TwitSent: A System for Analyzing Sentiments in Twitter

[1]Nishtha Srivastava, [2]Neeshu Kumari
[1]Assistant Professor, [2] Assistant Professor,
[1]Computer Science and Engineering,
[1]Parul University, Baroda,India

Abstract : In this paper, we present TwiSent, a sentiment analysis system for Twitter. Based on the topic searched, TwiSent collects tweets pertaining to it and categorizes them into the different polarity classes positive, negative and objective. However, analyzing micro-blog posts have many inherent challenges compared to the other text genres. Through TwiSent, we address the problems of 1) Spams pertaining to sentiment analysis in Twitter, 2) Structural anomalies in the text in the form of incorrect spellings, nonstandard abbreviations, slangs etc., 3) Entity specificity in the context of the topic searched and 4) Pragmatics embedded in text. The system performance is evaluated on manually annotated gold standard data and on an automatically annotated tweet set based on hashtags. It is a common practise to show the efficacy of a supervised system on an automatically annotated dataset. However, we show that such a system achieves lesser classification accurcy when tested on generic twitter dataset. We also show that our system performs much better than an existing system.

IndexTerms **-Sentiment Analysis, Twitter, Micro blogs, Spam, Entity Specific Twitter Sentiment**

I. INTRODUCTION

Social media sites, like Twitter, generate voluminous amounts of data which can be leveraged to create applications that have a social and an economic value. In this paper, we present a hybrid system, *TwiSent*, to analyze the sentiment of tweets based on the topic searched in Twitter. Even though Twitter generates a large amount of data, a text limit of 140 characters per tweet makes it a noisy medium for text analysis tasks. Compared to other text genres like News, Blogs *etc.*, it has a poor syntactic and semantic structure. For example, consider the following tweet "*Had Hella fun today with the team. Y'all are hilarious! &Yes, i do need more black homies.....* ". Apart from the irregular syntax, the following sentence has other problems like *slangs, ellipses, nonstandard vocabulary etc.* A direct analysis of such noisy text using commonly applied Natural Language Processing (NLP) tools would be futile, as it may not give the desired results. Further, the problem is compounded by the increasing number of *spams* in Twitter like *promotional* tweets, *bot-generated* tweets, *random links* to other websites *etc.* In this paper, we tackle the following problems which are exclusive to a micro-blog genre like Twitter for assessing the sentiment content: **Twitter based spam, Spell checker for noisy text, Entity detection** and **Pragmatics.**

II. RELATED WORKS

[1] provides one of the first studies on sentiment analysis on micro-blogging websites. [2] and [4] both cite noisy data as one of the biggest hurdles in analyzing text in such media. [1] describes a distant supervision-based approach for sentiment classification. They use hashtags in tweets to create training data and implement a multi-class classifier with topic-dependent clusters. [2] proposes an approach to sentiment analysis in Twitter using POS-tagged n- gram features and some Twitter specific features like hashtags. Our system is inspired from *C-Feel-IT*, a Twitter based sentiment analysis system [3]. However, *TwiSent* is an enhanced version of their rule based system with specialized modules to tackle Twitter spam, text normalization and entity specific sentiment analysis.

There has not been much work around text normalization in the social media, although some work has been done in the related area of sms-es [5]. We follow the approach of [6] and attempt to infuse linguistic rules within the minimum edit distance [7]. We adopt this simpler approach due to lack of publicly available parallel corpora for text normalization in Twitter.

Unlike in Twitter, there has been quite a few works on general entity specific sentiment analysis. Many approaches have tried to leverage dependency parsing in entity-specific SA. [8] exploits dependency parsing for graph based clustering of opinion expressions about various features to extract the opinion expression about a target feature. We use dependency parsing for entity specific SA as it captures long distance relations, syntactic discontinuity and variable word order.

The works [1][12][13] evaluate their system on a dataset crawled and auto-annotated based on *emoticons* while [14] annotate the crawled data based on *hashtags*. We show, in this work, that a good performance on such a dataset does not ensure a similar performance in a general setting.

