WikiSent : Weakly Supervised Sentiment Analysis Through Extractive Summarization With Wikipedia

Subhabrata Mukherjee, Pushpak Bhattacharyya
Dept. of Computer Science and Engineering, IIT Bombay
{subhabratam, pb}@cse.iitb.ac.in

Abstract: This paper describes a *weakly* supervised system for sentiment analysis in the movie review domain. The objective is to classify a movie review into a polarity class, *positive* or *negative*, based on those sentences bearing opinion on the movie alone, leaving out other irrelevant text. *Wikipedia* incorporates the world knowledge of *movie-specific features* in the system which is used to obtain an *extractive summary* of the review, consisting of the reviewer's opinions about the specific aspects of the movie. This filters out the concepts which are irrelevant or objective with respect to the given movie. The proposed system, *WikiSent*, does not require any labeled data for training. It achieves a better or comparable accuracy to the existing semi-supervised and unsupervised systems in the domain, on the same dataset. We also perform a general movie review *trend analysis* using WikiSent.

Keywords: Sentiment Analysis, Wikipedia, Information Extraction, Weakly Supervised System, Text mining, Summarization, Reviews

1 Introduction

In the movie domain, like many other product domains, there has been a flurry of review sites giving critics view about the performance of the actor, director, story as well as the public acceptance of the movie. This is of importance not only to the people directly related to the movie-making but also to the audience, whose viewing decisions are quite influenced by these reviews.

Sentiment analysis of movie reviews aims to automatically infer the opinion of the movie reviewer and often generates a rating on a pre-defined scale. Automated analysis of movie reviews is quite a challenge in text classification due to the various nuances associated with the critic reviews. The author may talk about a lot of topics which are not directly related to the movie in focus. Tightly intermixed with various objective statements are his subjective opinions about the movie, which are quite difficult to extract. Here, an objective statement is defined as not just a factual statement, but as objective from the point of view of analyzing the opinion about a particular movie.

This work is different from traditional automatic text summarization or abstractive summarization. This is because the objective is not to obtain a shorter text but to retrieve *relevant opinionated text*. This focused extraction requires external world knowledge about the various technical aspects of the movie (like *movie plot, film crew, characters, domain specific features* etc.). Wikipedia feeds the system with this technical knowledge which is used to create an extract of the review. This extract is subsequently classified by a lexicon. Figure 1 shows the system architecture.

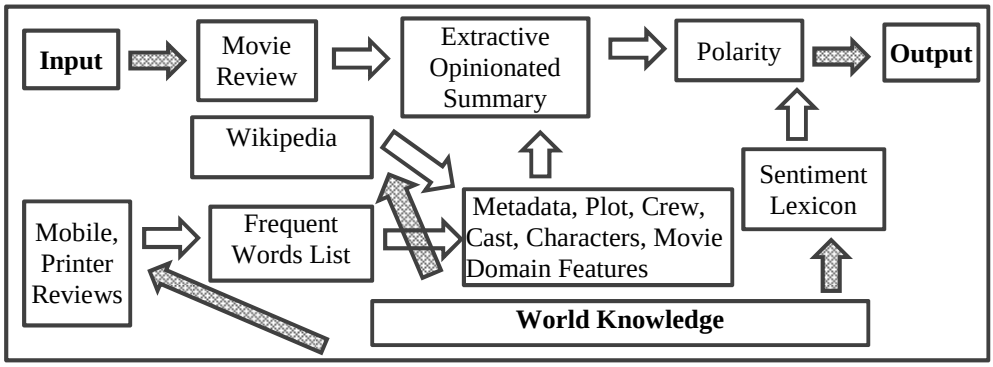

Fig. 1. System Block Diagram

Consider the fragment of a review of the movie L.I.E taken from the IMDB movie review corpus [14] which has been tagged as a negative review:

Example 1. Review of the Movie L.I.E

---

[1]Best remembered for his understated performance as Dr. Hannibal Lecter in Michael Mann's forensics thriller, Manhunter, Scottish character actor Brian Cox brings something special to every movie he works on. [2]Usually playing a bit role in some studio schlock (he dies halfway through The Long Kiss Goodnight), he's only occasionally given something meaty and substantial to do. [3]If you want to see some brilliant acting, check out his work as a dogged police inspector opposite Frances McDormand in Ken Loach's Hidden Agenda.

[4]Cox plays the role of Big John Harrigan in the disturbing new indie flick L.I.E., which Lot 47 picked up at Sundance when other distributors were scared to budge. [5]Big John feels the love that dares not speak its name, but he expresses it through seeking out adolescents and bringing them back to his pad. [6]What bothered some audience members was the presentation of Big John in an oddly empathetic light. [7]He's an even-tempered, funny, robust old man who actually listens to the kids' problems (as opposed to their parents and friends, both caught up in the high-wire act of their own confused lives.). [8]He'll have sex-for-pay with them only after an elaborate courtship, charming them with temptations from the grown-up world"

……

[9]It's typical of unimaginative cinema to wrap things up with a bullet, sparing the writers from actually having to come up with a complex, philosophical note. [10]In this regard, l.i.e. (and countless other indie films) share something in common with blockbuster action films: problems are solved when the obstacle is removed. [11]How often does real life work this way to extend the question : if a movie is striving for realism , do dramatic contrivances destroy the illusion?

The first paragraph of the review talks about the central character Brian Cox's notable performance in some earlier movie. The second paragraph gives a brief description of his character in an empathetic light which comprises of positive opinions about the character. The reviewer opinion about the movie comes only in the last paragraph, where he gives some negative opinions about the movie. The review consists of majority positive words, not all of which are significant to the reviewer opinion, outweighing the negative opinions about the movie. A bag-of-words classifier, thus, would wrongly classify this as positive.

In this work, we give an analysis of the various aspects of a movie review in Section 3. There we highlight the significant and non-significant concepts for sentiment analysis of movie reviews. Section 4 describes how the sectional information in Wikipedia helps in this task. It also gives the automated feature extraction process from Wikipedia. Section 5 gives the algorithm for the extraction of the opinion summary consisting of the *relevant* reviewer statements, which is classified by a sentiment lexicon in Section 6. Section 7 discusses different approaches for parameter learning for the model. The experimental

evaluation is presented in Section 8 on a gold standard dataset of 2000 movie reviews as well as on an unlabeled pool of 27,000 documents to find the trend. Section 9 discusses the results followed by conclusions in Section 10.

