Mood extraction from Franco-Arabic Facebook Posts

Mariam Magdy Badr, Caroline Sabty, Nada Sharaf, Slim Abdennadher

German University in Cairo
mariam.badr@student.guc.edu.eg,{caroline.samy, nada.hamed,
slim.abdennadher}@guc.edu.eg

Abstract. Facebook is now considered as one of the most powerful. Mood extraction and sentiment analysis can significantly enhance the experience of social media users. Detecting feelings and how they change of users from Facebook can help in understanding their preferences. This could help in many aspects like obtaining the feedback of users regarding any product without having to ask them for it explicitly. Most of the previous work that targeted the mood of users from Facebook, were built to analyze English text, few work aimed at Arabic text. The aim of this work is to build a prototype to perform sentiment analysis on Franco-Arabic text as it is currently widely used by Arabic-speaking social media users.

Key words: Sentiment Analysis, Franco-Arabic, Machine Learning, Facebook, Natural Language Processing

1 Introduction

Facebook is nowadays one of the most famous social media network applications. Facebook users utilize their accounts to express their feelings and thoughts with their friends and the world. Facebook thus contains a huge amount of data that could be helpful in many aspects. If the text could be analyzed to extract how the users feel, we will have powerful data that can be useful in many situations. Such data could be used to collect feedback on products for customer care, or to understand how users feel about current events or a given entity. The work done in sentiment analysis and mood extraction from text focused on English text [1]. Some work targeted Arabic text [2].

Arabic-speaking users, however, write some of their posts in Franco-Arabic which expresses Araboc text with English characters. In other words, Franco-Arabic words are written using Latin alphabet and some numerals such as "5 which is used to represent the Arabic letter pronounced as "kh. Due to the lack of work with this representation, valuable data is not used and is being lost.

Machine learning algorithms are commonly used for different text analysis tasks. It is based on using algorithms that benefit from a data set, usually referred to as training data. The training data is random pieces of data that is collected and tagged or in other words labelled by the tags the algorithm is expected to categorize text into. In sentiment analysis those tags can be positive, negative, happy, angry or any other tags depending on the chosen categorization for the system.

The aim of this work is to extract mood from Franco-Arabic text, specifically from Facebook posts, by implementing machine learning algorithms and using Natural Language Processing tools. Training data is collected from social networks and is written in Arabic and labelled with their corresponding sentiment and used to feed the machine learning algorithm to be able to classify the input text.

There are several advantages for using machine learning techniques for sentiment analysis. For example, the fact that there are several existing natural language processing tools and machine learning algorithms that were developed for data mining from text and can be efficiently used for text sentiment analysis. In addition, the data sets used for training can be collected from the most suitable environment depending on the aim of the sentiment analysis tool and what it is originally developed to serve. For the work presented in the paper, it would be most suitable to use a data set collected from social media. Additionally, more training data can be fed to the algorithm at anytime to improve the accuracy of the results without affecting the implementation of the application.

The paper is organised as follows. Section 2 presents some related work and the most common approaches used for sentiment analysis. Section 2 discusses the challenges in the field of sentiment analysis and dealing with franco-Arabic text. Section 4 presents the system architecture and goes into details of the steps of implementation. Section 5 concludes the work presented and presentes some related work.

2 Related Work

Much work aimed at extracting mood and feelings from text. However, most studies them targeted the English language. Some work has been done to analyse Arabic text, but almost non targeted Franco-Arabic text. The two main approaches used for sentiment analysis are machine learning and emotion lexicons.

The machine learning approach is based on using algorithms that benefit from a data set, usually referred to as training data. The training data is collected and tagged or in other words labelled by the tags that the algorithm is expected to categorise text into. In sentiment analysis those tags can be 'positive', 'negative', 'happy', 'angry' or any other tags depending on the system chosen for categorisation.

In [3] followed this approach and performed tests using three independent classifiers for Stress detection in computer users through non-invasive monitoring of physiological signals [3] and found that support vector machines (SVM) had the best performance in detecting and identifying user stress states.

On the other hand, according to the Oxford dictionary [4], a lexicon is “the vocabulary of a person, language, or branch of knowledge”. An *emotion lexicon* is one were words are available with their corresponding emotion categories. Some emotion lexicons follow a mono-emotion concept where each word is only labelled by one emotion. Other lexicons allow the labelling of words with two or even more labels depending on the word itself.

The work done for identifying expressions of emotion in text in [5] follows a model based on a Non-negative Matrix Factorisation (NMF) approach which makes use of emotion lexicons for the purpose of sentiment classification. The Word-Emotion Association Lexicon (NRC) [6] is an English emotion lexicon that also has versions in other languages, including Arabic. It classifies entries into two sentiment categories: positive and negative, and eight emotions: anger, anticipation, disgust, fear, joy, sadness, surprise and trust. On the other hand, Sentiment Lexicon for Arabic Social Media [7] is a list of Arabic terms and their associated sentiments. The categorisation is shown by the use of numbers or as they call it a (sentiment score). If the score for word x is higher than the score for y, then x is said to be more positive than y. The lexicon has more that 22,000 positive terms and 20,000 negative terms, and it is extracted automatically from tweets that have certain features they call (seed terms). Those seeds used to create the lexicon are a set of 23 positive and negative emoticons such as :) and :(which are referred to on social media as “smiley faces”. A tweet is considered positive if it has a positive seed, and negative if it has a negative one.

