Sentiment Analysis in A Cross-Media Analysis Framework

Yonas Woldemariam
Department of Computing
Science Umea University
Umea, Sweden
e-mail: yonasd@cs.umu.se

Abstract—**This paper introduces the implementation and integration of a sentiment analysis pipeline into the ongoing open source cross-media analysis framework. The pipeline includes the following components; chat room cleaner, NLP and sentiment analyzer. Before the integration, we also compare two broad categories of sentiment analysis methods, namely lexicon-based and machine learning approaches. We mainly focus on finding out which method is appropriate to detect sentiments from forum discussion posts. In order to conduct our experiments, we use the apache-hadoop framework with its lexicon-based sentiment prediction algorithm and Stanford coreNLP library with the Recursive Neural Tensor Network (RNTN) model. The lexicon- based uses sentiment dictionary containing words annotated with sentiment labels and other basic lexical features, and the later one is trained on Sentiment Treebank with 215,154 phrases, labeled using Amazon Turk. Our overall performance evaluation shows that RNTN outperforms the lexicon-based by 9.88% accuracy on variable length positive, negative, and neutral comments. How- ever, the lexicon-based shows better performance on classifying positive comments. We also found out that the F1-score values of the Lexicon-based is greater by 0.16 from the RNTN.**

Keywords-sentiment analysis; cross-media; machine learning algorithm; lexicon-based; neural network; (key words)

I. Introduction

A massive volume of both structured and unstructured mul- timedia data is being uploaded on the Internet due to rapidly growing ubiquitous web access over the world. However, analyzing those raw media resources to discover their hidden semantics is becoming a challenging task. As a result, it is difficult to retrieve the right type of media to satisfy multimedia content consumers. So, improving the searchability of multimedia contents on the web is one of the most appealing demands, especially for online audio/video content providers. Even if there are a lot of effective approaches for indexing textual contents, they cannot be applied to index media type such as audio and video, unless we transform them to some form of text, and add advanced metadata annotations using contextual information around the target media. This problem motivated for the genesis of the ongoing EU research project called Media in Context (MICO). MICO mainly aims at providing cross-media analysis framework, including orchestrated chain analysis components to extract semantics from the media in a cross-media context (eg. a web page containing text, image, audio, video, metadata and so on).

We are mainly concerned with the textual analysis aspect of MICO, including sentiment and discourse analysis, language identification, and named entity recognition. Sentiment analysis copes with the task of opinion mining from text. With the growth of user generated texts on the web, exploring the method to automatically extract and classify opinions from those texts would be enormously helpful to individuals, business and government intelligence and in decision-making. Some of the early research works in this area include [1], [2], in these works different methods have been used for detecting the polarity of product reviews and movie reviews respectively.

In general, sentiment analysis methods are classified into lexicon-based [3] and machine learning-based [4], [5]. Machine learning methods make use of learning algorithm and classifier models trained on a known dataset. The lexicon-based approach involves calculating sentiment polarity using dictionaries of words annotated with sentiment scores.

The general goal of this study is to assess the available sentiment analysis technologies and adapt to MICO. In order to achieve the goal, we compare these two broad categories of sentiment analysis methods regarding to their prediction accuracy and find out which method outperforms the other. We chose our test case to be Zooniverse (https://www.zooniverse.org,) forum discussion domain. Zooniverse is an online plat- form where volunteers contribute for scientific discovery for its several projects. One of its projects is Snapshots Serngeti (http://www.snapshotserengeti.org,) the purpose is to study animals in Tanzania Serengeti National Park, volunteers go to their website to analyze and classify animals into species, discuss about their classification and generally about the images, on the forum posts. Our focus is to run sentiment analysis on texts extracted from the forum to help them assess what the volunteers feel about the quality of the images and generally about their services. Unlike the comments found in social media such as Twitter, the nature of the texts we get from Serengeti Snapshot is highly characterized by descriptions about observed images rather than explicit opinions. So studying sentiment analysis with such kind of text creates its own new research challenges due to its unique features and worth to observe how the sentiment analysis methods behave on these dataset. In order to conduct our experiments, lexicon- based sentiment prediction algorithm and Recursive Neural Tensor Network (RNTN) [5] model are chosen. The former is implemented on single node version of Hadoop platform, the dictionary contains sentiment words annotated with sentiment scores, and the later one is trained on Sentiment Treebank containing 215,154 phrases, labeled using Amazon Turk. We found that RNTN outperforms lexicon- based by 9.88% accu- racy. In order to give the whole picture of the comparison, we have calculated other measures such as precision, recall and F1-score.

