Sentiment Analysis in Twitter with Lightweight Discourse Analysis

Subhabrata Mukherjee†, Pushpak Bhattacharyya†, Balamurali A.R.†‡

†Dept. of Computer Science and Engineering, IIT Bombay

‡IITB-Monash Research Academy, IIT Bombay

{subhabratam, pb, balamurali} @cse.iitb.ac.in

ABSTRACT

We propose a lightweight method for using discourse relations for polarity detection of *tweets*. This method is targeted towards the web-based applications that deal with *noisy* and *unstructured* text, like the *tweets*, and cannot afford to use heavy linguistic resources like *parsing* due to the frequent failure of the parsers to handle noisy data. Most of the works in micro-blogs, like *Twitter,* use a bag-of-words model that ignores the discourse particles like *but, since, although etc.* In this work, we show how *connectives, modals, conditionals* and *negation* can be used to incorporate discourse information in any bag-of-words model, to improve sentiment classification accuracy. We first give a linguistic description of the various discourse relations which leads to conditions in rules and features in SVM. Discourse relations and corresponding rules are identified with minimal processing - just a list look up. We show that our discourse-based bag-of-words model performs well in a noisy medium (*Twitter*), where it performs better than an existing Twitter-based application. Furthermore, we show that our approach is beneficial to structured reviews as well, where we achieve a better accuracy than a state-of-the-art system in the *travel review* domain. Our system compares favorably with the state-of-the-art systems and has the additional attractiveness of being less resource intensive.

Keywords

Sentiment Analysis, Discourse, Twitter, Connectives, Micro-blogs

1. INTRODUCTION

An essential phenomenon in natural language processing is the use of discourse relations to establish a coherent relation that links phrases and clauses in a text.

The presence of linguistic constructs like *connectives, modals, conditionals* and *negation* can alter the sentiment at the sentence level as well as the clausal or phrasal level. Consider the example, "*@user share 'em! i'm* **quite excited** *about Tintin, "despite"* **not really liking** *original comics. Probably because Joe Cornish had a hand in."* The overall sentiment of this sentence is *positive*, although there is equal number of positive and negative words. This is due to the connective *despite* which gives more weight to the previous discourse segment. Any bag-of-words model would be unable to classify this sentence without considering the discourse marker. Consider another example, "*Think i'll stay with the whole 'sci-fi'* **shit** *"but" this time...a* **classic** *movie."* The overall sentiment is again *positive* due to the connective *but,* which gives more weight to the following discourse segment. Thus, it is of utmost importance to capture all these discourse phenomena in a computational model.

Traditional works in *discourse analysis* use a discourse parser trained on Penn-Discourse-Treebank [1][2][3][4][5] or a Dependency Parser [6]. Many of the works [7][8] are centred around the Rhetorical Structure Theory (RST) proposed by [9] which tries to identify the relations between the nucleus and satellite in the sentence. Most of these theories are well-founded for *structured text,* and *structured* discourse annotated corpora are available to train the models. However, using these methods for micro-blog discourse analysis poses some fundamental difficulties.

Micro-blogs, like *Twitter*, do not have any restriction on the form and content of the user posts. Users communicate in the micro-blogs in an informal language. As a result, there are abundant *spelling mistakes, abbreviations, slangs, discontinuities* and *grammatical errors.* This can be observed in the given examples from real-life *tweets*. The errors cause natural language processing tools like *parsers* and *taggers* to fail frequently [10]. As the tools are generally trained on structured text, they are unable to handle the noisy and unstructured text in this medium. Hence most of the discourse-based methods, based on RST or parsing of some form, will be unable to perform very well in micro-blog data. Moreover, the web-based applications require a fast response time. Parsing, being a heavy-weight process, is not suitable to be used for real-time interactive systems.

Most of the previous research in micro-blogs, like *Twitter,* use a bag-of-words model with features like *part-of-speech information, unigrams, bigrams etc.* along with other domain-specific, specialized features like *emoticons, hashtags etc.* [11][12][13][14]. In *most* of these works, the *connectives, modals* and *conditionals* are simply ignored as stop words during feature vector creation. Hence, the discourse information that can be harnessed from these elements is completely discarded. In this work, we show how the *connectives, modals, conditionals* and *negation* based discourse information can be incorporated in a bag-of-words model to give better sentiment classification accuracy.

