Transformation of Sentimental Impact for Documents

Rutuja Karkar[1], Shweta Nagdev[2], Pranjal Gangrade[3], Deepali D. Gatade[4]

[1,2,3] B.E. (Computer Engineering), Sinhgad College of Engineering, Pune, Maharashtra, India
[4]Assistant Professor, Dept. of Computer Engineering, Sinhgad College of Engineering, Pune, Maharashtra, India

***

Abstract - *The growth of the web and social networking sites have been emerged into a huge volume of user reviews and opinions about particular aspects of products or services. People like to share their experiences, thoughts, opinions, feelings, and preferences according to their understanding and observation about the services. Their point of view or impression may be positive, negative or neutral. This opinion is used for identifying trends, user interest, political polls, and market researches, enhancing the user experience by presenting the things of their own interest and to influence them towards a particular direction. For one particular aspect, one may have a positive opinion while some other may have a negative opinion at the same time. Thus, classifying opinion and sentiment of people is a difficult task. Furthermore, the shared reviews and feelings are not in specifically structured format, thus identifying its positivity or negativity perspective automatically, is also inconvenient. Therefore, analysis of an unstructured format of text and extract the information for determining the user's sentiments requires special machine learning techniques and semantic algorithms for their classification. To find solution for above problem, we proposed algorithm to evaluate Quality of document and improve its impact. Algorithm will take input as document and it will find sentimental words in it. These words are used to evaluate whether it gives positive, negative or neutral impact. Thus the Quality of document can be improved by either increasing or decreasing the Sentimental Impact.*

Key Words: ***NLTK, PyDictionary, KNN, Prediction & Recommendation***

1. INTRODUCTION

1.1 Sentiment Analysis

Determining the emotional tone behind a piece of text is known as Sentimental Analysis. Sentiment analysis is a task of identifying positive and negative opinion, emotion and evaluation in text available over the social networking websites and the World Wide Web. People like to share their experiences, thoughts, opinions, feelings, and preferences according to their understanding and observation about the services. Their point of view or impression may be positive, negative or neutral. This opinion is used for identifying trends, user interest, and prediction of stock markets, political polls, and market researches. For one particular aspect, one may have a positive opinion while some other may have a negative opinion at the same time.

1.2 Prediction & Recommendation

When user uploads document then natural language processing is apply on it. NLP gives token of words present in document. These tokens are passed to the Sentimental analysis function. It gives result of sentimental impact of document. Also it shows sentimental words in document. Application will predict the appropriate class for the document using Machine Learning KNN algorithm.

2. SCOPE

This application basically works on text documents of format .txt, .doc. Quality of documents basically depends upon good and optimized tools thus analysis is carried out using a strong natural language processing library named as nltk (Natural Language Toolkit) and it is an optimized, fast and efficient library till date. Machine learning technique named as kNN (k-Nearest Neighbors) algorithm which gives suggestion to user according to their input document. This suggestion can be used for transformation of document or user can give their own choice and thus application works accordingly.

3. RELATED WORK

Raksha Sharma, Arpan Somani, Lakshya Kumar, Pushpak Bhattacharyya [1], Sentiment Intensity Ranking among Adjectives Using Sentiment Bearing Word Embeddings, focuses on identification of intensity ordering among polar (positive or negative) words which have the same semantics can lead to a fine grained sentiment analysis. In this paper, author propose a semi supervised technique that uses sentiment bearing word embedding to produce a continuous ranking among adjectives that share common semantics. Word2vec is the state of-the-art for intensity ordering task.

Preety, Sunny Dahiya [2], Research Article Sentiment Analysis Using SVM and NAIVE BAYES Algorithm, Unsupervised Learning, Multi-Layer Perceptron (MLP), Naive Bayes (NB), Weka2 toolkit, K-mean algorithm Above methods has been applied on mobile review. Author proposed a method using na¨ıve bayes and modified k means clustering and found that it is more accurate than naive bayes and support vector machine techniques individually.

Shenghua Bao, Shengliang Xu, Li Zhang, Rong Yan, Zhong Su, Dingyi Han, and Yong Yu [3], Mining Social Emotions from Affective Text, To achieve this, author first present two baseline models: 1) emotion-term model that uses Naive Bayes to model social emotion and affective terms via their co-occurrences and 2) a LDA topic model which utilizes the term co-occurrence information within a document and discovers the inherent topics within affective text. Then, they describe the proposed emotion-topic model that can jointly estimate the latent document topics and emotion distributions in a unified probabilistic graphical model. : This paper presents and analyzes a new problem called social affective text mining, which aims to discover and model the connections between online documents and user-generated social emotions.