III. System Architecture

In this section, we give an overview of the complete system and define the functionality of each module. Figure 1 presents the architecture of the system.

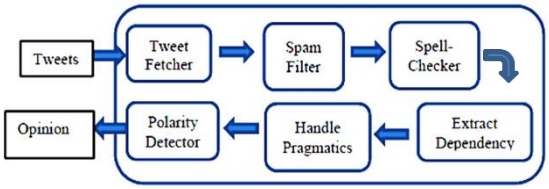

Figure 1. TwiSent Architecture Diagram

3.1 Tweet Fetcher and Polarity Detector

A Twitter API is used to obtain live feeds from Twitter. Based on the search string, we retrieve the latest 200 tweets in English. The tweets are in XML format which needs to be parsed to extract the tweet bodies. The tweet polarity is determined by a majority voting of four sentiment lexicons, following the approach in [3], namely, SentiWordNet, Subjectivity, Inquirer and Taboada.

3.2 Spam Filter

The Spam is the use of electronic messaging systems to send unsolicited bulk messages indiscriminately. [9] identifies three types of spam: Untruthful opinions, reviews on brands only and non-reviews. However, we provide a more detailed categorization of Twitter spams as: Re-tweets, Promotional tweets, Tweet containing links, Tweets in foreign language or having incomplete text, Bot-generated tweets, Tweets with excessive off-topic keywords or hashtags and Multiple tweets with same template.

The following set of features is used in the spam filter module:

1. *Number of Words / Tweet*	***8.*** *Freq. of First POS Tag*
2. *Average Word Length*	***9.*** *Freq. of Foreign Words*
3. *Freq. of "?" and "!"*	***10.*** *Validity of First Word*
4. *Numeral Character Freq.*	***11.*** *Presence / Absence of links*
5. *Frequency of hashtags*	***12.*** *Freq. of POS Tags*
6. *Frequency of @users*	***13.*** *Character Elongation*
7. *Extent of Capitalization*	***14.*** *Frequency of Slang Words*

Table 1. Spam Filter Features

3.3 Spelling Checker

Multiple spell-checkers are available today, but they are not effective in handling noisy text present in the social media. We give an overview of some of the most prevalent abbreviations and noisy text in Twitter. The list is compiled from the tagged tweets for this work and from [11]:

1. Dropping of Vowels - Example: btfl (beautiful), lvng (loving).
2. Vowel Exchange - Exchange between pairwise vowels due to phonetic similarity. Example: good vs. gud (o,u).
3. Mis-spelt words - Example: redicule (ridicule), magnificant (magnificent).
4. Text Compression - Example: shok (shock), terorism (terrorism).
5. Phonetic Transformation - Example: be8r (better), gud (good), fy9 (fine), gr8 (great).
6. Normalization and Pragmatics - Example: hapyyyyyy (happy), guuuuud (good).
7. Segmentation with Punctuation - Example: beautiful, (beautiful).
8. Segmentation with Compound Words - Example: breathtaking (breath-taking), eyecatching (eye-catching), good-looking (good looking).
9. Hashtags - Example: #notevenkidding, #worthawatch.
10. Combination of all - Example: #awsummm (awesome), gr88888 (great), amzng,btfl (amazing, beautiful) .

3.4 Handling Pragmatics

Pragmatics is a subfield of linguistics which studies how the transmission of meaning depends not only on the linguistic knowledge (e.g. grammar, lexicon etc.) of the speaker and listener, but also on the context of the utterance, knowledge about the status of those involved, the inferred intent of the speaker etc. We identified the different forms of pragmatics in Twitter as:

1. Happiness, joy or excitement is often expressed by elongating a word, repeating alphabets multiple times - Example: happppyyyyyy, goooooood.
2. Use of Hashtags - Example: #overrated, #worthawatch.

3. Use of Emoticons is common in social media and micro-blogging sites where the users express their sentiment in the form of accepted symbols. Example: □ (happy), □ (sad).
4. Happiness, joy, sorrow, hatred, enthusiasm, excitement, bewilderment etc. are also commonly expressed by capitalization where words are written in capital letters to express intensity of user sentiments. Full Caps - Example: I HATED that movie. Partial Caps- Example: She is a Loving mom. All these forms are given more weightage than other commonly occurring words by repeating them twice.