Our work builds on the discourse-related works of [15][16][17] and extends the idea further in the sentiment analysis of micro-blogs. We exploit the various features discussed in the Twitter specific works to develop a bag-of-words model, in which the discourse features are incorporated.

2. DISCOURSE RELATIONS CRITICAL FOR SENTIMENT ANALYSIS

A coherent relation reflects how different discourse segments interact. The interaction relations between discourse segments are listed in [15]. We pick up the discourse relations *Violated Expectations, Conclusions, Conditionals, Contrast*, *Modals* and *Negation* from their list for further analysis. The remaining relations like *Cause-Effect, Similarity, Temporal Sequence, Attribution, Example, Generalization and Elaboration* can be handled simply by taking the majority valence of the individual terms. These discourse relations do not provide any contrasting, inferential or hypothetical information and are easy to deal with.

- **Violating Expectations and Contrast**

These are the conjunctions that oppose or refute the neighboring discourse segment. We further categorize them into the following 2 sub-categories: *Conj_Fol* and *Conj_Prev.*

Conj_Fol is the set of conjunctions that give more importance to the discourse segment that follows. *Conj_Prev* is the set of conjunctions that give more importance to the previous discourse segment.

Example 1: *The direction was (not that great)$^-$, **but** still we loved$^+$ the movie.*

A bag-of-words model will find one positive and one negative sentiment and classify it as neutral, whereas the overall sentiment is positive. Here the final verdict is that we loved the movie, so the words following "*but"* should be given more weight.

Example 2: *India managed to win$^+$ **despite** the initial setback$^-$.*

This example has a similar drawback. Here, the emphasis is on the segment before "*despite"*.

- **Conclusion or Inference**

These are the set of conjunctions, *Conj_infer*, that tend to draw a conclusion or inference and hence the words following them should be given more weightage.

Example 3: *We were (not much satisfied)$^-$ with the greatly$^+$ acclaimed$^+$ brand X and **subsequently** decided to reject$^-$ it.*

Here, the final rejection matters more than the initial satisfaction making the final polarity negative, which cannot be captured by taking individual valence of terms.

- **Conditionals**

The *if…then…else* constructs depict situations which may or may not happen subject to certain conditions. In our work, the polarity of the discourse segment in a conditional statement is toned down, in *lexicon-based classification*. In *supervised classifiers*, the conditionals are marked as features. Such statements are not completely ignored as they bear some sentiment polarity.

Example 4: ***If** Brand X had improved$^+$ its battery life, it would have been a great$^+$ product.*

Here, the final polarity should be objective as we are talking of a hypothetical situation.

- **Modals**

Events that *have happened*, events that *are happening* or events that are certain to occur are called *realis events.* Events that have possibly occurred or have some probability to occur in the distant future are called *irrealis events.* Modals typically depict *irrealis events* or hypothetical situations. These constructs cannot be handled by taking a simple majority valence of terms. We further divide the modals into 2 sub-categories. *Strong_Mod* is the set of modals that express a higher degree of uncertainty in any situation. *Weak_Mod* is the set of modals that express lesser degree of uncertainty and more emphasis on certain events or situations.

Example 5: *That film **might** be good.*

*He **may** be a rising star.*

These strong modals are *not trustworthy* and are treated in the same way as we treat the *conditionals.*

Example 6: *I heard the movie is good, so you **must** go to watch that movie.*

*You **should** go to watch that awesome movie.*

As is evident from the above examples, these modals convey lesser degree of uncertainty.

- **Negation**

We consider the following negation operators: *not, neither, never, no* and *nor.* Negation is handled by considering a window of size 5 from a negation operator and reversing all the words in the window. This fails in the example below as it negates the "*like"* for Samsung as well.

Example 7: *I do (**not** like)$^-$ Nokia but I like$^+$ Samsung.*

We consider a negation window of size 5 and reverse all the words in the window, till either the window size exceeds or a *violating expectation* (Example: *but*) conjunction is encountered.

