2014 5th International Conference on Information and Communication Systems (ICICS)

Sentiment Analysis in Arabic Tweets

R. M. Duwairi
Department of Computer Information Systems
Jordan University of Science and Technology
Irbid 22110, Jordan
rehab@just.edu.jo

Raed Marji, Narmeen Sha'ban, Sally Rushaidat
Department of Computer Engineering
Jordan University of Science and Technology
Irbid 22110, Jordan

***Abstract*— Social media platforms such as blogs, social networking sites, content communities and virtual worlds are tremendously becoming one of the most powerful sources for news, markets, industries, and much more. They are a wide platform full of thoughts, emotions, reviews and feedback, which can be used in many aspects. Despite these great avails, and with the increasingly enormous number of Arabic users on the internet, little research has tied these two together in a high and accurate professional manner [1].**

This paper deals with Arabic Sentiment Analysis. We developed a framework that makes it possible to analyze Twitter comments or "Tweets" as having positive, negative or neutral sentiments. This can be applied in a wide range of applications ranging from politics to marketing. This framework has many novel aspects such as handling Arabic dialects, Arabizi and emoticons. Also, crowdsourcing was utilized to collect a large dataset of tweets.

Keywords*— *Arabic Sentiment Analysis, Opinion Mining, Data Analytics, Supervised Learning, Crowdsourcing.

I. INTRODUCTION

With the rapid growth of web applications and social media, there became reviews, comments, ratings and feedback generated by users. These opinions can be about virtually anything, including products, politics, news, people, services and events. All of which need to be processed and analyzed to obtain a good estimation of what the user thinks and feels. Before the availability of automatic sentiment analysis tools, the process of obtaining customers' reviews was extremely cumbersome and time consuming task. This probably explains the great interest of this field of research. [2]

Sentiment can be defined as the feeling or emotion behind a mention of your brand, campaign or service in the social universe. It is a way to measure comments and references to gain an understanding of the overarching feeling surrounding your brand, campaign or service.

When paired with more traditional measurement tools, sentiment analysis can help create a richer portrait and inform your future social media and content strategies. Sentiment analysis, on the other hand, can be seen as a set of algorithms implemented in computer software that detects and exploits opinions and emotions in online social media resources. It is a interdisciplinary field that borrows techniques from natural language processing, text analytics and computational linguistics to extract subjective information. [3]

Many sentiment analysis tools were developed for English, but we are trying to break a new ground in this arena, and come up with a high accuracy Arabic based sentiment analysis tool which is not affected by the use of dialects; a tool that gives Arab users the power to analyze the social media, giving them the ability to know the general sentiment about hot topics being discussed. The Arabic language has many dialects that should to be considered, where in each dialect meanings of words can be totally different. Arabic is a morphologically rich language and this can raise problems for any automatic text analysis tool [4, 5].

Even though people can use several social media platforms to interact and share comments, we chose to target Twitter in this research. Twitter is an informal channel where bloggers can use informal language or slang in their tweets. Also, there is a maximum length of 140 characters per tweet which makes it changeling to detect sentiment from such short and informal comment [6, 7].

The novel contributions of this research are: Arabic sentiment analysis tool, a lexicon that maps Jordanian Dialect to Modern Standard Arabic (MSA), a lexicon that maps Arabizi [8] words to MSA (Arabizi are Arabic words written using Roman Alphabet) and a lexicon of emoticons [9]. The last three lexicons are used to enhance the accuracy of Arabic sentiment analysis.

The rest of the paper is organized as follows: Section 1 has presented an introduction to the current research. Section 2, provides background information and related work. Section 3, by comparison, introduces the framework that we have developed for sentiment analysis. Section 4 highlights the experimentation setup and analyzes the results. Lastly, section 5 draws the conclusions of this work.

978-1-4799-3023-4/14/$31.00 ©2014 IEEE

II. Background and Related Work

A. *Applications of Sentiment analysis*

Sentiment analysis has applications in vital sectors. The following points highlight some of these applications:

- Marketing: Since social media has become a unique platform of customer interactions, the use of sentiment analysis can easily take marketing to a whole new level. Companies have figured that emotions of social media are shaping their brand's image. Therefore, sentiment analysis tools give marketers a way to measure their effectiveness, and help consumers who are trying to research a product or a service.
- Politics: Many valuable uses can be obtained for political organizations by fully understanding social media sentiment. Social media feedback has been used to inform political leaders of potential threats, problems or issues with their organizations. In addition, an essential role for sentiment analysis appeared earlier in predicting elections, and acquiring citizens' responses on important issues such as increasing prices and changing the constitution.
- Healthcare: Medical web blogs are all over the internet these days. These weblogs contain only medicine and health-care issues such as diseases, medical treatments and medications. Due to the health-related experiences and medical histories these webpages provide for practitioners and patients, sentiment analysis tools had to be developed for the use in medical fields.
- Finance: Sentiment Analysis can also be used in the financial world. Investors can easily follow their favorite companies and monitor their sentiment data in real time. With sentiment analysis, business investors can acquire business news easier and aggregate this information to make better financial decisions.