The remainder of this paper is organized as follows: Section II presents related literature review. Section III gives an overview of sentiment analysis component within the MICO architecture. Section IV and V discuss two selected methods to be compared. Section VI discusses evaluation and results. Finally, the last section briefly indicates the directions for future research.

II. Related Litrature Review

Even though there are several research works [2], [4], [6] which compare methods for sentiment analysis, most of them focus on comparing different machine learning methods. There are a few comparative studies [7], [8] on lexicon-based versus machine learning approaches. In [7], twitter testing dataset with a total of 1,000 tweets used to undertake comparison be- tween lexicon-based and machine learning approaches. After data pre-processing steps such as data cleaning, stemming, part of speech (POS) tagging, and tokenization, they run tests using Support Vector Machine (SVM), Maximum Entropy (ME), Multinomial Naive Bayes (MNB), and k-Nearest Neighbor (k- NN) machine learning techniques. Sentiwordnet has been used for lexicon-based sentiment classification. The result shows that machine learning methods produce better accuracy rate than lexical based approach. As they stated, the significant influence from lexical database has been set as reference in determining positive and negative opinion that means the lexical based method highly depends on the occurrence of the sentiment words present on the database. Another comparison study is conducted in [8], using 1,675 sentences from political news domain, the dataset is divided into, 1,137 positive and 538 negative sentences. After data cleaning, the authors ap- plied tokenization, stop word removal, lemmatization and POS tagging using natural language tool kit (NLTK) and Stanford POS tagger The lexical based was implemented using Senti- WordNet and Naive Bayes (NB) and Support Vector Machine (SVM) machine learning algorithms were implemented using WEKA. Among the methods the best F-measure shown by SVM. Our study aims at presenting a general comparison of two sentiment analysis methods (lexicon-based and supervised structured machine learning technique). The experiment is carried out by implementing sandboxed version of apache- hadoop and Stanford coreNLP library on sample Zooniverse dataset. As hadoop and Stanford coreNLP are being used in the cross-media software project, which motivated us to focus on the two methods.