Discourse Relations	Attributes
Conj_Fol	*but, however, nevertheless, otherwise, yet, still, nonetheless*
Conj_Prev	*till, until, despite, in spite, though, although*
Conj_Infer	*therefore, furthermore, consequently, thus, as a result, subsequently, eventually, hence*
Conditionals	*If*
Strong_Mod	*might, could, can, would, may*
Weak_Mod	*should, ought to, need not, shall, will, must*
Neg	*not, neither, never, no, nor*

Table 1: List of Discourse Coherent Features

3. ALGORITHM TO HARNESS DISCOURSE INFORMATION

The discourse relations (identified in *Section 2*) are used to create a feature vector, according to *Algorithm 1*. In *Step 1*, we mark all the *conditionals* and *strong modals* which are handled separately by the lexicon-based classifier and the supervised classifier. In *Step 2* and *Step 3*, the weight of any word appearing before *Conj_Prev* and after *Conj_Fol* or *Conj_Infer* is incremented by 1. In *Step 4*, the polarity of all the words appearing within a window (*Neg_Window* is taken as 5), from the occurrence of a negation operator and before the occurrence of a *violating expectation* conjunction, are reversed. Finally, we get the feature vector {w_{ij}, f_{ij}, $flip_{ij}$ and hyp_{ij}} for all the words in the review. Here, the assumption is that the effect of any conjunction is restricted to continuous spans of text till another conjunction or the sentence boundary.

Let a user post R consist of 'm' sentences s_i $(i=1…m)$, where each s_i consist of n_i words w_{ij} $(i=1…m, j=1…n_i)$. Let f_{ij} be the weight of the word w_{ij} in sentence s_i, initialized to 1. Let A be the set of all discourse relations in *Table 1*. Let $flip_{ij}$ be a variable which indicates whether the polarity of w_{ij} should be flipped or not. Let hyp_{ij} be a variable which indicates the presence of a *conditional or a strong modal* in s_i.

Input: Review R
Output: w_{ij}, f_{ij}, $flip_{ij}$, hyp_{ij}

for $i=1\ldots m$
  for $j=1\ldots n_i$
    $f_{ij}=1$;
    $hyp_{ij}=0$;
1.   **if** $w_{ij} \in$ Conditionals *or* $w_{ij} \in$ Strong_Mod
      $hyp_{ij}=1$;
    end if
  end for
  for $j=1\ldots n_i$
    $flip_{ij}=1$;
2.   **if** $w_{ij} \in$Conj_Fol *or* $w_{ij} \in$ Conj_Infer
      for $k=j+1\ldots n_i$ *and* $w_{ij} \notin A$
        f_{ik}+=1;
      end for
    end if
3.   **else if** $w_{ij} \in$ Conj_Prev
      for $k=1\ldots j\text{-}1$ && $w_{ij} \notin A$
        f_{ik}+=1;
      end for
    end if
4.   **else if** $w_{ij} \in Neg$
      for $k=1\ldots Neg\_Window$ *and* $w_{ik} \notin Conj\_Prev$ *and* $w_{ik} \notin Conj\_Fol$
        $flip_{i,j+k}$=-1;
      end for
    end if
end for

Algorithm 1: Using Discourse to Create Feature Vector

4. FEATURE VECTOR CLASSIFICATION

We devised a lexicon based system as well as a supervised system for feature vector classification.

4.1 Lexicon Based Classification

The Bing Liu opinion lexicon [21] is used to find the polarity *pol*(w_{ij}) of a word w_{ij}. It contains 6800 words which are manually polarity labeled. Polarity of the review (*pos* or *neg*) is given by,

$$sign\left(\sum_{i=1}^{m}\sum_{j=1}^{n_i} f_{ij} \times flip_{ij} \times p(w_{ij})\right)$$
$$where\ p(w_{ij}) = pol(w_{ij})\ \ if\ hyp_{ij} = 0$$
$$= \frac{pol(w_{ij})}{2}\ \ \ if\ hyp_{ij} = 1$$
$$\ldots\ \ Equation\ 1$$