The main contribution of this work is to show how Wikipedia info-box sectional information can be used to incorporate *World Knowledge* in a system, to obtain an *extractive opinionated summary* of a movie review. This, in turn, helps in sentiment analysis of the review due to the filtering out of objective concepts from subjective opinions. This work is *mostly* unsupervised, requiring no labeled training data. The weak supervision comes from the usage of resources like WordNet, POS-tagger and Sentiment Lexicons, due to their mode of construction.

2 Related Work

There are 2 prominent paradigms in automatic text summarization [3]: *extractive* and *abstractive* text summarization. While *extractive* text summarization attempts to identify prominent sections of a text by giving more emphasis on the content of the summary, *abstractive* text summarization gives more emphasis on the form so that the sentences are syntactically and semantically coherent. The *topic-driven* summarization paradigm is more common to IR where the summary content is based on the user query about a particular topic. [4] attempts to find the top-ranked significant sentences based on the frequency of the content words present in it. [5] gives importance to the position of a sentence i.e. where the sentence appears in the text and comes up with an optimum position policy and emphasis on the cue words. [6] uses tf-idf to retrieve signature words, NER to retrieve tokens, shallow discourse analysis for cohesion and also uses synonym and morphological variants of lexical terms using WordNet. [7] uses a rich set of features like *Title, Tf & Tf-Idf scores, Position score, Query Signature, IR Signature, Sentence Length, Average Lexical Connectivity, Numerical Data, Proper Name, Pronoun & Adjective, Weekday & Month, Quotation, First Sentence* etc. and uses decision trees to learn the feature weights. There are other works based on HMM [8], RTS [9], lexical chain and cohesion [10].

We use many of the above features for finding the extract of the summary. However, our objective differs in the fact that we intend to derive relevant *subjective sentences* significant for the movie, and not objective sentences. It is also topic-driven, depending on the *movie plot, actors, film crew, fictional characters* etc.

[11] proposes to find subjective sentences using lexical resources where the authors hypothesize that subjective sentences will be more similar to opinion sentences than to factual sentences. As a measure of similarity between two sentences they used different measures including shared words, phrases and the WordNet. [12] focuses on extracting top sentiment keywords which is based on Pointwise Mutual Information (PMI) measure [13].

The pioneering work for subjectivity detection is done in [14], where the authors use min-cut to leverage the *coherency* between the sentences. The fundamental assumption is that local proximity preserves the objectivity or subjectivity relation in the review. But the work is completely supervised requiring two levels of tagging. Firstly, there is tagging at the sentence level to train the classifier about the subjectivity or objectivity of individual sentences. Secondly, there is tagging at the document level to train another classifier to distinguish between positive and negative reviews. Hence, this requires a lot of manual effort. [15] integrates graph-cut with linguistic knowledge in the form of WordNet to exploit similarity in the set of documents to be classified.

Now, if a system possesses world knowledge about the technical aspects of the movie, then it would be easier for it to detect objective or subjective sentences based on the key

concepts or features of a movie. Wikipedia[1] can incorporate this world knowledge in the system. Wikipedia is recently used in a number of works mainly for concept expansion in IR for expanding the query signature [16], [17], [18] as well as for topic driven multi document summarization [19].

There has been a few works in sentiment analysis using Wikipedia [20], [21]. [20] focuses on concept expansion using Wikipedia where they expand the feature vector constructed from a movie review with related concepts from the Wikipedia. This increases accuracy as it helps in unknown concept classification due to expansion, but it *does not* address the concern of separating subjective from objective sentences.

These works do not take advantage of the *Sectional* arrangement of the Wikipedia articles into categories. Each Wikipedia movie article has sections like *Plot, Cast, Production etc.* which can be explicitly used to train a system about the different aspects of a movie. In this work, our objective is to develop a system that requires no labeled data for training and classifies the *opinionated extractive summary* of the movie; where the summary is created based on the extracted information from Wikipedia.

3 Facets of a Movie Review

Movie review analysis is a challenging domain in Sentiment Analysis due to sarcasm, thwarting and requirement of extensive world knowledge. The reviewer opinion about the movie may target the characters in the movie, the plot or his expectations from the crew involved. We categorize the reviewer statements in the following categories:

Table 1. Reviewer Statement Categories

1. General Perception about the Crew	6. Opinion about Movie Characters
2. Objective Facts about the Crew and Movies	7. Characteristics of a Movie or Genre
3. Past Performance of the Crew and Movies	8. Opinion about the Movie and Crew
4. Expectations from the Movie or Crew	9. Unrelated Category
5. Movie Plot	

We define *Crew* in a movie as the people who are involved in making of the movie like the *Producer, Director, Actor, Story-Writer, Cinematographer, Musician* etc. We are mainly interested in extracting opinions from *Category 8* where all the other Categories may lend a supporting role to back his opinions or add noise. We give examples (taken from the movie reviews of the IMDB corpus [2]) for each of the categories in *Table 2*.

Table 2. Reviewer Statement Categories with Examples

1.	General Perception about the Crew	*John Travolta is considered by many to be a has-been, or a one-hit wonder … Leornardo DeCaprio is an awesome actor.*
2.	Objective Facts about the Crew and Movie	*Born into a family of thespians -- parents Roger Winslet and Sally Bridges-Winslet were both stage actors – Kate Winslet came into her talent at an early age.*
3.	Past Performance of the Crew	*The role that transformed Winslet from art house attraction to international star was Rose DeWitt Bukater, the passionate, rosy-cheeked aristocrat in James Cameron's Titanic (1997).*
4.	Expectations from the Movie or Crew	*I cancelled the date with my girlfriend just to watch my favorite star featuring in this movie.*

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

5.	Movie Plot	*L.I.E. stands for Long Island Expressway, which slices through the strip malls and middle-class homes of suburbia. Filmmaker Michael Cuesta uses it as a metaphor of dangerous escape for his 15-year old protagonist, Howie (Paul Franklin Dano).*
6.	Opinion about the Characters in the Movie	*He's an even-tempered, funny, robust old man who actually listens to the kids' problems (as opposed to their parents and friends, both caught up in high-wire act of their confused lives.).*
7.	Characteristics of a Movie or Genre	*Horror movies are supposed to be scary.* *There is an axiom that directors who have a big hit with their debut have a big bomb with their second film.*
8.	Opinion about the Movie and Crew	*While the movie is brutal, the violence is neither very graphic nor gratuitous. It may scare the little ones, but not any teen-ager. Besides the awesome direction, the ageless glamor and fabulous acting of Leonardo DeCaprio and Kate Winslet made the movie titanic a timeless hit.*
9.	Unrelated Category	*So my grandson gives me passes to this new picture One Night at McCool's because the free screening is the same night as that horrible show with those poor prisoners trapped on the island who eat the bugs. "Go," he says, "it's just like Rush-o-Man."*

It is evident from the examples above why movie domain text is difficult to analyze. Consider the Example from *Category 5*, which talks about the movie plot. The keyword dangerous, there, makes the segment negative. But it expresses only a concept about the movie and not the reviewer opinion. Similarly, *Category 6 Example* talks about a character in the movie which expresses a positive opinion but unrelated w.r.t the opinion analysis of the review. *Category 7 Example* has the keywords *movie* and *audience* directly related to the movie domain. Thus it is more probable that they are expressing some direct opinion about a certain aspect of the movie. Similarly, the name of the *actors* in *Category 8 Example 2* makes it relevant, as they reflect opinions about the persons involved in the making of the movie. Hence, it is important to extract only those concepts which are significant from the point of view of opinion analysis of the movie and filter out the non-significant portion. A unigram based bag-of-words model would capture a lot of noise, if it considers all categories to be equally relevant.