4. PROPOSED WORK

The Transformation of Sentimental Analysis system is separated into various modules for the implementation. The user uploads the document into the system along with the type of document (example article, report, speech, mails, etc.). After uploading the document our system will open the file uploaded and the whole document is broken into tokens. The tokens are tagged with their part of speech from which our system works only on adjectives and adverbs i.e. the sentimental words. These sentimental words are appended into a new file. The words are also analyzed if they are positive negative or neutral with the help of nltk library. The synonyms for each word are stored into Dictionary and for synonyms PyDictionary library is used. All the synonyms which is given by nltk library are then arranged in ascending order of their sentiment. As there are five classes in our system, for each word there are five synonyms. By averaging the sentiment of all sentimental words the Sentiment of whole document is displayed to the user in the form of pie chart .The count of words i.e. positive, negative or neutral is used for displaying Impact to the user. After Display of Result if the user wishes to change the sentiment of Document he can change it by selecting class to which the sentiment is to be converted. By taking the Difference of actual class and user selected class the appropriate synonym is found out and replaced with the actual word. This document is then analyzed for giving new result and changed document is given to user.

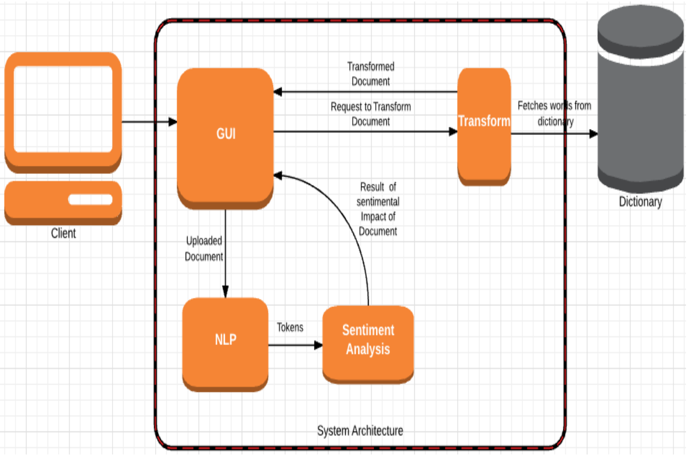

Figure 1: System Overview Diagram

5. IMPLEMENTATION TECHNIQUE

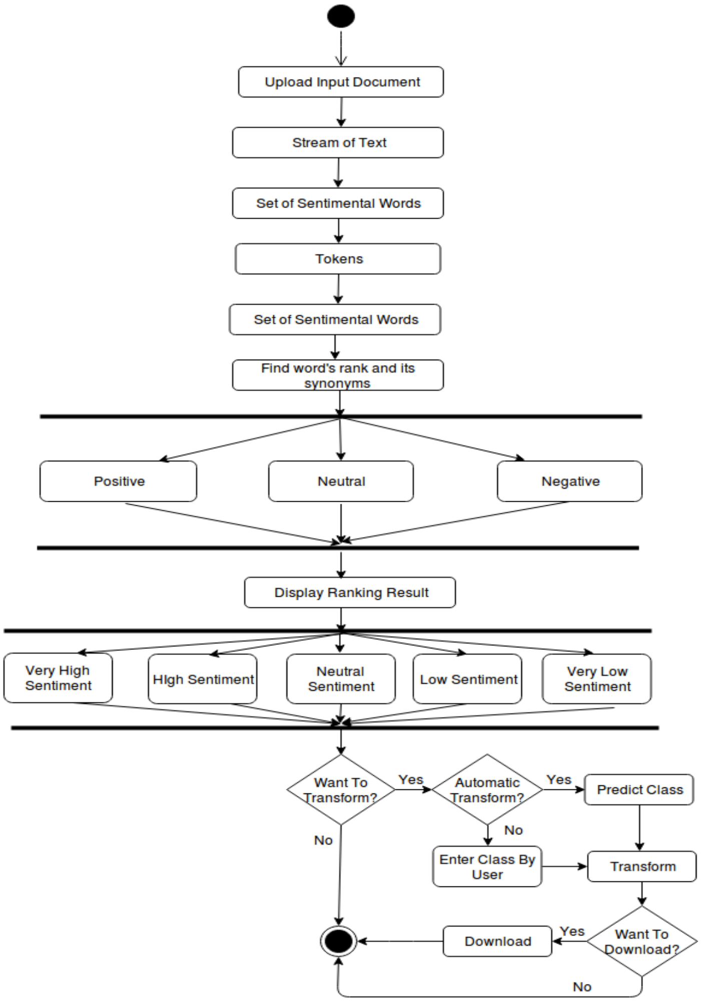

Figure 2: System Architecture

5.1 NLTK

NLTK stands for Natural Language Toolkit. NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.” Natural Language Processing with Python provides a practical introduction to programming for language processing. The words are also analyzed if they are positive negative or neutral with the help of nltk library. The words are tagged with their part of speech using nltk method pos_tag ( ) from which our system works only on adjectives and adverbs i.e. the sentimental words. Method polarity_score( ) of class Sentiment Intensity Analyzer gives the sentimental score of each word.