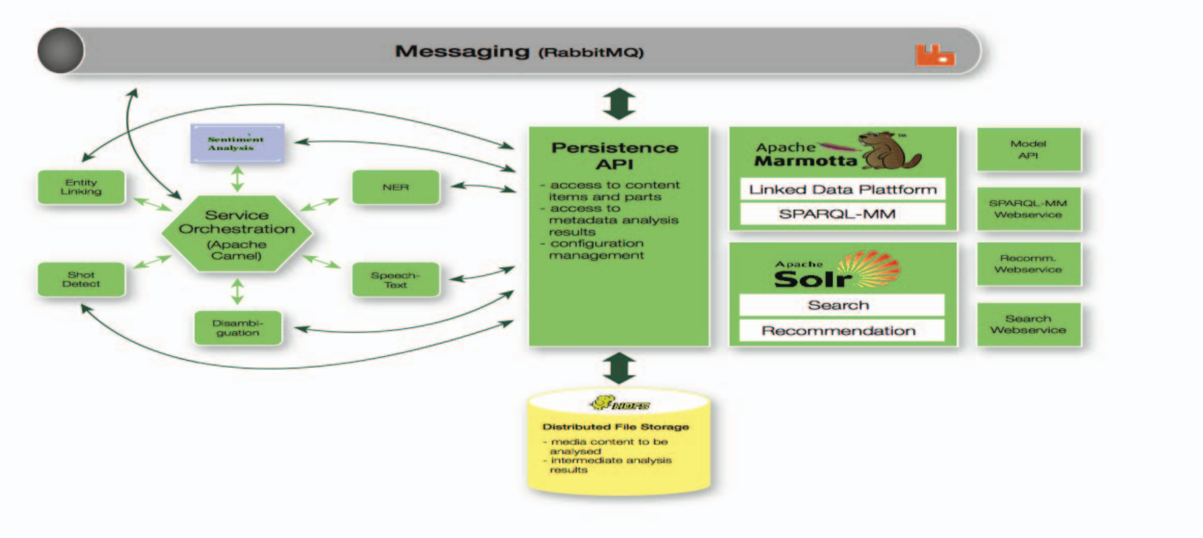


Figure 1. MICO General Architecture, adopted from [9].

III. An Overview of the Sentiment Analisis Component in A Mico Framework Architecture

The MICO framework uses a distributed service-oriented architecture (illustrated in Figure 1.), analysis components run independently and share communication and persistence infrastructure. Basically, the main services provided by the framework include, media analysis, search and recommendation. Once analysis components get registered with the framework and up running, the user can load a content item with its context. The service orchestration component notifies the respective analysis components about the input using its execution plan build as a result of service registration. The intermediate analysis results are stored with the metadata of the input in the persistence component, to enrich the existing basic metadata. Up on finishing processing the input content item, the final result is made available for further processing [9].

The input for a sentiment analysis component is a set of documents (or just a text from speech to text component within the framework), such as a HTML documents, news or movie reviews, comments from blog posts, or a text document in any format. The input is cleaned and pre-processed by chat room cleaner module, which removes non-standard characters and repeated spaces, and produces a plain text. Then the sentiment analysis uses its natural language processing sub component for tokenization, stemming, split into sentences, and so forth. Then the output texts are sent to the sentiment computation module which annotates them using the dictionary or machine-learning approaches, which includes annotations with sentiment polarity (positive/negative) of each word. The output of the sentiment analysis component is the annotations which can be attached to whole document.

IV. Lexicon-based Sentiment Classification Using Hadoop

Apache Hadoop (https://hadoop.apache.org), serves as big data solution for the processing of unstructured and complex sets of data. It uses the divide and rule methodology for processing through its parallel programming models. It mainly provides the Hadoop Distributed File System (HDFS) store the processed data. Apart from HDFS, Hadoop has several components and services including lexicon-based sentiment analysis. The main advantages we gain from this technology are big data analysis support and sentiment analysis service without having to prepare our own dictionary.

A. Data Pre-processing

Before we run lexicon-based algorithm for sentiment com- putation, we carried out the following pre-processing tasks:

1) Load the Snapshot Serengeti posts in CSV format into the HDFS.

2) Convert the raw posts into a tabular format.

3) Transform the data into a format that can be used for analysis.

B. Lexicon-based Algorithm

These are the major steps in the Algorithm 1.

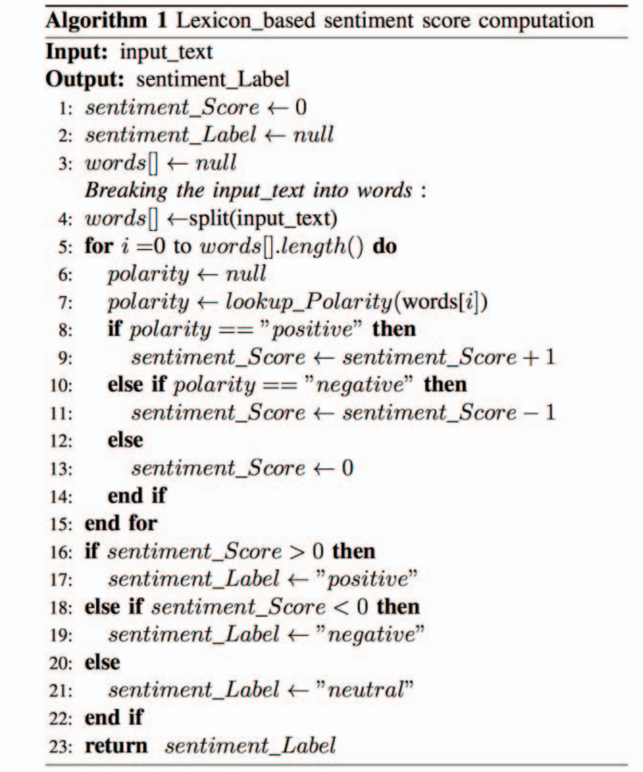
1) Tokenize the sentences into individual words.