4 Wikipedia Information Extraction for Movie Review Analysis

Wikipedia is the largest English knowledge repository consisting of more than 20 million articles collaboratively written by volunteers all around the world. It is open-access and regularly updated by about 100,000 active contributors daily. Each Wikipedia page is an article on some known concept or topic. Each article belongs to one of the many defined categories or subcategories. For Example, Category Film has 31 sub-categories like *Film making, Works about Films, Film Culture* etc. Furthermore, each article has a number of sections which can be edited separately. Any Wikipedia article on films may consist of sections like *Plot, Cast, Production, Marketing, Release* etc. which are common among most of the articles of that category. We utilize this feature to extract movie specific information from the Wikipedia.

A Wikipedia movie article consists of a small paragraph, in the beginning, giving information about the movie story, crew and achievements in short. We call this section the *Metadata* about the movie. There is a table on the extreme right of the article which provides information about the name of the *Producer, Director, Actors, Cinematographer*

etc. We call this section the *Crew* Information. There is a section on the movie plot which summarizes the story of the movie and talks about its fictional aspects. We call this section the *Plot* of the movie. There is another section which gives information about the actors in the movie, the roles they perform and the characters they enact. We call this section the *Character* of the movie. We use all the above information extracted from the Wikipedia article about the particular movie to incorporate World Knowledge into the system.

We used the IMDB movie corpus [2] consisting of 2,000 tagged positive and negative movie reviews, each class consisting of 1,000 reviews. The tagged information is used only for evaluation. Furthermore, there is a collection of 27,000 raw html documents taken from the IMDB review site, from which the authors extracted and tagged the above 2000 documents, used for finding the general trend in the movie domain.

4.1 Wikipedia Article Retrieval

The 2,000 processed review documents had their titles removed. Thus the corresponding reviews had to be retrieved from the unprocessed html documents and their titles extracted. The title of the movie review was used to construct a *http get* request to retrieve the corresponding html page of the movie directly from Wikipedia. In case of multiple articles in different domains with same name, the *film* tag was used to retrieve the desired article. For multiple movies with the same name, the year (in which the movie was released) information available with the title was used to extract the correct Wikipedia article. Thus the Wikipedia article retrieval was in the order *film name* → *film tag* → *film year*.

4.2 Crew Information

All the crew information were extracted from the table in the right hand side of the Wiki article, bearing the name of all the persons involved in the making of the movie like the *director, producer, cinematographer, story-writer etc.*, and added to the **Crew** list.

The first line in the Wiki article that contains the phrase *Directed by* or *Author* and is a part of any table (detected by the html tags */td, /tr, /th, /table etc.*) is taken as the start of the Crew info-section. The phrases *Release Date* or *Language* or the html tags */table* and */tbody*, that signify the end of the Crew table, is taken as the end of the info-section.

4.3 Metadata Extraction

The metadata was extracted from the html page just below the title of the article. The text was POS-tagged using a part of speech tagger[2] and all the *Nouns* were extracted. The Nouns were further stemmed[3] and added to the **Metadata** list. The words were stemmed so that *acting* and *action* have the same entry corresponding to *act*. Some movie articles in Wikipedia had the Plot section missing. The metadata was used in those cases to replace the Plot. In other cases, the *Metadata* information was simply appended to the *Plot* information.

According to the structure of the Wikipedia html page on movie articles, the meta-data information on the movie appears just after the Crew table in the html page. This section spans the page from the end of the Crew info-section till the start of the next info-box, which is indicated by the Wiki tag *editsection*. The Wiki tag *editsection* allows users to edit an info-box. All the Wikipedia info-boxes are editable.

4.4 Plot Extraction

The movie plot was extracted from the *Plot Section* of the Wikipedia. The words were extracted similarly as in Metadata extraction. In both the Plot and Metadata, the concepts

[2] http://nlp.stanford.edu/software/tagger.shtml

[3]http://sourceforge.net/projects/stemmers/files/Lovins-stemmer-Java/

were restricted to be *Nouns*. For example, in the Harry Potter movie the nouns *wizard, witch, magic, wand, death-eater, power* etc. depict concepts in the movie.
The Wikipedia html id-attribute *id="Plot"* is taken as the beginning of the Plot info-box which spans the text till the next info-box, indicated by the Wiki tag *editsection*.

If we consider all the Nouns, a lot of noise will be incorporated into the **Plot** list. This is because the commonly used Nouns like *vision, flight, inform, way* etc. are added as well. To prevent this, both in the *Metadata* and the *Plot*, a separate list was created comprising of the frequently found terms (it will be shortly discussed how this list was compiled) in a corpus. Subsequently, the frequently occurring words were filtered out, leaving only *movie* and *genre-specific* concepts in the Metadata and Plot list.

4.5 Character Extraction

The **Cast** section in the Wiki article has the name of the actors and the name of the characters they enact. These character names were extracted and added to the **Character** list. These depict the fictional roles played by the actors in the movie.

The Wiki html id-attribute *id="Cast"* is taken as the beginning of the Cast info-box which spans the text till the next info-box, indicated by the Wiki tag *editsection*.

4.6 Frequent Word List Construction

The underlying hypothesis for this list creation is that movie reviews will have certain *concepts and terms* those are exclusive to this domain and will less frequently occur in other domains. Review data from the *Printer* and *Mobile Phone* domains[4] were used to create a list of frequently occurring terms in those domains. Since those domains are completely disjoint from the movie review domain, words which *frequently* occur in all of these domains must be commonly occurring words. Thus the commonly used words list consists of the frequently occurring terms in all these domains. The *tf-idf* measure was used and all those words above a threshold were added to the **FreqWords** list. For example, the word *person* (which is a Noun) occurred in all the domains with a very high frequency and thus added to the **FreqWords** list.