B. *Related Work*

Local grammar was used by the authors of [10] to analyze sentiment in text written in Arabic, English or Urdu. The target reviews were financial news. The accuracy provided by their systems was not high. Rushdi-Saleh et al [11] built a small Arabic opinion corpus that consists of 500 positive and 500 negative reviews. A couple of classifiers were trained to predict sentiment from that corpus. Their choice of the classifiers and preprocessing techniques yielded good results.

The work reported by Farra et al [12] utilized a set of dictionaries that store positive, negative and neutral roots. To determine the sentiment or class of a sentence, a stemmer was applied to transfer words into roots. If the resultant root exists in the positive/negative/neutral root dictionary, then that sentence is considered positive/negative/neutral, respectively. If the word is not in the dictionary, the user is asked to specify its polarity and subsequently its root is added to the corresponding dictionary.

The authors of [13] proposed sentiment analysis methodologies for classifying Arabic and English languages. They used specific feature extraction components that are integrated to account for the linguistic characteristics of Arabic. The outcome accuracy was good but their domain was limited to hate and extremist words. Another main drawback was the extreme lack of preprocessing which is really crucial for Arabic text.

Another work worth mentioning is the project done by Elhawary and Elfeky [14]. They used business web forums as sources of input data and extracted the business reviews written in Arabic language. These reviews were then analyzed and their sentiments were calculated.

C. *Challenges of using Arabic language*

Arabic is a morphologically rich language, this can be explained by the following points:

- A given root can take several forms depending on the context such as (أحبُ,يحب,يحبون,أحبو,تحب).
- Many people tend to use the dialect of their country instead of using MSA (شاهد= شوفت).
- Repeating the letter more than once to intensify the meaning or feeling (A common style found in informal channels) such as جداااااااااااااااااااااااا, in MSA is written as جدا which means extremely too much.
- Arabic has various diacritics; based on the presence or absence of such diacritics, the meaning of words can be totally different. For example, "teacher" "مُدَرّسة" and "school" " مَدْرسَة", both can be read as the same word when written without diacritics "مدرسة".
- Negation words that are used to negate past or present tense verbs, which change the meaning of the verb to exactly the opposite. e.g.لم أعجب بهذا الكتاب. "I didn't like this book."

III. The Sentiment Analysis Framework

The process that was followed in this work consists of the following phases:

1. Collect the training dataset: A PHP script was written to interact with Twitter's Search API [15] to fetch the tweets based on certain search queries.

2. Label the training dataset: Now, in order to come out with a tool that has good level of accuracy, a huge number of tweets are needed to be annotated. Doing that by the authors of this paper only was almost impossible, so crowdsourcing was the answer. To get the help of crowdsourcing, we've created a well-organized, easy to use API, written in both PHP and SQL scripts. This API gives users the ability to login as an administrator (one of the authors) or simply as a normal user. The API's interface displays the tweet along with four choices: Positive, Negative, Neutral and Not Applicable. The user chooses the most relevant label to the tweet.

3. Analyze the dataset to gather information about what is the best methodology to normalize the data.

4. Normalize the data in the preprocessing phase.

5. Create the feature vector for each entry in the dataset.

6. Input the feature vectors into the classifier to build the classification model.

7. Use the model to verify results using cross validation on the same training set.

The following subsections explain the above phases in more details.

A. *Crowdsourcing Revisited*

The process begins with having all the fetched tweets enter the main dataset, where they will be saved. These tweets afterwards are filtered in order to decide which ones will move to the next stage. The filtering process is done on specific criteria:

1. Each tweet must be at least 100 characters.

2. Tweets shouldn't contain more than 4 Hashtags [16].

3. A tweet is preferred to be free of links and mentions.

4. Tweets that are duplicates or retweets are eliminated.

After that, the filtered tweets are moved to the next stage, and enter a smaller dataset called the “ToBeRated” dataset where they will be fed to the rating API, in order to be annotated. This new dataset contains at least 1000 tweets that are ready to be rated at any time. The user chooses the sentiment of the tweet from his/her point of view and the answer is saved in the database.