Equation 1 finds the weighted, signed polarity of a review. The polarity of each word, *pol*(w_{ij}) *being +1 or -1*, is multiplied with its discourse-weight f_{ij} (assigned by *Algorithm 1*), and all the weighted polarities are added. *Flip*$_{ij}$ indicates if the polarity of w_{ij} is to be negated. In case there is any *conditional* or *strong modal* in the sentence (indicated by $hyp_{ij} = 1$), then the polarity of every word in the sentence is toned down, by considering half of its assigned polarity ($\frac{+1}{2}$ or $\frac{-1}{2}$). Thus, if *good* occurs in the user post twice, it will contribute a polarity of $+1 \times 2 = +2$ to the overall review polarity, $if\ hyp_{ij} = 0$. In the presence of a *strong modal* or *conditional*, it will contribute a polarity of $\frac{+1}{2} \times 2 = +1$. All *stop words*, *discourse connectives* and *modals* are ignored during classification, as they have a zero polarity in the lexicon.

4.2 Supervised Classification

The Support Vector Machines have been found to outperform other classifiers, like *Naïve Bayes* and *Maximum Entropy*, in sentiment classification [30]. Hence, in our work, SVM's are used to classify the set of feature vectors {$flip_{ij}$, w_{ij}, f_{ij} and hyp_{ij}}.

Features used in the Support Vector Machines (SVM):

N-grams – Unigrams along with Bigrams are used.

Stop Words – All the stop words (like *a, an, the, is etc.*) and discourse connectives are discarded.

Feature Weight – In the *baseline bag-of-words* model, the feature weight has been taken as the feature frequency *i.e.* the number of times the unigram or bigram appears in the text. In the *discourse-based bag-of-words* model, the *discourse-weighted frequency* of a word is considered. *Algorithm 1* assigns a weight f_{ij} to every occurrence of a word w_{ij} in the post. If the same word occurs multiple times, the weights from its multiple occurrences will be added and used as a feature weight for the word.

Modal and Conditional Indicator – This is a *boolean* variable which indicates the presence of a strong modal or conditional in the sentence (*i.e.* $hyp_{ij}=1$).

Stemming – All the words are stemmed in the text so that "*acting*" and "*action*" have a single entry corresponding to "*act*".

Negation – A *boolean* variable ($flip_{ij}$) is appended to each word (w_{ij}) to indicate whether it is negated or not (*i.e.* $flip_{ij}=1$ or $flip_{ij}=0$).

Emoticons – An emoticon dictionary [23] is used to map each emoticon to a *positive* or *negative* class. Subsequently, the emoticon class information is used in place of the emoticon.

Part-of-Speech Information – The part-of-speech information is also used with a word.

Feature Space - We incorporate the discourse information extracted using *Algorithm 1* into two different feature spaces: lexeme space and sense space. In the *lexeme space* individual words are used as features; whereas in the *sense space*, the sense of the word (synset-id) is used in place of the word. A synset is a set of synonyms that collectively disambiguate each other to give a unique sense to the set, identifiable by the *synset-id*. This is beneficial in distinguishing between the various senses of a word.

For example, the word *bank* has 18 senses (10 Noun senses and 8 Verb Senses). Consider the two senses of a *bank* : 1) *Bank* in the sense of "*a financial institution*", identifiable by the synset "*depository financial institution, bank, banking concern, banking company"*, and 2) *Bank* in the sense of *relying*, identifiable by the synset *"trust, swear, rely, bank"*. Now, the first sense has an objective polarity whereas the second sense has a positive polarity. This distinction cannot be made in the lexeme feature space, where we consider only the *first* sense of the word.

5. EVALUATION

We performed experiments on three different datasets to validate our approach. 8507 tweets (*Dataset 1*) were crawled from Twitter based on a total of around 2000 different entities from over 20 different domains. These were manually annotated by 4 annotators into four classes: *positive, negative, objective-not-spam and objective-spam*. The *objective-spam* calss was subsequently ignored during evaluation. The Twitter API was used to collect another set of 15,214 tweets (*Dataset 2*) based on *hashtags*. Hashtags *#positive, #joy, #excited, #happy etc* were used to collect tweets bearing positive sentiment, whereas hashtags like *#negative, #sad, #depressed, #gloomy, #disappointed etc.* were used to collect negative sentiment tweets. *Travel Review Data* (*Dataset 3*) [19] contains 1190 polarity-tagged documents, with the positive and the negative class containing 595 documents each. All words in travel review documents were automatically sense-tagged using *IWSD* algorithm [20].

