A Sentiment Analysis Approach based on Arabic Social Media Platforms

La-sheng YU and Sadeq AL BAADANI*

School of Information Science and Engineering, Central South University, Changsha 410083, Hunan, China
*Corresponding author

Keywords: Sentiment analysis, Opinion mining, Text processing, Document polarity, Subspace ensemble, Lexicon classifiers.

Abstract. Apart from the major outstanding research issues facing Arabic social media sentiment analysis which includes handling of vernacular Arabic, slang vocabulary and shorthand writings. There is also a lack of comprehensive framework for Arabic social media sentiment analysis as existing works often focus on particular platforms (like twitter and Facebook). As such, models developed on one platform often perform poorly on other platforms due to lack of a representative feature space. To this regard we adopted a comprehensive approach utilizing a broad array of Arabic social media platforms to establish more generalized sentiment models using random subspace ensembles of MLP base learners. More importantly, we introduced a new sentiment classification scale and we classified sentiments as Highly Positive (HP), Fairly Positive (FP), No Sentiment (NS), Fairly Negative (FN) and Highly Negative (HN). The approach has been tested in a series of experiments and the results demonstrate significant improvements in terms of both classification accuracy and generalizing ability.

Introduction

Sentiment analysis is an increasingly attractive area of research giving its numerous application areas and prospects. In a broader sense, sentiment analysis also known as opinion mining is refers to the act of analyzing people’s opinions, feelings, attitudes, reactions and emotions towards issues and things such as products, services, policies and so on[1]. In technical terms, sentiment analysis is the act of determining the polarity of (that is positivity, negativity or neutrality) of a given text [2] [3]. This given text could be text obtained from social media platforms like Twitter, Facebook, YouTube, Amazon, Bing, Yahoo news and so on, which allow the creation and exchange of user-generated contents.

In this study we proposed comprehensive framework that uses dictionary look-ups, polarity tables and ensembles of MLP base learners for Arabic social media sentiment modeling and classification. The polarity tables are of three categories namely: Modern Standard Arabic (MSA), slang vocabulary (dialect and shorthand) and emotion icons (emoji’s). More importantly, we introduced a new sentiment classification scale and we classified sentiments as Highly Positive (HP), Fairly Positive (FP), No Sentiment (NS), Fairly Negative (FN) and Highly Negative (HN), instead of classifying sentiments as merely positive and negative like existing works.

The main contribution of this paper include: (1) evaluating and comparing existing machine leaning methods for Arabic language sentiment analysis. (2) Establishing a more representative feature space for Arabic social media sentiment analysis (3) Introduction of a new sentiment classification scale for sentiment analysis (4) Designing an integrated sentiment analysis framework using lexicon based techniques with machine learning methods. The rest of the paper is organized as follows: section 2 contains our proposed method, section 3 contains experiments and analysis and section4 concludes the paper with summary our contributions, findings and prospective future research direction.
Proposed Method

Our approach uses ensembles of weak MLP base learners and polarity tables consisting MSA text, slang vocabulary and emotion icons. The proposed approach is based on the following general step: Data Pre-processing, Feature Extraction, Sentiment Modeling and Classification. Hence in the proceeding subsections we shall be looking at each of these steps into detail.

Data Preprocessing

The data preprocessing step is divided into two phases namely segmentation phase and refining phase. In the segmentation phase the data (text) is first separated into Arabic and non-Arabic text categories. After which the Arabic text is further separated into Modern Standard Arabic (MSA) and slang vocabulary (which constitutes dialectical Arabic and shorthand writings). To do this we designed a segmentation algorithm using dictionary look-up method with Bayes rule. The Bayes rule is used to perform automatic spell checking and auto-correction. The algorithm was implemented using python programming. Algorithm 1 describes our segmentation phase.

**Algorithm 1:** Segmentation phase

**Input:** Arabic social media comments

**Output:** List $\alpha$. [MSA text] list $\beta$. [Slang vocabulary] list $\lambda$. [Non-Arabic text]

1. Tokenize input text into a list
2. Look up each token in a list of Arabic characters
3. If found store in list $\alpha$ else store in list $\lambda$
4. Given list $\alpha$ look up each token in MSA dictionary
5. If found maintain in list $\alpha$ else send to list $\beta$
6. Given list $\beta$ as the new input, find the correction $c$ out of all possible candidate corrections, that maximizes the probability that $c$ is the intended correction, given the original word as follows:

$$P\left(\frac{c}{w}\right) = \arg\max_{c \in C} P\left(\frac{w}{c}\right)$$

7. Applying Bayes' Theorem we have:

$$P\left(\frac{c}{w}\right) = \arg\max_{c \in C} \frac{P\left(\frac{w}{c}\right)P\left(c\right)}{P\left(w\right)}$$

8. Given that $P(w)$ is the same for every possible candidate $c$ and all candidates $c$ we have:

$$P\left(\frac{c}{w}\right) = \arg\max_{c \in C} P\left(\frac{w}{c}\right)P\left(c\right)$$

