.1 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.

1. **Adjectives**
   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 $+3$.

2. **Nouns, Verbs and Adverbs**
   SO-CAL also 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.

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.

   SO-CAL implements intensifiers and downtoners as *percentage modifiers* that act on the phrase which they are modifying. The percentage 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 $3+(3*0.50) = +4.5$, while the sentiment of *"somewhat good"* is $3-(3*0.30) = +2.1$.

4. **Negation**
   Most research, such as [10] 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 $5 - 4 = +1$.

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 occured. 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.

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.

SO-CAL has the following text-level features:

- **Negation Weighting**: SO-CAL implements *negation weighting* by increasing each negative sentiment score by 50%. This compensates for the inherent bias towards positivity [2].
- **Repetition Weighting**: SO-CAL implements *repetition weighting* by giving the n^{th} occurence of a word only $1/n$ of its sentiment score. As a result, the sentiment intensity reduces with increased repetition.

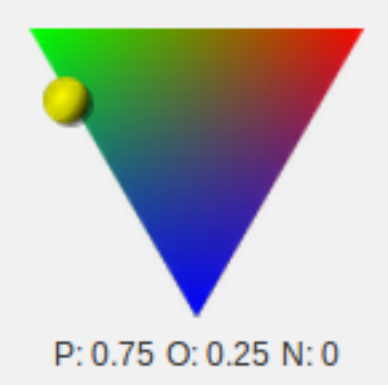

Figure 1: Visualization of synset `beautiful#1` in Sentiwordnet

4 SentiWordnet

Sentiwordnet, described first by [6], 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:

$Pos(s)$: The positive score of synset s
$Neg(s)$: The negative score of synset s
$Obj(s)$: The objective score of synset s
where,
$0 \leq Pos(s), Neg(s), Obj(s) \leq 1$
$Pos(s) + Neg(s) + Obj(s) = 1$

For instance, the scores for the synset `beautiful#1` = {`beautiful`} are[1]:
$Pos(\texttt{beautiful\#1}) = 0.75$
$Neg(\texttt{beautiful\#1}) = 0.00$
$Obj(\texttt{beautiful\#1}) = 0.25$
This can also be visualized as a triangle as shown in Figure 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.

4.1 Process of Creation

The process of creation of Sentiwordnet is an expansion of the approach used for the three-class

[1]URL: `http://sentiwordnet.isti.cnr.it/search.php?q=beautiful`

sentiment classification from [5, 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 [13]. The 14 terms gave rise to 47 positive and 58 negative synsets. Each synset was represented using the $\texttt{TDS}^{\neg}$ representation from [5, Ch. 1]. This representation vectorized the words in the synset, its Wordnet definition and the sample phrases together with explicit labels for negation.

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 Tr_p^k and Tr_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.

   Each ternary classifier was actually made up of two binary classifiers, *positive* vs. *not positive* and *negative* vs. *not negative*, combined as follows:

	Positive	**Not Positive**
Negative	Objective	Negative
Not Negative	Positive	Objective

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:

   $Pos(s)$ = No of classifiers stating positive / 8
   $Neg(s)$ = No of classifiers stating negative / 8
   $Obj(s)$ = 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:

Part of Speech	Pos	Neg	Obj
Adjectives	0.106	0.151	0.743
Nouns	0.022	0.034	0.944
Verbs	0.026	0.034	0.940
Adverbs	0.235	0.067	0.698
All	0.043	0.054	0.903

5 Sentiment Treebank

The prominent work to perform sentiment analysis using deep learning which will be examined in this section was presented by [11]. It is a system which performs sentiment analysis on the sentence level by parsing the sentence and identifying the sentiment of each node in the parse tree, starting at the leaves. In order to do this, the work also came up with a lexicon called the *Sentiment Treebank*, which is a lexicon consisting of partial parse trees annotated with sentiment.

The Sentiment Treebank is the sentiment lexicon introduced in the work. It is a lexicon that contains partial parse trees, with each parse tree annotated with a sentiment score.

The lexicon was obtained by taking the following steps:

1. The movie review corpus from [8] was obtained. This is a sentence-level movie review dataset obtained from `www.rottentomatoes.com`, consisting of 10,662 sentences.

2. Each of the sentences was parsed using the Stanford Parser. This gave a parse tree for each sentence.

3. The parse trees were split into phrases, i.e. each parse tree was split into its components, each of which was then output as a phrase. This gave rise to 215,154 phrases.

4. Each of these phrases was tagged for sentiment using Amazon's Mechanical Turks interface.

Initially the granularity of the sentiment values was 25, i.e. 25 possible values could be given for the sentiment, but it was observed from the data from the Mechanical Turks experiment that most responses contained any one of only 5 values. These 5 values were then called *'very positive'*, *'positive'*, *'neutral'*, *'negative'* and *'very negative'*.

6 Conclusion

To summarize, this paper covered three sentiment lexicons – SO-CAL, SentiWordnet, and Sentiment Treebank. SO-CAL was created manually and associates an integer between -5 and $+5$ to a word. SentiWordnet was created automatically and associates three values representing positivity, negativity and objectivity of a Wordnet synset. As it was created automatically, it has high coverage. Sentiment Treebank was created using crowdsourcing and associates a label between 'very positive' and 'very negative' to a phrase. As a result, one can see that the landscape of sentiment lexicons is highly varied.

References

[1] Dictionary facts: Oxford english dictionary. Accessed: 06-04-2014.

[2] Jerry Boucher and Charles E. Osgood. The pollyanna hypothesis. *Journal of Verbal Learning & Verbal Behavior*, 8(1):1–8, 1969.

[3] Julian Brooke. A semantic approach to automatic text sentiment analysis. M.A. thesis, Stanford University, 2001.

[4] Kenneth Church. A pendulum swung too far. *Linguistic Issues in Language Technology*, 6(5).

[5] Andrea Esuli. *Automatic Generation of Lexical Resources for Opinion Mining: Models, Algorithms and Applications*. PhD thesis, Universita di Pisa, 2008.

[6] Andrea Esuli and Fabrizio Sebastiani. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC-06)*, pages 417–422, 2006.

[7] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. pages 143–151, 1997.

[8] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*,

pages 115–124. Association for Computational Linguistics, 2005.

[9] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86. Association for Computational Linguistics, 2002.

[10] Roser Sauri. *A Factuality Profiler for Eventualities in Text*. PhD thesis, Brandeis University, 2008.

[11] Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*. Association for Computational Linguistics, 2013.

[12] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307, 2011.

[13] Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346, 2003.