Volume 4, Issue 9, September 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Sentiment Analysis and Classification Using Sentiment Sensitive Thesarus

Tinu Reshma R
Dept. of Computer Science
Federal Institute of Science and Tech.
Angamaly, India

Paul. P. Mathai
Dept. of Computer Science
Federal Institute of Science and Tech.
Angamaly, India

Abstract -- Sentiment analysis is one of the most prominent cases of artificial intelligence. Identifying emotions by a computer is a difficult task. In this paper, multiple features are extracted using a pipeline of stages called the EmoLib and then they are used to classify the text given. To eliminate the condition of classifying the document only in the domain trained, a cross-domain thesarus is introduced which analyses words and their affect in each domain. This paper enables easy analysis and classification of sentiment along with providing a cross-domain quality to the text.

Keywords: Sentiment, Sentiment Classification, feature selection, Cross-domain, EmoLib

I. INTRODUCTION

Sentiment analysis determines the attitude/opinion/emotion which is being expressed by a person regarding a particular topic. It makes use of the concepts of natural language processing and text analytics in identifying and extracting subjective information from the source materials in the document given. In the recent years it is found that there is a rise of social media that includes blogs and social networks that has increased interest in the concept of sentiment analysis. The relevance in identify ing the new opportunities and then managing their requirements, thus allowing people from different walks of lives to view the different data given in different forms of online opinion. The technique enhances the words that are expected to highlight the words that indicate the sentiment and then searches for possible relations among the words if any to signify any kind if difference in the actual sentiment itself. And if any such combinations exists they are definitely identified. It is possible to have positive, negative and neutral sentiment for words. Such different sentiments are identified by analysing sub sequent concepts regarding the same topic in better understanding of words.

Emotions are difficult to interpret and more difficult is the process of understanding the appropriate emotions from the word by a computer. This process is made easier by the process of emotion annotation in words and may even tag the same meaning for multiple words, or multiple meanings for a single word. This drawback provides an importance in the process itself so as to signify the actual emotion and then to provide the appropriate processing required for the text. Thus it is actually required for proper transition and annotation to be synchronised for the processing of text thus avoiding conflicts among different methods in annotation. It is required for a wide range of conversion of annotations from larger set to a smaller set otherwise from a set that is smaller to a larger set.

Articles should be classified irrespective of the domain in which the actual data belongs to. Domain meaning the category or the area or the concept that is described by the text. Thus irrespective of the domains the given text is expected to provide a classification of sentiment[1]. Sentiment annotation in text classification is basically done in the area of Natural Language Processing. What the writers mean by writing a specific text, the attitude of the writers, their intentions and the different area of their inclination with respective to different topics.

The section that follows is a survey on the possible methods in sentiment analysis and then classification. The section 3 then provides a mechanism to do effective sentiment analysis and classification. It also takes care of the detailed pipeline architecture and then extending it to a stage of cross-domain sentiment analysis by introducing a thesarus. The section 4 concludes the work and the possible future works. The next section has the references for the work.

II. LITERATURE SURVEY

Sentiment classification on being done sentence wise and is discussed in the work by Peifeng Li, Qiaoming Zhu, Wei Zhang in [2] where the corpus processed for dependency relations among words. Dependency relations are converted to dependency tree.Then the dependency tree is pruned to numerous sub-trees then polarity is identified using SVM. The next in this list is the work by Chunxi Liu, Li Su, Qingming Huang, Shuqiang Jiang [3] where a new framework for classifying sentiments in news stories. A semi-supervised learning with a set of selected words as features are done on news stories to classify the respective sentiment. Multimodal fusion performs sentiment ranking, sentiment mining using a novel framework for sentiment mining, affinity propagation and PageRank algorithm for representing sentiment.

Opinions on social issues was identified in the work by Mostafa Karamibekr, Ali A. Ghorbani in [4]. Generally based on adjectives, adverbs and nouns, this mechanism has identified verbs to be opinion indicators too. Verb based classification of sentiment is defined on the semantics in the text as verb identifies the semantics of the text.