After the tweet is rated it moves back to the “ToBeRated” dataset and waits its turn to enter the final stage. The final dataset contains only tweets that are assured to be rated correctly. A tweet is considered rated correctly if the label assigned by the supervisor matches the label assigned by non-supervisor. If the previous two labels are dissimilar, a third rater comes to the rescue, and the majority voting decides the final outcome. If these previous conditions are met then the tweet is moved to the final stage and is then used in the subsequent phases. A supervisor rater is one of the authors, and non-supervisor rater is the anonymous users who volunteered. This was used as a quality assurance step. Figure 1 shows the labeling or rating process. At the time of this writing we have more than 350,000 Arabic tweets and 25000+ rated tweets for the training dataset.

B. *Preprocessing*

Text pre-processing is an important stage in text mining. The major obstacle in text mining is the very high dimensionality and the large size of textual data. Natural language processing and morphological tools can be employed to reduce dimensionality and size of text data. Rapidminer (http://rapidminer.com/) has a collection of operators that are suited to text mining such as stemming, tokenization, and filtering unwanted words. In addition to that, we have extended Rapidminer to suite the current work. The following paragraphs describe the preprocessing that was applied to the tweets.

First, we need to split the text of the tweet to separate tokens, and this task comes with its own problems, because words are not split by just a space in informal text sometimes it comes with extra characters that need to be handled. We decided to split the text using space, comma, semicolon, colon and dot as delimiters. After that the text of the tweet was normalized following the rules shown in Table 1.

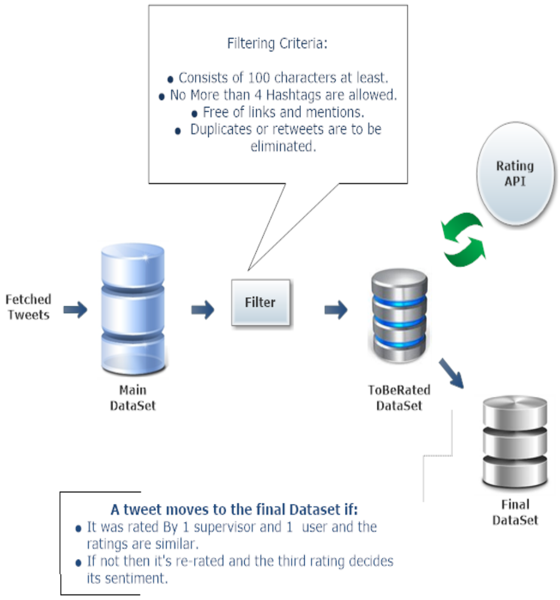
Filtering Criteria:
Consists of 100 characters at least.
No More than 4 Hashtags are allowed.
Free of links and mentions.
Duplicates or retweets are to be eliminated.
Rating API
Fetched Tweets
Filter
Main DataSet
ToBeRated DataSet
Final DataSet
A tweet moves to the final Dataset if:
It was rated By 1 supervisor and 1 user and the ratings are similar.
If not then it's re-rated and the third rating decides its sentiment.

Fig. 1: The labeling process

TABLE I: NORMALIZATION OF TWEETS

Value To Replace	Replace By
أ	ا
إ	ا
.	Null
,	Null
“	Null
‘	Null
(	Null
)	Null
Space	Null

Filtering Arabic stopwords was applied as a third preprocessing step. The built-in Rapidminer dictionary of Arabic stopwords was used. This, however, has created problems with valence shifter such as negation. Negation letters are treated as stopwords for topical classification but for sentiment classification they are vital as they can reverse the polarity of the sentence. To overcome this problem, we have built our own dictionary of Arabic stopwords and integrated it to Rapidminer. Lastly, stemming and light stemming were applied to the tweets. We have experimented with these to judge which stemming technique is more suitable for sentiment analysis.

C. Extensions to Rapidminer

Rapidminer has helped us in most parts of the process however it lacked some support for Arabic especially for normalizing and preprocessing, but it offered a way to extend its functionality with extensions and custom operators with its JAVA API. We started to develop the following operators under one package (extension).

1. Emoticons Convertor: Changes emoticons in tweets to their respective meanings in language.

2. Repetitions Remover: A simple regular expression based repetition remover to enhance stemming and other operations

3. Negation Detection: Detects negation and handles it properly by either removing spaces between the negation operator and the word or by changing the word to a form of our choice.

4. Dialect to MSA convertor: Maps dialect words to MSA.

5. Arabizi Convertor: This is used to convert words in tweets that are written in Latin letters to Arabic so we can apply stemmers on them and other tools to normalize them for classification.