4.7 Domain Specific Feature List Construction

Wikipedia articles on films and aspects of films[5] were extracted. The sentences in those documents were POS-tagged. The Nouns were retrieved and frequently occurring words were removed. The remaining words were stemmed and added to the **MovieFeature** list. *Table 3* shows a snapshot of the genre specific terms extracted from Wiki movie articles.

Table 3. Extracted Movie Domain Specific Terms

Movie, Staffing, Casting, Writing, Theory, Rewriting, Screenplay, Format, Treatments, Scriptments, Synopsis, Logline, Pitching, Certification, Scripts, Budget, Ideas, Funding, Plans, Grants, Pitching, Tax, Contracts, Law, Copyright, Pre-production, Budgeting, Scheduling, Pre-production, Film, Stock, Story, Boarding, Plot, Directors, Location, Scouting, …..

5 Algorithm to Extract Opinion Summary

Section 4 describes the creation of the feature lists *Metadata*, *Plot*, *Crew*, *Character*, *FreqWords* and *MovieFeature*. Now, given a movie review the objective is to extract all

[4]http://mllab.csa.iisc.ernet.in/downloads/reviewmining/fulldata.tar.gz
[5] http://en.wikibooks.org/wiki/Movie_making_manual

the sentences that reflect the true opinion of the reviewer about the movie. This forms the *OpinionSummary* of the movie. A sentence-by-sentence analysis of a review is performed.

Any sentence not involving any word from any of the above lists is not considered relevant at all, thus pertaining to the *Unrelated Category 9*. Sentences involving concepts from the *Plot, Metadata* and *Character Lists* are considered least significant, as they talk about the movie story and not about the reviewer opinion. But they are not considered completely irrelevant as they may contain sentences that back the reviewer's opinion about the movie. This covers *Category 5 and 6*. Sentences containing terms from the *MovieFeature* list are likely to comment on some specific aspect of the movie and are thus considered more relevant (*Category 7*). Finally, any sentence containing the movie *Title*, or *Crew* information is considered most relevant, as the reviewer is likely to express his opinion about the movie. This covers *Category 1-4. The final opinion of the reviewer (Category 8) is actually a weighted function of all the other Categories.*

The *Metadata, Plot and Character* lists are combined into a single list called the *Plot*. We now have 3 main categories of features corresponding to the *Plot*, *Crew* and *MovieFeature* lists with an auxiliary *FreqWords* list.

Given a movie review R with n sentences S_i, our objective is to determine whether each sentence S_i is to be *accepted* or *rejected* based on its *relevance* to the reviewer opinion. Let each sentence S_i consist of n_i words $w_{ij}, j \in 1 \dots n_i$. The *Plot* list does not contain any word from the *FreqWords* list or the *MovieFeature* list. Similarly, the *MovieFeature* list also does not contain any word from the *FreqWords* list. The *relevance factor* of the sentence S_i is given by,

$$Rel_{factor_i} = \alpha \sum_j 1_{w_{ij} \in Crew \text{ or } w_{ij} \in MovieTitle} + \beta \sum_j 1_{w_{ij} \in MovieFeature} - \gamma \sum_j 1_{w_{ij} \in Plot, w_{ij} \notin Crew, w_{ij} \notin MovieTitle}$$
$$where \ \alpha, \beta, \gamma > 0, \qquad \alpha > \beta \tag{1}$$

The relevance factor is actually a weighted combination of the various *features* in the *lists*. It *counts* the words appearing from different lists in a sentence and *weighs* them separately. The concepts belonging to the *Plot* are not so much relevant in judging the reviewer's opinion about the movie and may add noise. They play a dampening role, which is captured in the '-' sign before γ. More weight is given to any word referring to the *Crew* or *Movie Title* than any word simply referring to a *movie domain feature*, which is captured in $\alpha > \beta$. Any sentence S_i is accepted if Acc_{factor_i} corresponding to S_i is 1.

$$Acc_{factor_i} = 1 \ \boldsymbol{if} \ Rel_{factor_i} \geq \theta \ and \ \exists w_{ij} \in S_i \ \boldsymbol{s.t.} \ w_{ij} \in Crew \ or \ MovieFeature \ or \ Movie \ Title$$
$$= 0 \ otherwise \tag{2}$$

Here θ is a threshold parameter. Thus any sentence is accepted as being relevant, if its score is greater than some threshold value and there is *atleast one* word in the sentence that belongs to the *Crew*, *MovieFeature* or the *MovieTitle* lists.

Considering the Review *Example 1*, the algorithm works as follows: Let us consider α, β, γ to assume integer values. Let $\alpha = 2, \beta = 1, \gamma = 1$. The variables assume the first integer values satisfying all the conditions in *Equation 1*. *Let* $\theta = 0$.

In Sentence [1], *Brian Cox* is the only keyword present and it belongs to the *Cast* list (the other keywords are not present in the Wiki article for the film L.I.E.).
Rel_{factor_1}=2*1+1*0-1*0=2 ≥ 0, $Acc_{factor_1} = 1$ and the sentence is accepted. In [2], there is no keyword from the lists and it is rejected straightaway. [3] has the keyword *acting*

from MovieFeature and is accepted where, Rel_{factor_3}=1, $Acc_{factor_1} = 1$. [4] has the keywords *Cox, L.I.E* from *Cast and MovieTitle*, *John Harrigan* from *Character* list and *distributor* from the *MovieFeature* list. Rel_{factor_4}=2*2+1-1=4 $\geq$ 0 and the sentence is accepted. [5] has only the keyword *Big John* from *Character*. Its Rel_{factor_5}=0+0-1=-1$\not\geq$ 0 and the sentence is rejected. [6] has the keyword *audience* from *MovieFeature* and *Big John* from *Character*. Its Rel_{factor_6}=0+1-1=0 $\geq$ 0 and it is accepted. [7] has the keywords *temper, friend* from *Plot* and Rel_{factor_7}=0+0-2=-2 $\not\geq$ 0 and is rejected. [8] has the keywords *sex, charm* from *Plot* and Rel_{factor_8}=0+0-2=-2 $\not\geq$ 0 and is rejected. [9] has the keywords *cinema, writers* from *MovieFeature* and *bullet* from *Plot*. Thus Rel_{factor_9}=0+1*2-1=1 $\geq$ 0 and is accepted as being relevant. [10] has the keywords *l.i.e* from *MovieTitle, films(2), action* from *MovieFeature*. Thus $Rel_{factor_{10}}$=2*1+1*3-0=5 $\geq$ 0 and is accepted as being relevant. [11] has the keyword *movie* from *MovieFeature*. $Rel_{factor_{11}}$=0+1*1-0=1 $\geq$ 0 and it is accepted.

