A LEXICON BASED ALGORITHM FOR NOISY TEXT NORMALIZATION AS PRE-PROCESSING FOR SENTIMENT ANALYSIS

Sudipta Roy[1], Sourish Dhar[2], Saprativa Bhattacharjee[3], Anirban Das[4]

[1]*Department of IT, Triguna Sen School of Technology, Assam University, Silchar*
[2]*Department of IT, Triguna Sen School of Technology, Assam University, Silchar*
[3]*Department of IT, Triguna Sen School of Technology, Assam University, Silchar*
[4]*Department of IT, Triguna Sen School of Technology, Assam University, Silchar*

Abstract

Sentiment analysis in the most general sense refers to the classification of a piece of text into either of the three classes–positive, negative or neutral–according to its polarity. The text may be an entire document, a paragraph, a sentence, a phrase or even a single word. Most of the literature on sentiment analysis is dedicated to well-formed text as found in the newspapers, journals and magazines. The unprecedented rise in popularity of the social media brought with it a vast sea of user generated content many of which convey subjective opinions on products, services, organizations, public figures and what not. But the textual data obtained from such sources are extremely noisy. They are characterized by numerous spelling and grammatical errors, as well as by the heavy usage of acronyms, abbreviations, shortened words and slang. The currently available Natural Language Processing (NLP) tools are not designed for handling such types of data. In this report we suggest a number of methods for making the data obtained from social media less noisy and more suitable for sentiment analysis.

Keywords: *Sentiment analysis, opinion mining, natural language processing, text mining, noise reduction.*

----------***----------

1. INTRODUCTION

Some of the most pioneering works in the field of sentiment analysis are those of [1] and [2]. A lot of contributions have also been made by [3] and [4] to *aspect based sentiment analysis* in the domain of product reviews. But as pointed out earlier, all of these works are targeted towards well-formed text data.

Not much work has been done in sentiment analysis of noisy data obtained from the social media platforms such as Twitter and Facebook. Dey and Haque [5] initially employed a semi-supervised method to learn domain knowledge from a training repository which contains both noisy and clean text. Thereafter they employed localized linguistic techniques to extract opinion expressions from noisy text. They developed a system based on this approach, which provides the user with a platform to analyze opinion expressions extracted from repository. But the problem with this approach is that it is very tough to obtain a reasonable size of domain data for the semi-supervised learning of domain knowledge. Even if a large data set is collected it will be mostly noisy and clean data will be extremely difficult to gather. And without clean data this approach cannot be used satisfactorily. Further, the authors used a Java spell checker named Suggester along with a weighted function based on domain frequency of a word to suggest the correct spelling of a possibly misspelled word. This method might lead to a large number of correct words to be unnecessarily replaced by some other similar words. Thus instead of solving the problem, it may give rise to other problems.

Han et al. [6] used a classifier to detect lexical variants, and generate correction candidates based on morphophonemic similarity. Both word similarity and context are then exploited to select the most probable correction candidate for the word. The main advantage of their proposed method is that it doesn't require any annotations. One major shortcoming of their approach is that the normalized output must be a single-token word which means “smoking” would be normalised to “smoking” but “imo” will not be normalised to “in my opinion”. Our method will be addressing each of the above issues.

The paper is organized as follows: Sections 2 and 3 describe the dataset and the supplementary resources used respectively. The algorithm is proposed in Section 4. Results and discussions are presented in Section 5. Finally the conclusion is drawn in Section 6.

2. THE DATASET

Our dataset consists of 15000 comments crawled automatically as well as downloaded manually from Twitter, Facebook and Mouthshut. Out of these 7000 comments are for training and the rest 8000 are for testing. The training dataset is further divided into two sets, 5000 for training only and 2000 for validation. Each comment in the training set is annotated with five

attributes–mood, degree, category, sub-category and detailed category. Mood may be either positive, neutral or negative. Degree may be one of -2, -1, 0, +1 or +2 which corresponds to very negative, negative, neutral, positive and very positive respectively. The values taken by the other three attributes will be described in the following section. A point to be noted here is that the comments are annotated for facilitating invocation of machine learning algorithms for sentiment analysis and have no role to play in the noisy text normalization algorithm described in section 4.

3. SUPPLEMENTARY RESOURCES

In addition to the dataset mentioned above we also use a number of other useful resources such as stop words list, dictionary of English words and several manually built lexicons. Here we will discuss all such resources one by one.

3.1 Complaint Taxonomy

The complete taxonomy of categories, sub-categories and detailed categories is given below. We call it the Complaint Taxonomy because a significant portion of our dataset deals with consumer complaints.