2) Assign the polarity (positive, negative or neutral) for each word by using the sentiment dictionary.

3) Calculate the sum polarity value of all words within a sentence(s)

4) Compare the result with 0 and if result is greater than 0, then the sentiment is ‘positive ’or if result is equal to 0, then the sentiment is ‘negative ’, otherwise it is ‘neutral ’

5) Assign the sentiment value (2 for positive, 1 for neutral and 0 for negative) for the whole sentence



```
Algorithm 1 Lexicon_based sentiment score computation
Input: input_text
Output: sentiment_Label
$sentiment\_Score \leftarrow 0$
$sentiment\_Label \leftarrow null$
$words[] \leftarrow null$
$Breaking\ the\ input\_text\ into\ words$ :
$words[] \leftarrow$split(input_text)
for $i$ =0 to $words[].length()$ do
    $polarity \leftarrow null$
    $polarity \leftarrow lookup\_Polarity$(words[$i$])
    if $polarity ==$ "$positive$" then
        $sentiment\_Score \leftarrow sentiment\_Score + 1$
    else if $polarity ==$ "$negative$" then
        $sentiment\_Score \leftarrow sentiment\_Score - 1$
    else
        $sentiment\_Score \leftarrow 0$
    end if
end for
if $sentiment\_Score > 0$ then
    $sentiment\_Label \leftarrow$ "$positive$"
else if $sentiment\_Score < 0$ then
    $sentiment\_Label \leftarrow$ "$negative$"
else
    $sentiment\_Label \leftarrow$ "$neutral$"
end if
return $sentiment\_Label$
```

V. Stanford Sentiment Treebank

In [5], Stanford Sentiment Treebank and Recursive Neural Tensor Network (RNTN) are introduced. The Treebank contains fully labeled parse trees constructed from the corpus of movie reviews that allows for a complete analysis of the compositional effects of sentiment in language. The main reason we use RNTN as machine-learning technique is, it has been already trained so we do not need to have labeled dataset for training purpose.

For the case of RNTN, we use already trained sentiment model, so we only need to extract texts from the Snapshot Serngeti database dump without being too much engaged with the text preprocessing tasks. Here is the description of Algorithm 2:

1) Tokenize sentences into individual words which are represented as a numeric vector

2) Lemmatize each word into their basic forms

3) Tag words with part of speech tagger (POS)

4) Parse sentences into their constituent subphrases and build a syntactic tree

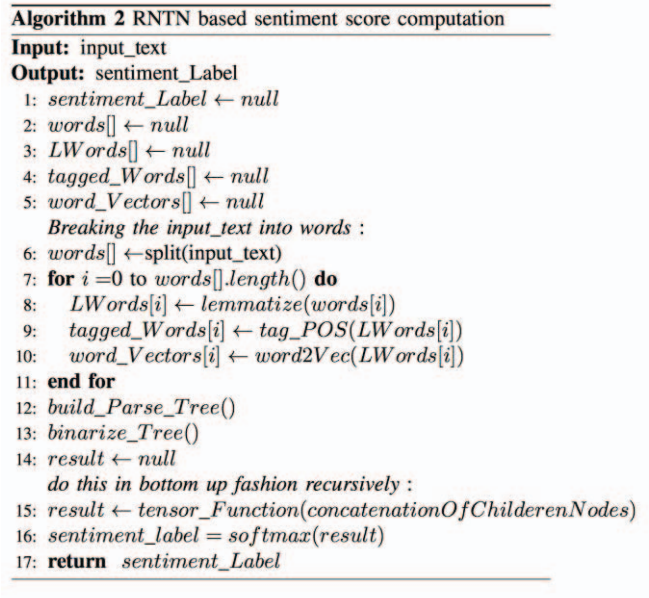
5) Binarize the tree, so that any parent node will have a maximum of 2 child nodes