3.5 Entity Specificity

A tweet may have multiple entities and the user may express a different opinion expression regarding each entity there. Thus, it is of utmost importance to extract the specific opinion expression relating to a particular entity. Consider the tweet, "The film bombed at the box office although the actors put up a reasonable performance". Here the sentiment of the tweet with respect to film is negative whereas that with respect to the actors is positive. [8] proposes a Dependency Parsing based method to capture the association between any specific feature and the expressions of opinion that come together to describe that feature. The underlying hypothesis is that: More closely related words come together to express an opinion about a feature.

IV. Conclusion and Future Scope

In this paper, we introduced a Twitter based sentiment analysis system, *TwiSent*.
It is a multistage system with specialized modules to tackle the nuances of micro-blogging genres. Our results suggest that we outperform a similar Twitter based sentiment application by 14%. One of the major contributions of our work is in introducing Twitter based spams in the context of sentiment analysis. Our Spam Filter performs well not only as a part of the system but also as a stand-alone application. The Spell- Checker module helps in handling the noisy text, whereas the Pragmatics Handler can loosely capture the pragmatics in text which assists in improving the classification performance. The Entity-Specific module helps in capturing sentiment pertaining to the search entity. A more sophisticated approach to Spell- Checker, in presence of a parallel corpora, and Pragmatics Handler may add to the system performance. The system cannot capture sarcasm or implicit sentiment due to the usage of a generic lexicon in the final stage for classification. Overall, the paper not only highlights the issues associated with the micro- blogs but also presents an effective system to handle many of them. We also show that a superlative system performance on an auto-annotated dataset does not guarantee a similar or comparable performance on real-life micro-blog data

References

1. Alec, G.; Lei, H.; and Richa, B. 2009. Twitter sentiment classification using distant supervision. Technical report, Standford University.
2. Barbosa, L., and Feng, J. 2010. Robust sentiment detection on twitter from biased and noisy data. In Proceedings of the Computational Linguistics Posters, 36–44.
3. Joshi, A.; Balamurali, A. R.; Bhattacharyya, P.; and Mohanty, R. 2011. C-feel-it: a sentiment analyzer for microblogs. In Proceedings of ACL Demo Papers, HLT '11, 127–132.
4. Bermingham, A., and Smeaton, A. 2010. Classifying sentiment in microblogs: Is Brevity an Advantage, ACM 1833–1836.
5. Raghunathan, K., and Krawczyk, S. 2009. Investigating sms text normalization using statistical machine translation. CS224NProject Report, Stanford University.
6. Church, K. W., and Gale, W. 1991. Probability scoring for spelling correction. Statistics and Computing 1(2):91–103.
7. Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8.
8. Mukherjee, S., and Bhattacharyya, P. 2012. Feature specific sentiment analysis for product reviews. Part 1, Lecture Notes in Computer Science, Springer 7181:475–487.
9. Jindal, N. and Liu, B. 2008. Opinion spam and analysis. In Proceedings of the 2008 WSDM. pp. 219-229.
10. Liu, B., Lee, W., Yu S., and Li X. 2002. Partially supervised classification of text documents. In Proceedings of ICML.
11. Bieswanger, M. 2007. 2 abbrvi8 or not 2 abbrevi8: A contrastive analysis of different shortening strategies in English and german text messages. SALSA XIV.
12. Jonathon Read. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In Proceedings of the ACL Student Research Workshop.
13. Pak, Alexander and Paroubek, Patrick. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In Proceedings of the LREC.
14. Gonzalez-Ibanez, Roberto and Muresan, Smaranda and Wacholder, Nina. 2011. Identifying sarcasm in Twitter: a closer look, In Proceedings ofACL Short Papers.
15. Website. http://chat.reichards.net/. Retrieved Aug. 11, 2012
16. In Wikipedia. Retrieved on August 11, 2012, from Website http://en.wikipedia.org/wiki/A/B_testing