- Complaint/Praise
  - Network
    - Coverage
    - Speed
    - Availability
    - Call/Packet drop
    - Poor audio/video quality
  - Service
    - Service Request pending for many days
    - Too many follow ups are required
    - Rude behavior of the technician
    - Knowledge of the technician
    - Quality of Service being suboptimal
    - Long turn-around time to fix fault
    - Long wait for connecting to Customer Care
    - Website access problem
    - Did not keep the customer informed
  - Communication
    - Incorrect Name or Address
    - Incorrect item list
  - Equipment
    - Faulty handset
    - Faulty dongle
    - Faulty Set Top Box
    - Installation/Activation Error/Delay
    - Configuration Problem
    - Compatibility Problem
  - Mis-sale
    - Misleading Advertisement
    - Misleading Webpage content
    - Misled by Customer Service
    - Misled by Advisor
    - Less than what was promised
  - Billing
    - High billing
    - Bill not received in time
    - Rebate not given
    - Missing Payment
    - Billed even though service was not used
    - Problem with Standing Instruction (SI)
    - Wrong Charging
    - Wrong Penalty Calculation
    - Harassment for recovery
- Query
  - Service Line
    - Basic Service
    - Broadband
    - DTH
    - 3G
    - 4G
    - LTE
    - VAS
  - Product
    - Various Plans
  - Customer Service
    - Address
    - Phone Number
    - E-mail address
    - Website address
- Request
  - Demo
    - Various Equipments
  - Service
    - Upgrade Requirements
    - MNP
- Statement
- Advertisement

3.2 Non-Dictionary to Dictionary Lexicon

A lexicon has been created manually for substituting the most commonly occurring abbreviations, acronyms, emoticons, shortened and misspelled words to their dictionary equivalents. The lexicon has two columns, the one on the left contains the non-dictionary word and the other one on the right contains its dictionary form. At present the lexicon consists of a total of 5830 entries. Given below are a few example entries:

Table 1: Non-Dictionary to Dictionary Lexicon

Non-Dictionary Word	Dictionary Equivalent
Owesome	Awesome
Phab	Fabulous
Probs	Problems
:)	Happy

:(	Sad

3.3 Stop Words List

The stop words list consists of 571 words. It has words such as a, about, ain't, become, believe, hadn't, when, your, zero, etc. These words do not carry any sentiment and are equally likely to appear in positive comments as many times as in either of neutral or negative comments. This list is used for removing such words from the dataset.

3.4 English Dictionary

The English dictionary is a simple text file consisting of 3,00,249 words with one word per line. In addition to the usual dictionary words this file also has names of countries, major cities and even some Sanskrit words such as adharma.

4. THE PROPOSED ALGORITHM

We propose the following algorithm for cleaning the noisy text data obtained from the social media sites:

Input: A set of raw comments $<c_1, c_2, ..., c_n>$ belonging to the dataset D.

Output: A set of processed comments $<c_1', c_2', ..., c_n'>$ corresponding to $<c_1, c_2, ..., c_n>$.

```
begin
for each comment ci in D
do:
      remove URLs and twitter usernames;

      replace a sequence of two or more
      consecutive punctuation characters
      with the first punctuation character;

      replace a sequence of three or more
      consecutively repeating alphabetic
      character by two characters of the
      same alphabet;

      replace each occurrence of
      consecutive multiple whitespace
      characters by a single whitespace
      character;

      for each word wj in ci
      do:
            if (wj is not a dictionary
            word)
            then
                  replace wj with its
                  dictionary
                  equivalent wj';
      end for
      remove all the stop words;
      return ci';

end for
end
```

5. RESULTS AND DISCUSSION

In the first step all the URLs and twitter usernames (those starting with `@') are removed. In the next step a sequence of consecutive punctuation characters such as `???????' and `.....' are replaced by `?' and `.' respectively. Note that a sequence such as `\#@?' will be replaced only by the first punctuation character encountered in the sequence i.e. `\#' but this substitution in no way effects the sentiment carried by the comment. Then, three or more consecutive occurrences of the same alphabet is replaced by two characters of that alphabet. As for example `cooool' is replaced by `cool'. After that each occurrence of multiple consecutive whitespace characters is replaced by a single whitespace character. The last two steps involve the substitution of non-dictionary terms with their dictionary equivalents and removal of stop words with the help of the supplementary resources already described.

6. CONCLUSIONS

Earlier works on sentiment analysis mainly dealt with clean text as encountered in the newspapers and English literature but more recently with the advent of social networking sites such as Facebook and Twitter, the focus has shifted to the analysis of noisy text. In this paper we have proposed an algorithm for preprocessing of such noisy text data for making them more suitable for sentiment analysis. Our algorithm successfully addresses the issues faced by the earlier related works.

REFERENCES

[1] P. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002.

[2] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP*, pp. 79–86, 2002.

[3] M. Hu and B. Liu, "Mining and summarizing customer reviews," *Proceedings of the tenth ACM SIGKDD* international *conference on Knowledge discovery and data mining New York, NY, USA*, 2004.

[4] A. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing Morristown, NJ, USA*, no. October, pp. 339–346, 2005.

[5] L. Dey and S. Haque, "Opinion mining from noisy text data," Proceedings *of the second workshop on Analytics for noisy unstructured text data*, no. iv, pp. 83–90, 2008.

[6] B. Han, P. Cook, and T. Baldwin, "Lexical Normalisation for Social Media Text," *ACM* Transactions *on Intelligent Systems and Technology*, 2012.