6) Classify the sentences sentiment in a bottom up fashion using tensor-based composition function. The compo- sitionality function concatenates the vector of the two child nodes for each parent node, transforms the

vector resulted from the concatenation and analyse similarity.

7) The resulting vector is given to the softmax () classifier which computes its label probabilities, then the maxi- mum probability value will be returned as the sentiment label of the tree (sentence).

A. RNTN Algorithm



```
Algorithm 2 RNTN based sentiment score computation
Input: input_text
Output: sentiment_Label
$sentiment\_Label \leftarrow null$
$words[] \leftarrow null$
$LWords[] \leftarrow null$
$tagged\_Words[] \leftarrow null$
$word\_Vectors[] \leftarrow null$
Breaking the input_text into words :
$words[] \leftarrow$split(input_text)
for $i = 0$ to $words[].length()$ do
    $LWords[i] \leftarrow lemmatize(words[i])$
    $tagged\_Words[i] \leftarrow tag\_POS(LWords[i])$
    $word\_Vectors[i] \leftarrow word2Vec(LWords[i])$
end for
$build\_Parse\_Tree()$
$binarize\_Tree()$
$result \leftarrow null$
do this in bottom up fashion recursively :
$result \leftarrow tensor\_Function(concatenationOfChilderenNodes)$
$sentiment\_label = softmax(result)$
return $sentiment\_Label$
```

VI. Experimental Evaluation and Discussion

In order to evaluate the performance of the two methods (Lexicon-based and RNTN), we randomly chose 600 sample tweets from Zooniverse, particulary from Serengeti Snapshot forum posts. The dataset has been made to contain 200 positive, 200 negative and 200 neutral tweets from each class, which are annotated by human judge. We apply commonly used performance metrics [10] in sentiment analysis. These are Accuracy (A), Precision (P), Recall (R) and F1-score. Precision measures the exactness of a classifier. A higher precision means less false positives (FP) (explained with equa- tion (1)), while a lower precision means more false positives. Recall measures the completeness, or sensitivity, of a classifier. Higher recall means less false negatives (FN) (explained with equation (2)), while lower recall means more false negatives. F1-score is harmonic mean of precision and recall, 1 its an ideal value, where as 0 is its minimum value.

$$A = AI/T \tag{1}$$

$$P = TP/(TP + FP) \tag{2}$$

$$R = TP/(TP + FN) \tag{3}$$

$$F1 - score = 2(PR)/(P + R) \tag{4}$$

Where AI is the number of accurately predicted instances, T is the total number of instances,TP is the number of accurately predicted positive instances, FP is the number of incorrectly predicted as positive instances and FN is the number of posi- tive instances, but incorrectly predicted as negative instances.

TABLE I. Evaluation Results of the Lexicon-based and RNTN

Metrics	Lexicon-based	RNTN
Accuracy	38.45	48.34
Precision	0.63	0.82
Recall	0.96	0.46
F-score	0.74	0.59

As experimental evaluation shown in table I, the RNTN method outperforms lexicon-based by 9.88%, it is just the overall accuracy. However, the lexicon-based shows better performance on positive comments. The Lexicon-based also

Scores nearly a perfect R, that means every positive instance (which does not include reversed negative instance “not bad”) is correctly classified. Even if a wide gap has been shown by the two methods in terms of P and R, they have quite closer F1-score value, which makes sense as R does not show a measure of false negative.