Evaluations are performed in *Dataset 1* and *Dataset 2* under a *2-class* and a *3-class* classification setting. In the 2-class setting, only *positive* and *negative* tweets are considered; whereas in the 3-class setting *positive*, *negative* and *objective-not-spam* tweets are considered. All the experiments in these two datasets are performed in the **lexeme feature space** using *lexicon-based classification* as well as *supervised classification*. The baseline system (for *Datasets 1* and *2*), is taken as *C-Feel-It* [18]. It is a rule-based system which implements a bag-of-words model using lexicon-based classification. The accuracy comparisons between C-Feel-It and the discourse system are performed under identical settings. The only difference between the two systems is the handling of *connectives, modals, conditionals* and *negation*, as indicated by *Algorithm 1*. *Table 2* shows the accuracy comparison between C-Feel-It and the discourse system, in Datasets 1 and 2, using lexicon-based classification. *Table 3* shows the accuracy comparison between the baseline SVM and SVM integrated with discourse features, in Datasets 1 and 2. All the SVM features discussed in *Section 4.2*, except the discourse features arising out of the incorporation of *discourse weighting*, *modal and conditional indicator* and *negation*, are used in the baseline SVM. A linear kernel, with default parameters (C=1, ϵ=0.0010), is used in LIBSVM [22] with 10-fold cross-validation.

Dataset	C-Feel-It	TwiSent	Stat. Sig. (%)
Dataset 1: 2-class	68.58	72.81	99.9
Dataset 1: 3-class	57.2	61.31	99.9
Dataset 2: 2-class	80.55	84.91	99.9

Table 2. Accuracy Comparison between C-Feel-It and Discourse System using Lexicon in Datasets 1 and 2 (Lexeme Space)

Dataset	C-Feel-It	TwiSent	Stat. Sig. (%)
Dataset 1: 2-class	69.49	70.75	90
Dataset 1: 3-class	63.11	64.23	90
Dataset 2: 2-class	91.99	93.01	95

Table 3. Accuracy Comparison between C-Feel-It and Discourse System using SVM in Datasets 1 and 2 (Lexeme Space)

Systems	Accuracy (%)	Stat. Sig. (%)
Only Unigrams	84.90	95
Only IWSD Sense of Unigrams [19]	85.48	90
Unigrams + IWSD Sense of Unigrams [19]	86.08	90
Unigrams + IWSD Sense of Unigrams + Discourse Features	**88.13**	-

Table 4: Accuracy Comparison in Travel Review Data (Dataset 3) using SVM in Sense Space

The *travel review* dataset [19] is used to determine whether our discourse-based approach performs well for structured text as well. An automatic word sense disambiguation algorithm, *IWSD* [20], has been used in [19] to auto-annotate the words in the review with their corresponding synset-id's. The same dataset is used in this work. A linear kernel, with default parameters, is used in the SVM with 5-fold cross-validation, similar to the compared system [19]. *Table 4* shows the performance of the discourse system along with the compared system using different features, on *Dataset 3*, using supervised classification.

6. DISCUSSIONS

Accuracy improvements over the baseline and the compared systems in all the datasets clearly signify the effectiveness of incorporating discourse information for sentiment classification. The bag-of-words model integrated *with discourse information* outperforms the bag-of-words model, *without this information*, under all the settings; although, the performance improvements vary in different settings. Z-Significance tests [24] were done and the confidence with which the accuracy changes were accepted to be statistically significant is shown in *Tables 2, 3* and *4*.

Accuracy comparisons between *C-Feel-It* and *Discourse System* are performed under a 2-class and a 3-class classification setting, using lexicon-based classification, in the lexeme space under identical settings - the only difference being the incorporation of discourse features. In *Dataset 1*, there is an accuracy improvement of around *4%* over C-Feel-It for both 2-class and 3-class classification. The discourse system accuracy at *72.81%* for 2-class classification is higher than that of the 3-class classification accuracy of *61.31%*. This shows that 3-class classification of tweets is much more difficult than 2-class classification.