5.2 KNN (What is k-Nearest Neighbors)

The model for kNN is the entire training dataset. When a prediction is required for a unseen data instance, the kNN algorithm will search through the training dataset for the k-most similar instances. The prediction attribute of the most similar instances is summarized and returned as the prediction for the unseen instance. The kNN algorithm is belongs to the family of instance-based, competitive learning and lazy learning algorithms.

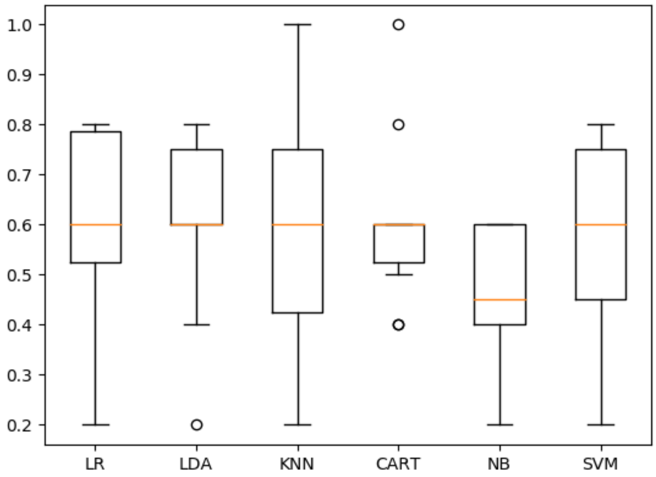

Figure 3: Algorithm Comparison

5.2.1 Dataset Used for kNN

The dataset is being provided through a .csv (Comma-separated values) file in which there are 120 documents. Documents are of various types such as blogs, formal letters, articles, emails etc. Out of which training phase has 80% of the documents to train the machine and testing phase has 20% of the documents to test the accuracy.

In our system dataset consists of seven fields, of which first is class of input document i.e. very low, low, neutral, high or very high sentiment. The second field is the type of document i.e. article, blog, speech etc. The other five fields contains the percentage of each class(i.e. one field of percentage of class 1 i.e. very low sentiment, one field of percentage of class 2 i.e. low sentiment etc.). Using this dataset our system takes type of document and percentage of each class values and predicts the class of the document.

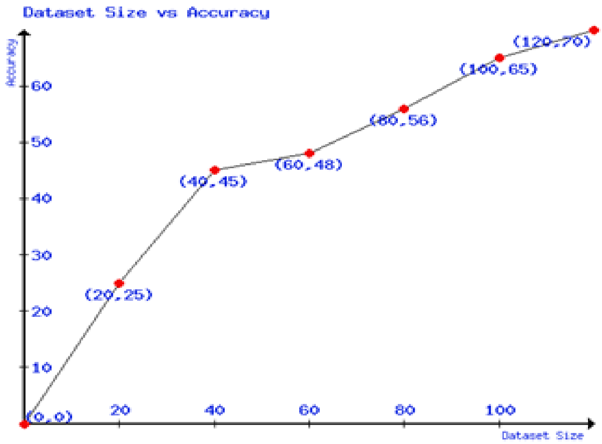

Figure 4: Dataset Size vs. Accuracy Graph

This graph shows the information about accuracy variation with respect to the change in the size of the dataset. Thus it shows that as the size of dataset increases the accuracy of kNN algorithm also increases. Hence Accuracy of kNN algorithm is directly proportional to the size of dataset.

5.3 Transformation

Approach 1:

1. Append all Sentimental words in list.

2 .Confirm class in which document will be transformed.

3. Select the synonym of each sentimental word belonging to that class and append it in another list.

4. Replace old sentimental words with new one.

Approach 2:

1. Append all Sentimental words in list.

2. Confirm class in which document will transformed.

3. Take difference of Sentimental result and transform class.

4. Select the synonym of each sentimental word according to the difference and append it in another list.

5. Replace old sentimental words with new one.

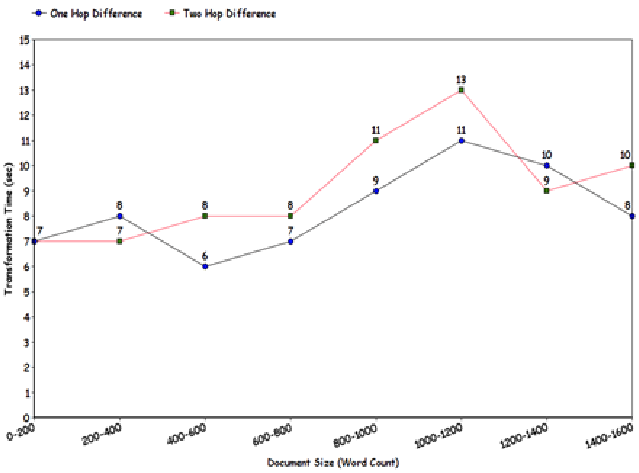

Figure 5: Transformation time vs. document size Graph

In the above graph the red line shows the Transformation time(in sec) vs. document size(in words) plot for two hop difference i.e. from result class to two class above the result class, while the blue line shows the Transformation time(in sec) vs. document size(in words) plot for one hop difference i.e. from result class to the one class above the result class. The time for both one hop difference and two hop differences goes on increasing linearly hence algorithm used for transformation of sentiment of document can be said as stable.