9. Return $c$ to list $\alpha$ else keep the original word $w$ in list $\beta$

where $P\left(\frac{c}{w}\right)$ is probability that the user will type $w$ when he actually meant $c$. $P\left(c\right)$ is the probability that the word we are suggesting actually appears in Arabic text and $P\left(w\right)$ is the probability of the original word $w$ or the misspelled word. To find this value of $P\left(c\right)$ we made use of a large Arabic corpus that contains many different words and we used frequency of each word to compute the probability its occurrence in Arabic language. To compute $P\left(\frac{w}{c}\right)$ we made use of the edit distance assumption which suggests that people are more likely to interchange alphabets with closer proximity on the keyboard or omit an alphabet than spell the entire word wrongly. So we created our set of candidate corrections using valid dictionary words at 1 or 2 edit distance from the original word $w$.

Given the segmented data ($\alpha$. MSA text, $\beta$. slang text and $\lambda$. non-Arabic text) as inputs to the refining phase, various preprocessing treatments adopted. For list $\alpha$ and $\beta$ text standardization is...
performed by converting all Arabic texts to their standard forms. This process involves: removing elongation effects, removing the diacritical markings, removing the letter ‘Hamza’ (ُ), replacing ٍ with َ, replacing ِ with ِ. After standardization, annotation is performed on lists α and β by assigning polarity values to tokens using look-ups from respective polarity tables. Then all un-annotated texts from lists α and β are looked-up in a supplementary dictionary to identify function words (not, no, neither, never and so on) with negation effects while all unidentified text in β are discarded as noise. Upon detecting a function word, polarity normalization is performed by reversing the polarity value of the closest tagged token to the left (Note: Arabic is written from right to left). Finally all tokens in list λ are looked-up in an emoji polarity table. If the token is found a polarity tag is assigned else the token is discarded as noise.

The original text is then recomposed by combining the segmented lists to obtain a refined version while discarding all stop words. Algorithm 2 describes the refining phase.

Algorithm 2: Refining phase

Inputs: List α. [MSA text], list β. [slang text] and list λ. [non-Arabic text]

Outputs: Refined version of text

1. Initialize
2. For every token in lists α and β transform to standard form.
3. For every token in list α search MSA polarity table and assign a polarity tag if token is found, else search supplementary dictionary and reversed the polarity value of the closest tagged token to the left if token is found.
4. For every token in list β search Slang polarity table and assign polarity tag if token is found, else search supplementary dictionary, discard as noise if token is not found, else perform normalization by reversing the polarity value of the closest proceeding tagged token to the left.
5. For every token in list λ search emoji table and assign a polarity tag if token is found, else discard as noise.
6. Combine lists α, β and λ to recompose a fine version of the original text while discarding all stop words through look-ups.
7. Return refined version.

Feature Extraction

In this study we deployed statistical Bag of Words representation and word co-occurrence [4] [5]. To begin with statistical Bag of Words representation, we treated the recomposed text as our feature vocabulary and we computed a utility measure $\phi$ for each term of the vocabulary. Specifically we used three different utility measures namely: mutual information, chi square and term weight. We created feature histograms using K terms with highest values of $\phi(t,c)$ and then we obtained our feature vector from the histograms [5].

Sentiment Modeling

For our sentiment modeling and classification we decided to treat it as a multiclass problem unlike the earlier works where sentiment analysis is treated as a binary classification problem. As such instead of classifying sentiments as merely positive or negative, we took a step further and introduced the following five sentiment classes: Highly Positive (HP), Fairly Positive (FP), No Sentiment (NS), Fairly Negative (FN) and Highly Negative (HN). This is one of the major contributions of our work. In this study we used Random Subspace (RS) Ensemble of Neural Network Classifiers and we trained each base learner with a subset of randomly selected features from our original feature space obtained in section 2.2.

To train the classifier, we collected a training corpus from various Arabic social media platforms and we annotated the corpus at sentence level according to our defined sentiment classes: Highly Positive (HP), Fairly Positive (FP), No Sentiment (NS), Fairly Negative (FN) and Highly Negative (HN). Then using the annotated training corpus we trained an ensembles of MLP based learners in a
random subspace to model each of our defined sentiment classes. Thus given a query text we can perform classification using our MLP ensemble sentiment models.

Data Collection and Preprocessing

Training Data
For our dataset we manually collected comments and reviews from seven different popular Arabic social media platforms namely: Google Plus, AreebaAreeba, Facebook, YouTube, Twitter, Yahoo news and WeChat Moments. Thus we created Arabic corpora of seven distinct categories as mentioned above. Each corpus is organized in 15 different text documents giving us a total of 105 text documents. Each document in the corpora has 200 comments given us 15,750 comment from diverse social media platform. The entire corpora consist of 396,585 Arabic words. Annotation was done at paragraph (comment) level and only the first 10 documents of each corpus were annotated resulting in 14000 annotated comments. Five different groups (with each group having 3members) participated in the annotation and majority voting was used to resolve disagreements. The annotation was done based on our newly introduced sentiment classification scale and 2317 comments were labeled as Highly Positive (HP), 3852 as Fairly Positive (FP), 1098 as No Sentiment (NS), 3571 as Fairly Negative (FN) and 3162 as Highly Negative (HN).