Accuracy comparisons between *baseline SVM* and *Discourse System* are performed under a 2-class and a 3-class classification setting, using supervised classification, in the lexeme space. A similar feature set, except the discourse features, is used for both the systems. In *Dataset 1*, there is an accuracy improvement of *1%* in both the 2-class and 3-class classification, which has been found to be statistically significant. In *Dataset 2*, there is an accuracy improvement of *2%* over baseline SVM for 2-class classification. It is observed that in the 2-class setting, the discourse system performs better in the lexicon-based classification with an accuracy of *72.81%* compared to the supervised classification accuracy of *70.75%*. This is contrary to the common scenario in text classification, where the supervised classification system always performs much better than the lexicon-based classification. This may be due to the very sparse feature space, owing to the length limit of tweets (140 characters). The discourse system attains a high accuracy of *84.91% in Dataset 2* compared to the accuracy of *72.81%* in *Dataset 1* for lexicon-based classification. In supervised classification, the discourse system has an accuracy of *70.75%* in *Dataset 1* and *93.01%* in *Dataset 2*. In the *Travel review* dataset, lexicon-based classification yielded an accuracy improvement of *2%* for the discourse model over simple bag-of-words model, in *sense space*. In the SVM classification, in the *sense space,* under a 2-class setting, the discourse system achieved an accuracy of *88%* compared to *86%* accuracy of [19]. A similar feature set has been used in both the models, which attributes the performance improvement to the incorporation of discourse features in SVM.

The lexicon-based classification suffers from the usage of a generic lexicon in the *lexeme space*, where it cannot distinguish between the various senses of a word. The lexicons do not have entries for the interjections like *wow, duh etc.* which are strong indicators of sentiment. The frequent spelling mistakes, abbreviations and slangs used in the tweets do not have entry in the lexicons. For example, *love* and *great* are frequently written as *luv* and *gr8* respectively, which will not be detected. A spell-checker may help the system in this regard.

The supervised system suffers from a sparse feature space due to very short contexts. A concept expansion approach, to expand the feature vectors, may prove to be useful. This is due to the extensive world knowledge embedded in the tweets. For example, the tweet "*He is a Frankenstein*" is tagged as objective. The knowledge that *Frankenstein* is a negative concept is not present in the lexicon. The IWSD algorithm for automatic sense annotation has an F-Score of *70%* [20], which means many of the word-senses were wrongly tagged. A better WSD algorithm may improve the system performance in the travel review dataset.

In the absence of *parsing* and *tagging* information, due to the noisy nature of the tweets, the scope of the discourse marker has been heuristically taken till the sentence end or the next discourse marker. Consider the sentence, "*I wanted to follow my dreams and ambitions despite all the obstacles, but I did not succeed.*" Here, *want* and *ambition* will get a polarity *+2 each*, as they appear before *despite*; *obstacle* will get a polarity *-1* and *not succeed* will get a polarity *-2*. Thus the overall polarity is *+1,* whereas the overall sentiment should be *negative*. This is because we do not consider the *positional importance* of a discourse marker in the sentence and consider all the discourse markers to be equally important. A better method is to give a ranking to the discourse markers based on their *positional* and *pragmatic* importance.

7. FUTURE WORKS AND CONCLUSIONS

In this work, we showed that the incorporation of discourse markers in a bag-of-words model improves the sentiment classification accuracy by *2 - 4%*. This approach is particularly beneficial for applications dealing with noisy text where *parsing* and *tagging* do not perform very well.

Most of the works in micro-blogs, like *Twitter*, build on a bag-of-words model that ignores the discourse markers. We demonstrated an approach to incorporate discourse information to improve their performance, retaining the simplicity of the bag-of-words model. We validated this claim on two different datasets (manually and automatically annotated) from *Twitter,* where we achieved an accuracy improvement of *4%* for lexicon-based classification over an existing application [18], and 2% for supervised classification over the baseline SVM with advanced features. We also showed that our method fares well for structured reviews as well, where we achieved similar accuracy improvements over [19].

The method can be further improved by employing concept expansion to extend the context. A ranking of discourse features, based on their *positional importance* and *pragmatics*, and a better selection of their scope may improve the system performance.