The next work gives the difference between social issues and different products. This work published by Mostafa Karamibekr, Ali A. Ghorbani. [5] classifies social issues on the basis of sentiment. Semantic components assign polarity on specific social issues to the document that comprises of the semantic components. Verbs are the components that are analysed well. The next work discussed is by Costin-Gabriel Chiru, Asmelash Teka Hadgu [6] that does sentiment based text segmentation are considered relevant for sentiment analysis. There are disadvantages of identifying the relavant features in product reviews. Identifying, extraction and then assigning sentiment polarity.

Another important method proposed by Eric T. Nalisnick and Henry S. Baird [7] is identifying sentiment in Shakespearen Plays. Sentiment analysis is done by means of an AFINN word list. Each word is labelled by means of a valence with a polarity varying from +5 to -5 with +5 denoting the most positive words and -5 denoting the most negative words. The next proposal on this is about developing a Part-of-speech model done by Ms.K.Mouthami, Ms.K.Nirmala Devi, Dr.V.Murali Bhaskaran in [8] which uses a vector model to represent the features in the document. Part-of-speech tags that are the retrieved information from the document identifies the sentiment expression. Adjectives are again considered along with subjectivity in the sentence with a measure of high correlation. The next similar type of work is the twitter information classification has been discussed by Rabia Batool , Asad Masood Khattak , Jahanzeb Maqbool and Sungyoung Lee in their work [9] that works by means of domain specific classification. Overlapping data are preprocessed for the required information by filtering not required data. Submappings are prodeced that provide a complete character listings of people like a politician or an actor.

III. THE PROPOSED WORK

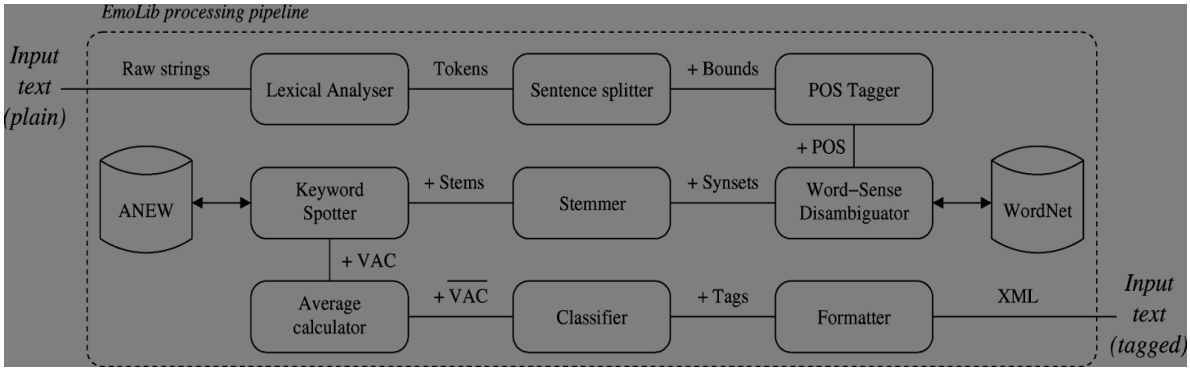

Fig 1: EmoLib pipeline with all stages.

The work proposed is a process of sentiment classification [10] that will give the a class of positive and negative as the two classes in the result of the entire process. An input text document is provided to the classifier which is then passed through each step in the EmoLib pipeline given in 3.2. The output of the pipeline is a classified input based on the sentiment. Here the denoted sentiments are positive and negative. The dataset that is used for training the classifier is the movie dataset. Since there exists a restriction on the number of different domains that are possible, hence to avoid that difficulty, a cross-domain sentiment sensitive thesarus is used for the classification. The first phase is the EmoLib pipeline and it is followed by the thesarus.

A. The EmoLib pipeline

The Pipeline is comprised of a series of stages that convert the input to a series of features and then classify the text based on the extracted features. As it is a pipeline, the stages, each of them take input from the previous preceeding block and the output is given to the next following block. The output of the pipeline is the class of the text given as input. The pipeline with the different stages are given below[12].