Dictionaries and Tables
For our work we used Modern Standard Arabic dictionary and three polarity tables: Modern Standard Arabic (MSA), slang vocabulary and emotion icons (emoji’s).We also created a supplementary dictionary (for function words with negation effect).The MSA dictionary contains 1,159 words/phrases and the supplementary dictionary contains 27 words.

The MSA polarity table is composed of 427 positive Arabic words/phrases and 350 negative Arabic words/phrases. The slang polarity table contains 528 positive Arabic words/phrases and 837 negative Arabic words/phrases. The emoji polarity table contains 150 positive emoji’s and 211 negative emoji’s.

Experiments and Results

Parameters Selection
Our MLP based method has 3 layered network: input, hidden and output. The number of inputs is the same as the number of features. For the hidden layer we tested different number of neurons (10, 20, 40, 80 and 160 nodes) as shown in figure 1. For the output layer we used a single linear unit to represent the class label. The network was trained using 500 epochs. During the training session Levenberg-Marquardt algorithm [9] was used to update weight and bias values.

![Figure 1. Average accuracy of MLP with different activation functions in the input & output layers.](image)

From the figure it can be observed that using logistic function as the activation function in the hidden nodes and hyperbolic tangent function in the output nodes (logistic & Hyper-tan) gives the best performances. So in our work we used Logistic & Hyper-tan. Besides that, the best
performance for the MLP with Logistic & Hyper-tan is realized with 80 nodes in the hidden layer (with 89.75% accuracy).

**Method Evaluation**

For evaluation of our RS Ensemble of MLP classifier, we used 10-fold leave-one-out cross validation approach. We created our sentiment models utilizing annotated comments (paragraphs) of 9 folds and the rest for validation. For comparison of our method with other well-known methods, we implemented Support Vector Machine (SVM), K Nearest Neighbor (KNN), Random subspace ensemble with SVM base learners (RS-SVM) and Random subspace ensemble with k Nearest Neighbor base learner (RS-KNN). We generated the confusion matrix for each method using True Positive (TP), FN False Negative (FN), False Positive (FP) and True Negative (TN) scores. We then computed the accuracy of each method [5] [6] [7]. Fig.2 and Fig.3 are confusion matrices showing the how our proposed RS Ensemble of MLP classifier performed against other classifiers across our newly introduced sentiment classification scale.

**Implementation Results**

Table 1 shows examples of preprocessed and classified comments based on our newly introduced sentiment classification scale.

<table>
<thead>
<tr>
<th>Original Comment</th>
<th>Preprocessed Text</th>
<th>Preprocessed Text English Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>لا يناسبها جدا بالنسبة ل حلقة البروت</td>
<td>Not bad enough for today’s episode</td>
<td></td>
</tr>
<tr>
<td>The Hemi Hamk series ended, now what they are talking about is not important</td>
<td></td>
<td></td>
</tr>
<tr>
<td>مسلسل حلووووووور مره انتقالي للكلك لا يك</td>
<td>Mسلسل حلوووووور مره انتقالي للكلك لا يك</td>
<td></td>
</tr>
<tr>
<td>This part failed by all standards, neither photography nor the content ... Frankly Hami Hamk lost creativity in the absence of the main characters of the series</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The characters and events in this TV program are just imaginary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Preprocessed and classified text using RS Ensemble with MLP.
Conclusion

The paper presented a comprehensive approach to Arabic social media sentiment analysis by combining lexicon based ideas with machine learning techniques. The approach generally has three phases namely data pre-preprocessing, feature extraction and sentiment modeling and classification. For data preprocessing we adopted a divide and conquer approach and we segmented the input text into Modern standard Arabic, slang vocabulary and non-Arabic text. Then we cleaned the segmented text separately after which we recomposed a refined version of the input text. For feature extraction we computed the mutual information, chi square and term weight for each term and we represented them using statistical Bag of Words and word co-occurrence. And finally for sentiment modeling and classification we used adopted an Ensemble approach and we created a random subspace ensemble with MLP base learners.

Unlike existing works, in this study we introduced a new classification scale for sentiment analysis: Highly Positive (HP), Fairly Positive (FP), No Sentiment (NS), Fairly Negative (FN) and Highly Negative (HN). This is a more realistic way of sentiments analysis as it provides the freedom to classify sentiments over a broader scope of polarity.

To evaluate the performance of our RS-MLP method we made comparison with standard SVM and KNN classifiers as well as RS-SVM and RS-KNN. The experimental results showed that RS-based methods are not only superior to methods using standard classifiers but RS-MLP has a significant edge over other RS-based method.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (Grant No. Z201610110620003).

References