8. ACKNOWLEDGEMENTS

We thank Akshat Malu, IIT Bombay, for his assistance during certain parts of this study.

9. REFERENCES

1. Marcu, Daniel. 2000. The Theory and Practice of Discourse Parsing and Summarization, MIT Press, Cambridge, MA.
2. Zirn, Cicilia and Niepert, Mathias and Stuckenschmidt, Heiner and Strube, Michael. Fine-Grained Sentiment Analysis with Structural Features. In Proc. of IJCNLP. 2011
3. Wellner, Ben and Pustejovsky, James and Havasi, Catherine and Rumshisky, Anna and Saur\'{\i}, Roser. Classification of discourse coherence relations: an exploratory study using multiple knowledge sources. Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue. 2006
4. Pitler, Emily and Louis, Annie and Nenkova, Ani. Automatic Sense Prediction for Implicit Discourse Relations in Text. In Proc. of ACL and IJCNLP. 2009
5. Elwell, Robert and Baldridge, Jason. Discourse Connective Argument Identification with Connective Specific Rankers. In Proc. of IEEE ICSC. 2008
6. Ng, Vincent and Dasgupta, Sajib and Arifin, S. M. Niaz. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In Proc. of the COLING/ACL: Poster Sessions, 2006
7. Taboada, Maite and Brooke, Julian and Tofiloski, Milan and Voll, Kimberly and Stede, Manfred. Lexicon-based methods for sentiment analysis. Computational Linguistics. 2011
8. Zhou, Lanjun and Li, Binyang and Gao, Wei and Wei, Zhongyu and Wong, Kam-Fai. Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In Proc. of EMNLP. 2011
9. Mann, William C. and Sandra A. Thompson. Rhetorical Structure Theory: Toward a functional theory of text organization. Text, 8 (3), 243-281. 1988
10. Dey, Lipika and Haque, Sk. Opinion Mining from Noisy Text Data. International Journal on Document Analysis and Recognition 12(3). pp 205-226. 2009
11. Alec, G.; Lei, H.; and Richa, B. Twitter sentiment classification using distant supervision. Technical report, Standford University. 2009
12. Read, Jonathon. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In ACL. The Association for Computer Linguistics. 2005
13. Pak, Alexander and Paroubek, Patrick. LREC. Twitter as a Corpus for Sentiment Analysis and Opinion Mining,2010
14. Gonzalez-Ibanez, Roberto and Muresan, Smaranda and Wacholder, Nina. Identifying sarcasm in Twitter: a closer look, In Proc. of ACL: short paper. 2011
15. Wolf, Florian and Gibson, Edward. Representing discourse coherence: A corpus-based study, Computational Linguistics, 31(2), pp. 249–287. 2005
16. Polanyi, Livia and Zaenen, Annie. Contextual valence shifters. In Exploring Attitude and Affect in Text: Theories and Applications, AAAI Spring Symposium Series. 2004
17. Taboada, Maite and Brooke, Julian and Voll, Kimberly. Extracting Sentiment as a Function of Discourse Structure and Topicality. Simon Fraser Univeristy School of Computing Science Technical Report. 2008
18. Joshi, A.; Balamurali, A. R.; Bhattacharyya, P.; and Mohanty, R. C-feel-it: a sentiment analyzer for microblogs. In Proc. of ACL: Systems Demonstrations, HLT '11, 127–132. 2011
19. AR, Balamurali and Joshi, Aditya and Bhattacharyya, Pushpak. Harnessing WordNet Senses for Supervised Sentiment Classification. In Proc. of EMNLP. 2011
20. Mitesh Khapra, Sapan Shah, Piyush Kedia, and Pushpak Bhattacharyya. Domain-specific word sense disambiguation combining corpus basedand wordnet based parameters. In Proc. of GWC'10, Mumbai, India. 2010
21. Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In Proc. of ACM SIGKDD. 2004
22. Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011
23. In Website. Retrieved on August 11, 2012, from Website. http://chat.reichards.net/
24. In Wikipedia. Retrieved on August 11, 2012, from Website http://en.wikipedia.org/wiki/Z-test