Algorithm 1. Extractive Opinion Summary from Review

Input : Review R
Output: OpinionSummary
Step 1: Extract the Crew list from Wikipedia
Step 2: Extract the Plot list from Wikipedia
Step 3: Extract the MovieFeature list from Wikipedia
Step 4: Extract the FreqWords list as the common frequently occurring concepts in Mobile Phone, Printer and Movie domains.
Let $OpinionSummary = \emptyset$
for i=1..n
  if $Acc_{factor_i} == 1$
    add S_i *to* $OpinionSummary$

---

6 Classification of the Opinion Summary

The words in the extracted opinion summary can be directly used as features in a supervised classification system. But since we do not use any labeled data for training, a sentiment lexicon is used in the final phase to classify the opinion summary. A sentiment lexicon contains an opinion word along with its polarity. SentiWordNet [22], Inquirer [23] and the Bing Liu [24] sentiment lexicons are used to find the polarity of a word. SentiWordnet is tagged at the synset level (*word sense and polarity*) whereas the Inquirer and Bing Liu sentiment lexicons contain the words and their most commonly considered polarity. While using the SentiWordNet, we use the first sense of a word as we do not perform word sense disambiguation.

Let $pol(w_{ij})$ be the polarity of a word w_{ij}, where i indexes a particular sentence and j indexes a particular word in the sentence. Let $flip_{ij}$ be a variable which indicates whether the polarity of w_{ij} should be flipped or not. Negation handling is being done in the lexical classification, in which the polarity of all the words in a window of 5, from the occurrence of any of the negation operators *not, neither, nor and no*, are flipped.
The final polarity of the review (*pos* or *neg*) is given by,

$$sign\left(\sum_{i=1}^{m}\sum_{j=1}^{n_i} flip_{ij} \times pol(w_{ij}) \times g(sign\left(flip_{ij} \times pol(w_{ij})\right)\right)$$
$$where\ g(x) = negation\_bias\ if\ x < 0$$
$$= 1 \qquad if\ x \geq 0 \qquad (3)$$

The polarity is given by the signed sum of the polarity bearing opinion words in the sentence weighted by the *negation_bias*. The range of the polarity function is $[-m\ \ m]$, where m is the number of polarity-bearing words in the sentence.

Any review has more explicit positive expressions of opinion than negative ones [1],[25],[26],[27]. This is because negative sentiment is often implicit as in sarcasm and thwarting. Likewise sentiment lexicons have a high bias towards positive and objective words [1]. A negation bias is added, so that the occurrence of any negative word in the review is weighed more than a positive word. The SentiWordNet has a high coverage as well as a high bias towards positive and objective terms since it uses a semi-supervised learning method. Inquirer, being manually hand-tagged, has a low coverage but high accuracy similar to the Bing Liu sentiment lexicon, although the latter has a higher coverage than the Inquirer. We experimented with all the *three* lexicons.

7 Parameter Setting

A simple but effective strategy used in information retrieval and automatic text summarization for feature weighting is to use weights that are simple integral multiples, preferably prime, to reduce the possibility of ties [35]. There are 5 parameters for the model we used: $\alpha, \beta, \gamma, \theta\ and\ negation\_bias$. The first 3 parameters can be best trained if sentence level label (whether each sentence is relevant or not) information is available. However, in the absence of any label information, we adopt a simpler approach as mentioned above. We took the first set of integer values, satisfying all the constraints in *Equation 1*, and assigned them to the first 3 parameters : $\boldsymbol{\alpha = 2, \beta = 1, \gamma = 1}$.

The value of θ should be set such that the number of significant keywords, from *Crew, MovieFeature and MovieTitle* lists, should be more than the number of keywords from less significant concepts like *Plot* and *Character* lists. This means Rel_{factor_i} should be greater than or equal to zero which, implies $\theta = 0$.

The authors in [1] weighted up the negative expressions by a fixed amount (50%) over positive expressions. In our experiment, value of the *negation bias* is determined as:

$$negation\_bias = \frac{positive\ opinion\ words\ in\ the\ corpus}{negative\ opinion\ words\ in\ the\ corpus}$$

This is done to give any positive or negative opinion word equal importance in the review. In the ideal case, since the corpus is balanced having equal number of positive and negative reviews, the ratio should have been close to 1, in absence of any negation bias. However, due to the bias problem explained before, the *negation_bias* comes out to be 1.4.

Semi-Supervised Learning of Parameters

This work *does not* evaluate this angle for parameter learning, since our objective has been to develop a system that requires no labeling information at the sentence level or the document level. However, if some sentence level information is available, a robust learning of parameters is possible.

Equation 1 and 2 can be re-written as:

$$Rel_{factor_i} = \alpha \times X_{i,1} + \beta \times X_{i,2} - \gamma \times X_{i,3}$$

$$Acc_{factor_i} = Rel_{factor_i} - \theta$$

$$= \alpha \times X_{i,1} + \beta \times X_{i,2} - \gamma \times X_{i,3} - \theta$$

$$= \alpha \times X_{i,1} + \beta \times X_{i,2} - \gamma \times X_{i,3} - \theta \times X_{i,4} \text{ (where } X_{i,4} = 1)$$

Let Y_i be the binary label information corresponding to each sentence in the development set, where Y_i=*1* if $Acc_{factor_i} \geq 0$ and -1 otherwise.

$$Y_i = \boldsymbol{W^T}.X_i^T \quad \text{where, } \boldsymbol{W} = [\alpha\ \beta\ -\gamma\ -\theta]^T\ and\ \ X_i = [X_{i,1}\ X_{i,2}\ X_{i,3}\ X_{i,4}]$$

$$or, \boldsymbol{Y} = \boldsymbol{W^T}.\boldsymbol{X}\ where, \boldsymbol{X} = [X_1^T\ X_2^T\ \dots\ X_n^T]\ and\ \boldsymbol{Y} = [Y_1\ Y_2\ \dots\ Y_n]$$

This is a linear regression problem which can be solved by the ordinary least squares method by minimizing the sum of the squared residuals i.e. the sum of the squares of the difference between the observed and the predicted values [37]. The solution for $\boldsymbol{W}$ is given by $W = (\boldsymbol{X^T X})^{-1}\boldsymbol{X^T Y}$.
A regularizer can be added to protect against over-fitting and the solution can be modified as: $\boldsymbol{W} = (\boldsymbol{X^T X} + \delta \boldsymbol{I})^{-1}\boldsymbol{X^T Y}$ $\quad where\ \delta\ is\ a\ parameter\ and\ \boldsymbol{I}\ is\ the\ identity\ matrix.$

8 Evaluation

The experimental evaluation is performed on the IMBD movie review corpus [2]. It consisted of 2000 reviews collected from the IMDB movie review site and polarity labeled at the document level, 1000 from each of the two classes. This forms our gold standard data. Apart from this, there is an unlabeled corpus of 27,886 unprocessed html files from which the above 2000 reviews had been extracted and labeled by the annotators.
The parameters are set as: $\boldsymbol{\alpha = 2, \beta = 1, \gamma = 1, \theta = 0, negation\_bias = 1.4}$.