We also observed that stronger sentiment often builds up in longer phrases and the majority of the shorter phrases are neutral, which supports with the claim, demonstrated in [5]. It has been hard to classify short comments, for example some comments have just only hash tags with a single word. Mostly, these comments tend to be classified as neutral. Some of the comments are really hard to be classified even by human due to their ambiguity. We have to be careful what aspects and context to consider, for example the comment might explain the scene on the image very well, that means the volunteer has got reasonably clear image to discuss, so from the quality point of view, we classify the comment as positive, on the contrary, the comment does not bear any explicit opinion thus, which leads us to classify it to be neutral. That is one of the potential challenges of this study.

Another interesting fact is, unlike to lexicon-based algorithm, RNTN has a potential to capture negation and learn the sentiment of phrases following the contrastive conjunction “but”. In the case of lexicon-based, the major reason for the prediction errors is the algorithm fails to understand the context of the words including negation. In general, the performance lexicon-based algorithm could be improved by capturing the context of the words and stemming the input words into their basic form. In the case of RNTN, the main source of the prediction errors is the mismatching of domain knowledge between training dataset and test dataset. The training dataset is collected from movie reviews where as the test dataset is obtained from citizen-science domain; as a result the algorithm is challenged to recognise some unseen positive/negative phrases specific to the domain. Therefore, the straightforward approach to improve the prediction accuracy is to further train the RNTN model on Snapshot Serngeti posts.

VII. Future Work

For this study, we just focused on the comparison of two sample methods from each broad category of sentiment analysis approaches with limited test dataset. In the future, we are planning to experiment with other kind of methods such as, Naive Bayes, and Support Vector Machines. We are also interested to go beyond positive/negative polarity detection, and extend our work to extract other emotional knowledge from text such as the confidence and competence of the Snapshot Sernget users while they discuss in the forms.

ACKNOWLEDGMENTS

We acknowledge the financial support from the EU FP7 MICO project. We also thank the Zooniverse people for providing the dataset used in this study. Finally, we thank Adil (student of Umea□ University), for labeling the test dataset.

REFERENCES

[1] B.Pang and L.Lee, “Opinion mining and sentiment analysis,” vol. 3, no. 1-2, 2008.

[2] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up sentiment classification using machine learning techniquest,” in In ACL Conference on Empirical Methods in Natural Language Processing, 2010, pp. 354-368.

[3] C. Musto, G. Semeraro, and M. Polignano, “A comparison of lexiconbased approaches for sentiment analysis of microblog posts,” on 8th International Workshop on Information Filtering and Retrieval, 2014.

[4] G.Vinodhini and R.Chandrasekaran, “Sentiment analysis and opinion mining: A survey,” International Journal of Advanced Research in Computer Science and Software Engineering., vol. 2, no. 2277 128X, 2012.

[5] R.Socher, A.Perelygin, J. Wu, J.Chuang, C.Manning, A. Ng, and C.Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in Empirical Methods in Natural Language Processing, 2013.

[6] P. Gonalves, M. Arajo, F. Benevenuto, and M. Cha, “Comparing and combining sentiment analysis methods,” in In Proceedings of COSN, 2013, pp. 27-38.

[7] W. Maharani, “Microblogging sentiment analysis with lexical based and machine learning approaches,” in Information and Communication Technology (ICoICT), 2013 International Conference of. IEEE, 2010, pp. 439-443.

[8] S.Padmaja, S.Sameen, and S.Bandu, “Evaluating sentiment analysis methods and identifying scope of negation in newspaper articles,” vol.3 , No.11, no. 11, 2014.

[9] S. Schaffert and S.Fernandez, “D6.1.1 mico system architecture and development guide.” MICO, 2014.

[10] Olson, D. L, and Delen, Advanced data mining techniques. Dursun:Springer, 2008.