6. Links Remover: removes links from tweets

7. Mentions Remover: Unnecessary features in the tweets that might lower accuracy like links and mentions are removed from the tweets before applying the classifier.

D. Dictionaries

In the process of creating the above operators we needed to create the following dictionaries.

Jordanian dialect to MSA parallel dictionary

After searching the Internet and going through books, we couldn't actually find a full dictionary for Jordanian dialect. And due to the importance of having such dictionary in our project we decided to manually build a new one considering the main features that can help us here which is the most common used words in social websites. The following represent the steps taken to accomplish this task:

1. Choosing 100 long chat histories between Jordanian users we collected from friends.

2. Manually extracting each word that is related to the Jordanian dialect.

3. Putting Synonyms for each Jordanian dialect word, taking into consideration that many words in our dialect can have the same meaning. We ended up having 300 words in this dictionary.

Negation Dictionary:

This contains the most used Arabic "Valence Shifters" [17] that infer the polarity of the sentence; this is used as stated earlier to detect negation and normalize them appropriately. These words were gathered in a manner similar to the dialectical dictionary mentioned above and it includes words used formally in Arabic (e.g.: لا ,لن).

Arabizi Dictionary:

This contains most used words in Arabizi which we collected from the internet and from chat logs. We built a PHP code to interact with an open API that converts the Arabizi words into Arabic, then we stored the results into a parallel dictionary, and each new word that goes through the PHP script is saved into dictionary if it's not already there.

E. Classification

A classifier is a function that maps a set of object to a set of labels [18]. In this work, it maps a set of tweets to the classes Positive, Negative and Neutral. While multi-label classification is possible in topical classification, for sentiment analysis, it is common to assign a review or tweet to a single label or class. Three built-in classifiers in Rapidminer were used, namely, Naïve Bayes (or NB) [19], k-nearest classifier (k-NN) [21], and Support Vector Machines (SVM) [20]. Because running Rapidminer on the whole dataset resulted in memory issues, we applied our framework on a basic dataset sampled using stratified shuffling from the main training dataset to get the result for our comparisons. The next section explains the results obtained.

IV. Experimentation and Result Analysis

The following tables show the types of settings that we have used. The first column represents the classifier name. The second column specifies whether a stopword filter was used (1) or not (0). The third column shows whether stemming was used (1) or not (0). The fourth column shows the number of folds in cross-validation. The fifth column shows the accuracy. These tables show the baseline experiments that used the default behavior of Rapidminer. The size of the dataset consists of 1000 tweets. k=1 for the KNN classifier.

Table 2 shows that the best performance of NB was achieved when no filtering of stopwords and no stemming were used. This can be attributed to two reasons: the stemming algorithm is not accurate and in informal tweet not many stopwords, apart from negated words, are used.

Again the best accuracy achieved by SVM was 71.68% when both stopword filter and stemming were disabled and 10-

fold cross validation was used. The accuracy of SVM is slightly lower than NB for the same setting.

TABLE II: SEVERAL TRIALS WITH NB

Classifier	Stopwords Filter	Stemming	Folds	Accuracy
NB	1	1	5	76.67%
NB	1	1	10	75.64%
NB	1	0	10	75.65%
NB	0	0	10	75.42%
NB	0	0	5	**76.78%**

TABLE III: SEVERAL TRAILS WITH SVM

Classifier	Stopword Filter	Stemming	Folds	Accuracy
SVM	1	1	5	71.23%
SVM	1	0	5	69.89%
SVM	0	0	5	69.68%
SVM	0	0	10	**71.68%**

TABLE IV: SEVERAL TRAILS WITH KNN

Classifier	Stopwords filter	Stemming	Folds	Accuracy
KNN	1	1	5	55.72%
KNN	1	0	5	56.23%
KNN	0	0	5	**59.99%**
KNN	0	0	10	51.58%

Table 4 shows that KNN performs best when stopword filter and stemming were not used. The accuracy of KNN is worse than the accuracy of NB and SVM. A common trend in the baseline experiments is that stopword filter and stemming did not improve the results.

Table 5 shows the effects of two created operators in Rapidminer, namely: Negation Detection and dialect to MSA on the accuracy of NB (Recall that NB gave the highest accuracy in the baseline settings).

As it can be seen from Table 5, Process 3 gave the best results for NB. It is interesting to note that the use of Negation Detection operator and the Dialect to MSA dictionary did not improve the accuracy of NB over the baseline accuracy (Refer to Table2). This definitely needs further investigation but as a first attempt, one could refer these not so impressive results for the extensions to the small size of the Dialect dictionary (300 words) and to the small size of the dataset (1000 tweets). However, we will have more conclusive results once we run our process on a more powerful machine that can handle the whole dataset.