8.1 Movie Review Analysis using WikiSent

This analysis is performed on the unprocessed pool of 27,886 html documents. The movie reviews belong to 20 different genres. *Figure 2* shows the number of movies belonging to each genre in the dataset as well as all the genre names.

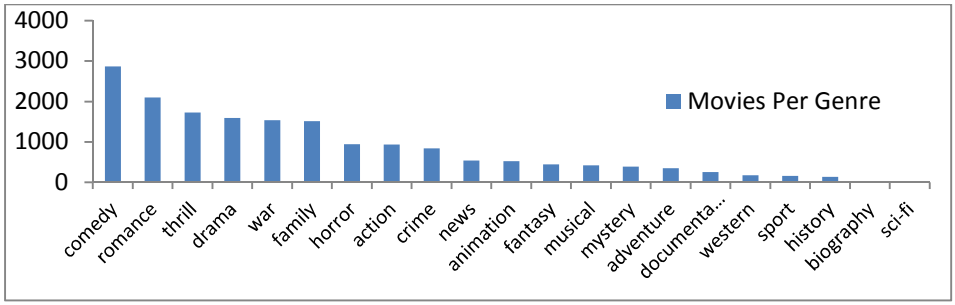

Fig. 2. Movies per Genre in the Dataset

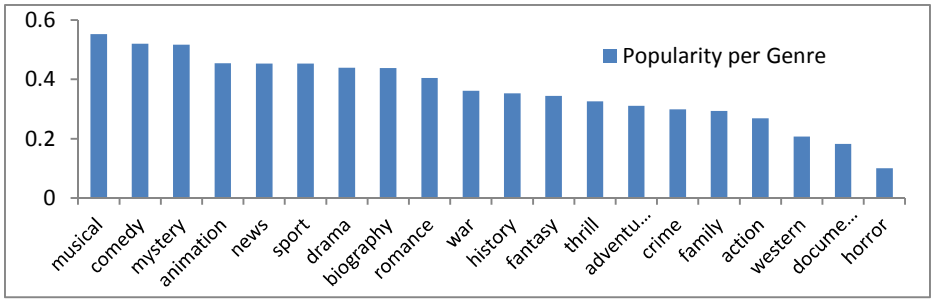

Fig. 3. Genre Popularity in the Dataset

The genre popularity (refer to *Figure 3*) is given by:

$$Genre\ Popularity = \frac{Positive\ Movie\ Reviews\ per\ Genre}{Total\ Movie\ Reviews\ per\ Genre}$$

Table 4 gives the fraction of the movie reviews that are predicted to be positive and negative by WikiSent and the baseline bag-of-words model (expressed in *percentage*). *Figure 4* shows the total number of movies released and the total number of movies predicted to be positive *per year* (as is present in the dataset).

Table 4. Movie Review Polarity Comparison of WikiSent vs. Baseline System

	WikiSent	Bag-of-Words Baseline
Positive Reviews (%)	48.95	81.2
Negative Reviews (%)	51.05	18.79

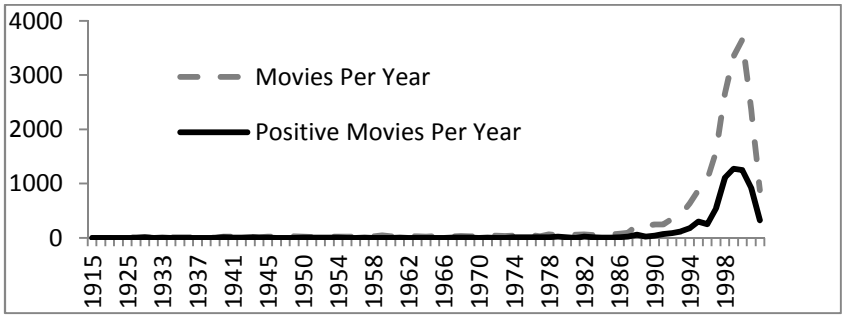

Fig. 4. Movie Popularity per Year in the Dataset

8.2 WikiSent Evaluation on the Gold Standard Data

The baseline for WikiSent has been taken as the bag-of-words model, in which all the terms in a review are considered relevant and classified with the help of a lexicon. The baseline accuracy *Table 5* is adapted from [1], in which the evaluation is done on the same dataset as ours, but with different sentiment lexicons. It also shows the performance of *WikiSent* using different sentiment lexicons. *Table 6* shows the accuracy variation with $\boldsymbol{\theta}$ and *Table 7* shows the accuracy variation with *negation_bias*. *Table 8* shows that accuracy comparison of WikiSent with the *best performing unsupervised and semi-supervised systems* in the domain. The compared systems have been evaluated on the *same* corpus [2]. We directly incorporated the accuracies from the respective papers for comparison.

Table 5. Accuracy using Different Lexicons Without and With WikiSent

Only Google-Full	66.31	*Only* SentiWordNet-Basic	62.89
Only Google-Basic	67.42	*Only* Subjectivity-Full	65.42
Only Maryland-Full-Now	67.42	*Only* Subjectivity-Basic	68.63
Only Maryland-Basic	62.26	*WikiSent* + SentiWordNet	73.30
Only GI-Full	64.21	**WikiSent + Inquirer (GI)**	**76.85**
Only GI-Basic	65.68	*WikiSent* + Bing Liu	69.80
Only SentiWordNet-Full	61.89	*WikiSent* + Above 3 Lexicons	74.56

Table 6. Accuracy Variation with $\boldsymbol{\theta}$

$\boldsymbol{\theta}$	Accuracy
3	63.63
2	66.74
1	69.31
0	**76.85**
-1	71.43

Table 7. Accuracy Variation with *negation_bias*

Negation_Bias	Accuracy
1	70.89
1.1	73.90
1.3	74.74
1.4	**76.85**
1.5	75.53
1.6	73.59