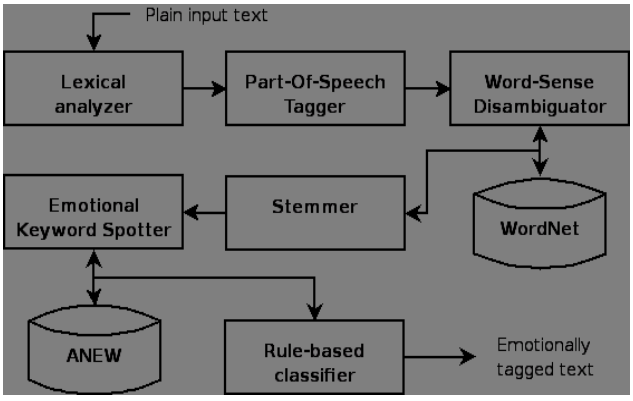

Fig 2: EmoLib basic pipeline

The following are the different pipeline stages

1) Lexical analysis:

This is the first phase of the pipeline. Its main function is the process of tokenizing the input text. Once the input text is given, the lexical analyser outputs the tokens corresponding to the text given as input. The module enables in identifying the tokens for every word in the text. Another important function is the process of eliminating the stop words in the text document. Stop words as mentioned previously are the words that have no significance in the actual sentiment of the document. The tokenised output of the input text defines the easy access of the words in the text.

2) Sentence Splitter:

It is always easier to analyse natural laguage sentence-wise. Sentiment classification is widely done sentence wise. The same methodology is used here. Hence this module has the most important function of splitting the document to sentences. The tokens that are the output of the lexical analysis phase are diven as input to this phase. This module then splits the tokenised document into seperate sentences. Each sentence is the sentence in the same document but each sentence in its tokenised form.

3) Part-of-Speech Tagger:

The POS Tagger explains the significance of the words in the document. The adverbs, adjectives and verbs are the most significant set of words that have any effect on the sentence. Thus this module helps in identifying the most affective word in the given document. From this block the adverbs, adjectives and verbs are then used so as to continue the further processing.

4) Word Sense Disambiguator:

This section allows to understand the different senses possible to a word. Senses include the different contexts, their definitions and location of the words in the sentence and how it affects the word is identified in this section. It is extracted from WordNet data[14]. Once the different senses are obtained then possible affective words are analysed for their affect on the text. Some may not have an affect but others may and hence is better useful if all the words defining the affect are obtained. A dictionary datastructure is used for this process.

5) Stemmer:

This section does the process of stemming the words to their root stems. An algorithm for stemming the words under consideration is the Porter Stemming Algorithm[13]. Here as mentioned earlier each of the word is stemmed backwards from the end to the beginning of each word. The truncation continues until the same stem of the word remains in the word. The porter stemmer algorithm does this process of stemming words and generating the stems of the words. The stemmer phase takes the input from the word sense disambiguator and stems the words that have been obtained from the word sense disambiguator.

6) Keyword Spotter:

This section works on developing monograms and bigrams. Thus this block identifies n-grams and then using them as features. This phase takes up three features where initially the word features are identified by confirming whether the word exists in the document. This is similar to the next feature to identify the unigrams in the word and hence check for their availability.

7) Classifier:

A Naive Bayes classifier is used for classification[15]. The initial phase implements the training with the movie review dataset and then training with the rest of the review data. Once the training has completed, the input text's feature is given to the classifier for the output. The classifier then based on the training outputs the class that the input text belongs to either positive or negative.

B. The Thesarus

Once the classification has completed a thesarus is introduced with the possible words and their respective sentiment. The thesarus is a cross-domain thesarus [11] and uses the words that may give the same sentiment meaning the affect or the sentiment to the different words in different domains. The synonymns of each word is analysed and then their polarities are assigned to the same polarity. Once a text comes in and if the words are not identified based on their domain, then the thesarus is checked for the similar word and then the polarity assigned to it is assigned to the word. After this the text then is completely processed and then the text identifies the sentiment corresponding to the text and is then given as output. Thus the advantage of this method is to allow the process of a cross domain sentiment classification. Thus there is no restriction over what kind of text is to be given as input.