TABLE V: NB PERFORMANCE WHEN EXTENSIONS ARE USED

Process	Classifier	Negation Detection	Dialect to MSA	Accuracy
1	NB	1	1	74.55%
2	NB	1	0	75.42%
3	NB	0	1	**76.78%**
4	NB	1	0	74.42%

V. CONCLUSIONS AND FUTURE WORK

This paper has addressed sentiment analysis in Arabic tweets. To this end, we have collected about 350,000 tweets. Crowdsourcing was used to label 25000+ tweets. Every tweet is assigned a label as Positive, Negative, or Neutral. Majority voting was used to decide the final label for every tweet. This word has many novel contributions such as handling negations, Arabizi and Arabic dialects. Three built-in classifiers in Rapidminer were used to assess our framework. The results obtained are promising and this encourages us to continue working on this topic. We definitely, need to expand the dictionaries and solve the memory issue which is inherent in Rapidminer.

REFERENCES

[1] Friedman, Thomas L. The Lexus and the olive tree: Understanding globalization. Farrar, Straus and Giroux, 2000.

[2] Waters, John K. The Everything Guide to Social Media: All you need to know about participating in today's most popular online communities. Adams Media, 2010.

[3] "Sentiment analysis", Wikipedia, the free encyclopedia. Last Accessed 5 May. 2013 <http://en.wikipedia.org/wiki/Sentiment_analysis>

[4] Chiang, David et al. "Parsing Arabic dialects." Proceedings of the European Chapter of ACL (EACL) 2006: 112.

[5] "Morphology (linguistics)", Wikipedia, the free encyclopedia." Last Accessed 14 Apr. 2013 <http://en.wikipedia.org/wiki/Morphology_(linguistics)>

[6] "Getting Started | Twitter Developers." Last Accessed. 5 May. 2013 <https://dev.twitter.com/start>

[7] "GET search | Twitter Developers." Last Accessed 14 Apr. 2013 <https://dev.twitter.com/docs/api/1/get/search>

[8] "Anglo-Arabic alphabet." Last Accessed 14 Apr. 2013 <http://www.omniglot.com/writing/angloarabic.htm>

[9] "Emoticon - Wikipedia, the free encyclopedia." Last Accessed 14 Apr. 2013 <http://en.wikipedia.org/wiki/Emoticon>

[10] Abbasi A, Chen H and Salem A. Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. ACM Transactions on Information Systems (TOIS) 2008; 26(3): 12.

[11] Rushdi□Saleh, Mohammed et al. "OCA: Opinion corpus for Arabic." Journal of the American Society for Information Science and Technology 62.10 (2011): 2045-2054.

[12] Farra, Noura et al. "Sentence-Level and Document-Level Sentiment Mining for Arabic Texts." Data Mining Workshops (ICDMW), 2010 IEEE International Conference on 13 Dec. 2010: 1114-1119.

[13] Almas, Yousif, and Khurshid Ahmad. "A note on extracting ‘sentiments’ in financial news in English, Arabic & Urdu." The Second Workshop on Computation, al Approaches to Arabic Script-based Languages 21 Jul. 2007: 21-22.

[14] Elhawary, Mohamed, and Mohamed Elfeky. "Mining Arabic Business Reviews." Data Mining Workshops (ICDMW), 2010 IEEE International Conference on 13 Dec. 2010: 1108-1113.

[15] "Using the Twitter Search API | Twitter Developers." Last Accessed 5 May. 2013 <https://dev.twitter.com/docs/using-search>

[16] "Hashtag - Wikipedia, the free encyclopedia." Last Accessed 15 Apr. 2013 <http://en.wikipedia.org/wiki/Hashtag>

[17] Kennedy, Alistair, and Diana Inkpen. "Sentiment classification of movie reviews using contextual valence shifters." Computational Intelligence 22.2 (2006): 110-125.

[18] "Classification theorem - Wikipedia, the free encyclopedia." Last Accessed 5 May. 2013 <http://en.wikipedia.org/wiki/Classification_theorem>

[19] "Naive Bayes classifier - Wikipedia, the free encyclopedia." Last Accessed 15 Apr. 2013 <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>

[20] "Support vector machine - Wikipedia, the free encyclopedia." 15 Apr. 2013 <http://en.wikipedia.org/wiki/Support_vector_machine>

[21] "K-nearest neighbors algorithm - Wikipedia, the free encyclopedia." Last Accessed 8 May 2013 <http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>