Table 8. Accuracy Comparison with Different Systems

System	Classification Method	Accuracy
Li [33]	Semi Supervised with 10% doc. label	60.00
Li [33]	Semi Supervised with 40% doc. label	60.00
Lin [32] LSM	Unsupervised without prior info	61.70
Taboada SO-CAL Basic [1]	Lexicon Generation	68.05
Shi [28], Dasgupta [29]	Eigen Vector Clustering	70.90
Lin [32] LSM	Unsupervised with prior info	74.10
Taboada SO-CAL Full [1]	Lexicon Generation	76.37
Socher [30] RAE	Semi Supervised Recursive Auto Encoders with random word initialization	76.80
WikiSent	**Wikipedia+Lexicon**	**76.85**
Nakagawa [31]	**Supervised** Tree-CRF	77.30
Socher [30] RAE	Semi Supervised Recursive Auto Encoders with 10% cross-validation	77.70

9 Discussions

9.1 Movie Trend Analysis

The movie review corpus contains most movies from the genres *comedy, romance, thrill, drama, war, horror, action, crime* and least number of movies from the genres *west, sport, history, biography, sci-fi* (in the descending order). This depicts a general trend in the movie-making of sticking to the most popular genres.

It is observed that movies belonging to the categories *musical, comedy, mystery, animation* and *news* received the most number of positive reviews whereas movies belonging to the genres *family, action, western, documentary* and *horror* received the least number of positive reviews. This shows that there are a large number of movies from the *comedy* genre and in general these movies tend to do well; whereas the movies from the *action and horror* genres, despite having a large number of releases, do not fare very well. The movies from the genres *musical and animation*, generally have a good acceptance despite less number of releases.

The number of movies per year has grown exponentially with time as is observed from *Figure 4*. This also highlights the importance of movie review analysis in the socio-economic aspect. It is seen that the number of movies as well as good movies have increased with time. The dip after the year 2000 may be attributed to the fact that the data collection process was only till 2002, so the reviews crawled after 2000 were less.

The number of negative reviews in the movie domain actually outweighs the number of positive reviews, to some extent (refer to *Table 4*). This shows that people are a bit skeptical in calling a movie *good*, despite the large number of movies that are being released. It is also seen that the baseline bag-of-words model, which tags a *huge* number of movie reviews as positive, is unable to catch this trend. This also shows the limitation of the baseline model which considers all words to be relevant, in analyzing movie reviews.

9.2 WikiSent Performance Analysis

It is observed that *WikiSent* performs the best with *Inquirer* (GI) among all the other lexicons used with it. It is interesting to find the huge accuracy leap from the baseline accuracy using *Only SentiWordNet* (**61.89 and 62.89**) and *SentiWordNet + WikiSent*

(**73.3**) in *Table 5*. This accuracy improvement is achieved through the deployment of extractive summarization using Wikipedia, by which the objective sentences are eliminated from the review before classification. However, using all the resources (3) together does not give the maximum accuracy.

As the value of $\boldsymbol{\theta}$ increases, fewer sentences are considered relevant due to which many informative sentences are left out. Higher value of $\boldsymbol{\theta}$ means a single sentence should have a large number of representatives from the *Crew*, *MovieFeature* lists, which is rare. Again, as $\boldsymbol{\theta}$ decreases more number of sentences are considered relevant which captures noise due to the inclusion of many insignificant sentences. Low value of $\boldsymbol{\theta}$ means the number of insignificant features from the *Character, Plot* lists outnumber those from the *Crew*, *MovieFeature* lists.

As *negation_bias* increases, negative expressions outweigh positive expressions and accuracy decreases. A low value of the *negation_bias* is unable to offset the inherent corpus bias of the positive expressions and accuracy falters.

WikiSent achieves a better accuracy than most of the existing unsupervised and semi-supervised systems, as is evident from *Table 8*. Its performance is comparable to *SO-Cal Full* [1], *Recursive Auto Encoders* (RAE) [30] and *Tree-CRF* [31]. The accuracy difference of WikiSent with these systems is *not* statistically signficant. The SO-Calculator does not use any document label like WikiSent, whereas the Tree-CRF is supervised and RAE [30] reports a 10-fold cross-validation. It is notable that WikiSent is able to perform better or at par with the semi-supervised systems, which use partial document labels, without using any labeled training data.

9.3 WikiSent Drawbacks

One of the biggest drawbacks of the system is that we do not perform co-reference resolution due to which valuable information is lost. Thus any sentence having a feature anaphorically referring to a relevant feature in the previous sentence will be ignored, due to which significant sentences may be rejected. We do not perform word-sense disambiguation[6], in this work. Since we consider only the first sense of the word (which is not always the best sense according to the context) we miss out on the actual sense of a word and its proper polarity in many cases [36]. For example, we use a simple lexicon which does not distinguish between the various meanings of the same word, like '*bank*' in the sense of '*relying'* which is a *positive term* and '*bank*' in the sense of a '*river bank*' which is *objective*. Furthermore the lexicon, that we use, has a low coverage. If a more specialized lexicon had been used, like SO-CAL [1], more accuracy improvement would have been possible. *Inquirer* suffers from a low coverage since it is manually hand-tagged. Thus many of the polarity-bearing words are absent in it. Though SentiWordNet has a large coverage, it is biased towards positive and objective terms and classifies less number of words as negative.

10 Conclusions and Future Work

In this work, we proposed a weakly supervised approach to sentiment classification of movie reviews. The polarity of the review was determined by filtering out irrelevant objective text from relevant subjective opinions *about the movie in focus*. The relevant opinionated text extraction was done using Wikipedia.

We showed that the incorporation of world knowledge through Wikipedia, to filter out irrelevant objective text, can significantly improve accuracy in sentiment classification.

[6] http://en.wikipedia.org/wiki/Word-sense_disambiguation

Our approach differs from other existing approaches to sentiment classification using Wikipedia in the fact that *WikiSent* does not require any labeled data for training. Weak supervision comes from the usage of resources like WordNet, POS-Tagger and Sentiment Lexicons. This work is different in the way it creates an *extractive opinionated summary* of the movie using the *sectional* information from Wikipedia and then uses a lexicon to find its polarity. The work extensively analyzes the significance of the various aspect specific features in movie reviews that are relevant to sentiment analysis and harnesses this information using Wikipedia. We define an *acceptance factor* for each sentence in the review based on which it should be included in the *extract* or not. In the final stage, we use a simple sentiment lexicon to classify the words in the extract to find the final polarity (*positive or negative*) of the review.