The output is the class either positive or negative. The thesarus enables a better text processing [16][17] with the data being analysed irrespective of the domains. One representation is given below with the possible changes in EmoLib pipeline.

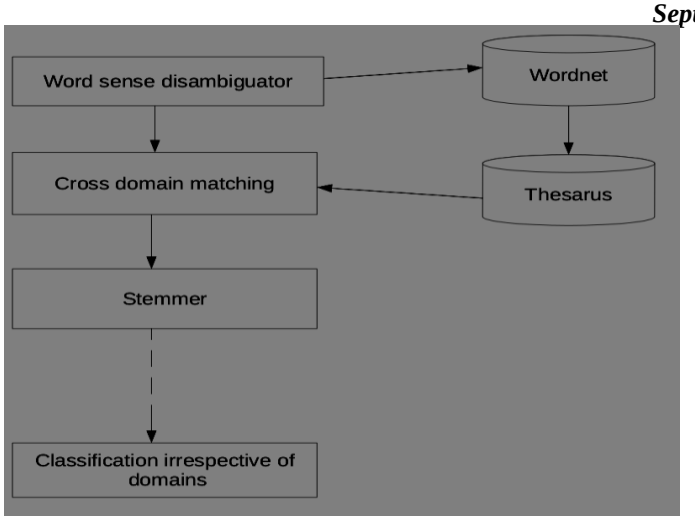

Fig 3: Cross-domain sentiment classification

The above figure explains the change introduced by the cross-domain thesarus. As the different senses are given and then the actual sense is identified, then it becomes important to search for the emotion of the data in its actual sense. The identification of the emotion 's actual sense is done by the thesarus that is introduced in the pipeline between the word sense phase and the stemmer phase. The words where the senses are not present or unknown, especially that do not belong to the domain in which the training has been done, are searched in the data that is stored in the thesarus. If such a match is found, then the emotion tagged to the word is given to the word from the thesarus. For example, in a word like "**heavy rusting**" the meaning might not be relevant in a movie review domain but is a negative emotion in a kitchen appliance dataset. And hence its meaning will be that of a very negative entity in a data about metals. Once the synonyms are checked in the wordnet data and if the data is not found, then the thesarus is checked for the matching emotion. If the data is found, then it is accessed from the wordnet database itself. Otherwise the thesarus checks for emotions of the given words and then the corresponding emotion is extracted. This is then reduced to the appropriate stem which generalises the possible meaning it has. And then the process of classification is proceeded.

IV. IMPLEMENTATION AND RESULTS

The implementation of the methodology that is defined in the previous section is done by means of the different packages that are available in Python 2.7. Each of the phases are either externally defined or predefined built-in functions from the required package is used.

From the initial phase i.e, right from tokenising, the document is tokenised into tokens by means of spaces, punctuations,etc. Then the sentence splitter is implemented to identify individual sentence seperately. Then the part-of-speech of the words are identified and the nouns, verbs, adjectives and adverbs are all extracted to the sense disambiguator which makes out the required sense. If such a sense is found from the wordnet itself, then the word and the appropriate sense is given, but if the sense is not found then the thesarus searches through the data and then through multiple domains if the possible meaning is retrieved the respective sense with the emotion is returned to the next stemmer block.

The property of porter stemmer algorithm is that it truncates words with the sequence like '-ing', '-ous', '-ity', etc. Thus for the words 'banker' and 'banking' the stemmed value is 'bank'. Thus the meaning is expected to be similar for words that share the same stem. Then the process of spotting the relevant unigrams as well as the relevant bigrams based on the sentiment is done.

For example,

Consider the words: **'excellent' , 'broad', 'survey'**

which are a set of unigrams, then the possible bigrams are: **'excellent+broad**' and **'broad+survey'** . These are expected to have positive sentiment. After the unigram as well as bigram extraction, they are given to the trained classifier for classification.