6. RESULTS

While transforming document Approach 2 gives better result than approach 1. As Approach 2 transforms Sentimental word between its suitable class margin. But in Approach 1 all sentimental words are replacing with same class.

For example, sentimental word from class 1 can change to class 5 internally which may affect impact of sentence due to that word. While in Approach 2 sentimental words of class 1 can transforms to class 2 or 3 (as per range or difference) which gives better result than previous approach.

Hence we used Approach2 for implementation.

From Algorithm Comparison results, it would suggest that both linear discriminate analysis and K-Nearest Neighbor are perhaps worthy of further study on this problem.

Time complexity depends on the number of data and features.

LDA time complexity is O(Nd^2)

k-NN time complexity is O(Nd).

k-NN should run incrementally faster than LDA as you add more dimensions to your problem.

Also, k-NN time complexity is pretty much insensitive to the number of classes in most implementations. LDA on the other hand has a direct dependence on that.

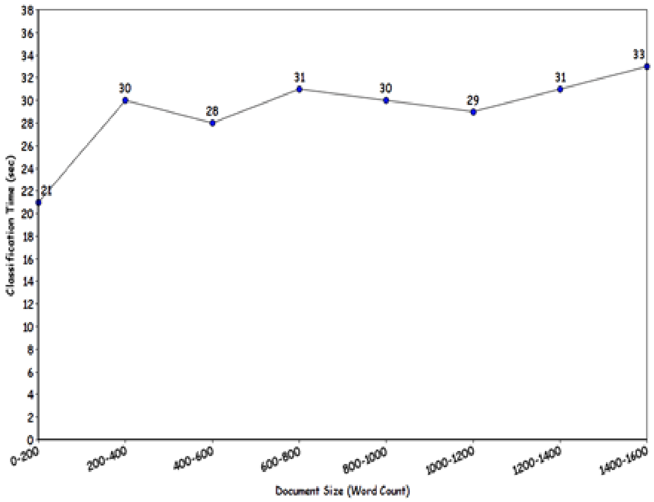

Figure 5: Classification time vs. document size Graph

The above graph shows that as we increase the document size (In words) the Classification time(in sec) of sentimental analysis phase goes on increasing linearly by a small difference. Hence the algorithm implemented for sentimental analysis phase of document can be said as stable.

7. CONCLUSION AND FUTURE WORK

Hence by using the above approach successful analysis of sentiment impact of document is performed and the document is transformed accordingly as per the choice or need of user. Here the User has two options to transform the document i.e. either manually or automatically. If the user choses Manual transformation then the user has to select the class to which document has to be converted. In Automatic transformation the document automatically gets converted to the class predicted by the system using machine learning algorithm.

The system can further be improved by incorporating accounts for each user and then by referring the past choice history of user's suggestions can be given about which class of sentimental impact the user should convert its document. The accuracy of the algorithm can be increased. More Sentimental words can be identified. Grammar check can be performed on the document to give better results. Application can be further improved by working on documents of PDF file format.

REFERENCES

[1] Raksha Sharma, Arpan Somani, Lakshya Kumar, Pushpak Bhattacharyya," Sentiment Intensity Ranking among Adjectives Using Sentiment Bearing Word Embeddings", IITB Monash Research Academy, IIT Bombay, July 2011.

[2] Preety, Sunny Dahiya, "Research Article Sentiment Analysis Using SVM and NA¨IVE BAYES Algorithm ", 5th International Conference on Informatics, Electronics and Vision (ICIEV), 2016.

[3] Shenghua Bao, Shengliang Xu, Li Zhang, Rong Yan, Zhong Su, Dingyi Han, and Yong Yu,"Mining Social Emotions from Affective Text", Sensys08, November 57, 2008, Raleigh, North Carolina, USA.

[4] Sunny Kumar, Dr. Paramjeet Singh, Dr. Shaveta Rani. "Study of Different Sentimental Analysis Techniques: Survey", Volume 6, Issue 6, June 2016 in International Journal of Advanced Research in Computer Science and Software Engineering.

[5]http://scikit-learn.org/stable/modules/neighbors.html

[6] http://www.nltk.org/book/

[7] http://flask.pocoo.org/docs/0.12/