WikiSent has a number of parameters which have been simplistically set in the absence of any label information. In case the polarity information is used, the parameters can be set robustly (the semi-supervised learning method describes this aspect) which may further increase accuracy. The system suffers from the lack of handling *anaphora resolution* and *word sense disambiguation*. Usage of a simple lexicon at the final stage for polarity calculation also mars its accuracy. Addressing these concerns, significant performance improvement may be possible.

Nevertheless, we showed that *WikiSent* attains a *better* or *comparable accuracy* to all the existing unsupervised and semi-supervised systems in the domain on the same dataset, without using any labeled data for training. Furthermore, we also do a general analysis of the movie domain using WikiSent (based on the *genre*, *year of release* and *movie review polarity*) to show the general trends persisting in movie-making as well as public acceptance of the movie.

References

1. Taboada, Maite and Brooke, Julian and Tofiloski, Milan and Voll, Kimberly and Stede, Manfred, Lexicon-based methods for sentiment analysis, Computational Linguistics 2011
2. Pang, Bo and Lee, Lillian and Vaithyanathan, Shivakumar, Thumbs up? Sentiment Classification using Machine Learning Techniques, Proceedings of EMNLP 2002.
3. Das, D. and Martins, A. F. T. A Survey on Automatic Text Summarization, Literature Survey for the Language and Statistics II course at CMU, Pittsburg 2007
4. Luhn, H. P., The automatic creation of literature abstracts, IBM Journal of Research Development. 2(2):159-165, 1958
5. Edmundson, H. P., New methods in automatic extracting. Journal of the ACM, 16(2): 264-285, 1969
6. Aone, C., Okurowski, M. E., Gorlinsky, J., and Larsen, B., A trainable summarizer with knowledge acquired from robust nlp techniques, In Mani, I. and Maybury, M. T., editors, Advances in Automatic Text Summarization, pages 71-80, 1999
7. Lin, C.-Y., Training a selection function for extraction, In Proceedings of CIKM '99, pages 55-62, 1999
8. Conroy, J. M. and O'leary, D. P., Text summarization via hidden markov models, In Proceedings of SIGIR '01, pages 406-407, 2001
9. Marcu, D., Improving summarization through rhetorical parsing tuning, In Proceedings of The Sixth Workshop on Very Large Corpora, pages 206-215, pages 206{215, Montreal, Canada, 1998
10. Barzilay, R. and Elhadad, M. (1997), Using lexical chains for text summarization, In Proceedings ISTS'97, 1997
11. Yu, Hong and Vasileios, Hatzivassiloglou, Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences, In EMNLP, 2003
12. M. Potthast and S. Becker, Opinion Summarization of Web Comments, Proceedings of the 32nd European Conference on Information Retrieval, ECIR 2010,

13. P. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews, In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), 2002.
14. Pang, Bo and Lee, Lillian, A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, Proceedings of the ACL, 2004
15. Alekh Agarwal and Pushpak Bhattacharyya, Sentiment Analysis: A New Approach for Effective Use of Linguistic Knowledge and Exploiting Similarities in a Set of Documents to be Classified, International Conference on Natural Language Processing (ICON 05), IIT Kanpur, India, December, 2005.
16. Christof Müller and Iryna Gurevych, Using Wikipedia and Wiktionary in domain-specific information retrieval, Proceeding CLEF'08 ,Springer-Verlag Berlin, Heidelberg, 2009
17. Fei Wu, Daniel S. Weld, Automatically Refining the Wikipedia Infobox Ontology, WWW 2008
18. David N. Milne and Ian H. Witten and David M. Nichols, A knowledge-based search engine powered by wikipedia, Proceedings of the Sixteenth ACM conference on Conference on information and knowledge management, ACM New York, NY, USA 2007
19. Wang H., Zhou G., Topic-driven Multi-Document Summarization, In Proceedings International Conference on Asian Language Processing, 2010
20. Gabrilovich, Evgeniy and Markovitch, Shaul, Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge,Proceedings of the 21st national conference on Artificial intelligence - Volume 2, AAAI 2006
21. Wang, Pu and Domeniconi, Carlotta, Building semantic kernels for text classification using Wikipedia, KDD 2008.
22. Stefano Baccianella and Andrea Esuli and Fabrizio Sebastiani, SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining, LREC 2010
23. Stone, P.J., Dunphy, D.C., Smith, M.S., Ogilvie, D.M. and associates, The General Inquirer: A Computer Approach to Content Analysis. The MIT Press, 1966
24. Hu, Minqing and Liu, Bing, Mining and summarizing customer reviews, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Seattle, Washington, USA, Aug 22-25, 2004.
25. Alistair Kennedy and Diana Inkpen, Sentiment classification of movie and product reviews using contextual valence shifters, Computational Intelligence, 22(2):110–125, 2006
26. Kimberly Voll and Maite Taboada, Not all words are created equal: Extracting semantic orientation as a function of adjective relevance, In Proceedings of the 20th Australian Joint Conference on Artificial Intelligence, pages 337–346, Gold Coast.
27. Boucher, Jerry D. and Charles E. Osgood, The Pollyanna hypothesis, Journal of Verbal Learning and Verbal Behaviour, 8:1–8, 1969.
28. Jianbo Shi and Jitendra Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000
29. Sajib Dasgupta and Vincent Ng, Topic-wise, Sentiment-wise, or Otherwise? Identifying the Hidden Dimension for Unsupervised Text Classification, EMNLP 2009
30. Socher, Richard and Pennington, Jeffrey and Huang, Eric H. and Ng, Andrew Y. and Manning, Christopher D., Semi-supervised recursive autoencoders for predicting sentiment distribution, EMNLP 2011
31. Tetsuji Nakagawa and Kentaro Inui and Sadao Kurohashi, Dependency tree-based sentiment classification using CRFs with hidden variables, NAACL 2010
32. Lin, Chenghua and He, Yulan and Everson, Richard, A comparative study of Bayesian models for unsupervised sentiment detection, CoNLL 2010
33. Li, Tao and Zhang, Yi and Sindhwani, Vikas, A nonnegative matrix tri-factorization approach to sentiment classification with lexical prior knowledge, In Proceedings of (ACL-IJCNLP), pages 244–252, 2009
34. Manning, Christopher D. and Raghavan, Prabhakar and Schtze, Hinrich, Introduction to Information Retrieval. Cambridge University Press, 2008
35. McCarthy, Diana and Koeling, Rob and Weeds, Julie and Carroll, John, Finding Predominant Word Senses in Untagged Text, Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), 2004
36. Bishop, Christopher M., Pattern Recognition and Machine Learning (Information Science and Statistics), 2006, Springer-Verlag New York, Inc.