Before giving the data from the actual data, the classifier is trained using movie review data. The result is either positive or negative. If a text does not have one particular emotion then it is considered neutral and classified as positive. The dataset consisted of 2000 reviews, with the first 1500 of them are given as training data and the remaining 500 are the reviews for testing. Once the Naive Bayes classifier is trained, then the features from the text are given as input and then the classification is done. The result is the emotion in the text which is either positive or negative.

For example:

In giving the following sentence as input :

"Excellent and broad survey of the development of civilisation"

the output is obtained as:

"Positive"

It is important to note that the input is of a survey which is not be the actual domain trained. But it provides the correct sentiment irrespective of the domain which is enabled by the domain.

V. CONCLUSION

This work was about classifying the sentiment using the EmoLib pipeline and doing it with the help of all the possible features extracted in the pipeline. The different stages along with their detailed working was discussed. A survey on the different techniques used are analysed along with when and by whom. Before that a basic introduction on the process of sentiment analysis was also given. After explaining the pipeline, an explanation on the cross domain sentiment sensitive thesarus was given. One possible future step is the implementation of text to speech by generating expressive text to speech. This work has thus proposed a method for sentiment classification of text irrespective of domain.

REFERENCES

[1] F. Alas, X. Sevillano, J. C. Socor, and X. Gonzalvo, Towards high-quality next- generation text-to-speech synthesis: A multidomain approach by automatic domain classification, IEEE Trans. Audio, Speech, Lang. Process., vol. 16, no. 7, pp. 13401354, Sep. 2008.

[2] "A Dependency Tree based Approach for Sentence-level Sentiment Classification" Peifeng Li, Qiaoming Zhu, Wei Zhang, 2011 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing

[3] News video story sentiment classification and ranking, Chunxi Liu, Li Su, Qingming Huang, Shuqiang Jiang

[4] Sentiment Analysis of Social Issues, Mostafa Karamibekr, Ali A. Ghorbani. 2012 International Conference on Social Issues

[5] Mostafa Karamibekr, Ali A. Ghorbani, Verb Oriented Sentiment Classification, 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology

[6] Costin-Gabriel Chiru, Asmelash Teka Hadgu, Sentiment-Based Text Segmentation, 2013 2nd International Conference on Systems and Computer Science (ICSCS) Villeneuve d'Ascq, France, August 26-27, 2013

[7] Eric T. Nalisnick and Henry S. Baird , Extracting Sentiment Networks from Shakespeares Plays, 2013, 12th International Conference on Document Analysis and Recognition

[8] Ms.K.Mouthami, Ms.K.Nirmala Devi, Dr.V.Murali Bhaskaran, Sentiment Analysis and Classification Based On Textual Reviews,

[9] Rabia Batool , Asad Masood Khattak , Jahanzeb Maqbool and Sungyoung Lee, Precise Tweet Classification and Sentiment Analysis

[10] Alexandre Trilla and Francesc Alías, Member, IEEE ,Sentence-Based Sentiment Analysis for Expressive Text-to-Speech , IEEE transactions on audio, speech, and language processing, Vol. 21, No. 2, February 2013

[11] Danushka Bollegala, Member, IEEE, David Weir, and John Carroll, Cross-Domain Sentiment Classification Using a Sentiment Sensitive Thesaurus. IEEE transactions on knowledge and data engineering, Vol. 25, No.8,August 2013

[12] Alexandre Trilla , Francesc Al´as , Sentiment classification in English from sentence-level annotations of emotions regarding models of affect.

[13] M. F. Porter, "An algorithm for suffix stripping," Program, vol. 14, no. 3, pp. 130–137, 1980.

[14] C. Fellbaum, Ed., WordNet: An Electronic Lexical Database, 1st ed. MIT Press, 1998.

[15] F. Sebastiani and C. N. D. Ricerche, "Machine learning in automated text categorization," ACM Computing Surveys, vol. 34, pp. 1–47, 2002.

[16] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs Up? Sentiment Classification Using Machine Learning Techniques," Proc. ACL-02 Conf. Empirical Methods in Natural Language Processing (EMNLP '02), pp. 79-86, 2002.

[17] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," Foundations and Trends in Information Retrieval, vol. 2, nos. 1/2, pp. 1-135, 2008.