Another interesting lexicon is Thawra [8]. Thawra is a three way lexicon that has entries in Dialectal Arabic, Modern Standard Arabic and English correspondents. It focuses on Egyptian Arabic dialectal Arabic. However it is not an emotion lexicon so words are not tagged with emotions.

3 Sentiment Analysis Challenges

There are different challenges in the field of sentiment analysis. Some of the challenges depend on the implementation choices others are common challenges regardless of the choices made.

On extracting mood or feelings from text, the first problem is *uncertain words and phrases.* Some words or phrases do not have sentiment polarity on their own. In other words, they would give different meanings and different feelings in different sentences; one can only understand the feelings and emotions associated with them when reading the whole sentence or the whole text. For example: “We’re back to school” in English and it equivalent Franco-Arabic statement “rge3na lel madrasa”, is an example of an uncertain phrase. One can not tell how the person speaking feels about being back to school.

Another challenging point is *sarcasm* because text, especially on social media, can be sarcastic in many situations. For example “Oh, great!” can be be used to express the feeling of happiness or excitement regarding a certain topic which means positive on the sentiment scale. However in other cases the person who wrote it might be sarcastic and trying to express their disappointment or sadness regarding the topic being discussed which would correspond to negative on the sentiment scale. The same concept applies to Arabic and Franco-Arabic text, for example “eih el gamal da” or “mesh 2ader 22olak 2ad eih mabsoot”, are both sentences that if taken literally would give a positive feeling, while in reality they are widely used in sarcastic situations to express negative feelings.

In addition, text usually contains words that are said to be *emotionless words.* Those words have no significance when it comes to feelings or the mood of a person. However they are widely used in texts for various reasons such as to form a full understandable sentence or to build a grammatically correct sentences. Examples of those words in the English language would be 'The', 'and', 'here' and 'because'.

One of the toughest challenges is *Variance.* Generally human beings are different. We have different ways of expressing ourselves and our feelings. As a result of that, a text can be viewed by a person and labelled as 'happy'. However another person would label the same text as “surprised”. In other words, the way of one person to express negative feelings can be another person's way to express positive ones.

Other challenges might be related to the language being used because each language is different and has its own complexities. When classifying Arabic text *inflectional morphology* might cause a few challenges, it is when a spelling of a word might vary depending on its position grammatically in text. Difficulties can also be caused by *derivational morpheme*, it is when an affix is added to words to create a new form of the word. In Arabic this is the case when using a verb to address different genders for example. The reason those properties might cause some trouble is that the algorithm might have the same word in training data set and data that needs to be classified, but fails to recognise that it is the same resulting in faulty or less accurate classification.

Last but not least, more challenges result from dealing with Franco-Arabic because text written in Franco-Arabic is not written in Modern Standard Arabic (MSA) which is used in official settings. It is written in spoken version of Arabic used in everyday life which is known as Dialectal Arabic (DA).

This work focuses on the Egyptian version of DA. The problem with written DA is that it has no standard or fixed spelling for its words. To overcome this problem natural language processing tools such as stemming are applied to training data as well as the text to be analysed.

There are different ways to deal with text written in Franco-Arabic each having its pros and cons. Most of the previously done work in the field of sentiment analysis and mood extraction was developed to analyse English text. In addition, most of the available data sets as well as emotional lexicons are in English.

One approach would be to translate text into English first then extract emotions. Other than the availability of more data sets, lexicons and previous work, a pro of using this approach would be that most language processing tool kits have functions and tools developed to work on English text. For example there is the part of speech tagging in Natural Language Toolkit (NLTK) [3] that can be used to identify part of speech of every word and label it accordingly. Consequently those labels can be used to construct regular expressions that extract only words labelled with certain part-of-speech tags and in a specific order from the text. There are also available stop-words list for English text, those are lists of words that can be filtered and removed from text before processing it (as they have no significance and might affect the results negatively if considered when training the classifiers, 'the', 'those', 'who' and 'him' are all examples of English stop words).

However, this approach has a major drawback, to translate text from Franco-Arabic to English it must go through two or three steps of translation. It has to first be translated from Franco-Arabic to Arabic words written in Arabic letters. Since Franco-Arabic is considered as an informal typing style, those resulting words would most probably be written in DA not MSA. Thus they will need to be translated to MSA which in turn will be translated into English text. The resulting text might keep its meaning or deliver the same message however there is a concern that it would lose its feelings and emotions along the way.

A second approach would be stopping at the step where we have the text written in Arabic letters, whether in Dialectal Arabic or Modern Standard Arabic, and classifying the text at this point. This might have the same drawback as the previous approach but at a much lower degree since words would still be of the same language and even the same pronunciation with just different spelling. Thus, the probability of expressing different feelings than the ones in the original text is lower. As for its pros, they are also similar to those of the previous approach because there are available data sets and tools that can be used for natural language processing on Arabic text, but noticeably less than those available for English.

The third and last approach would be to use the text as it is, written in Franco-Arabic. This method guarantees keeping the emotions and meanings of the original text, as there is no chance of loss because the words entered by the users are the ones being classified. However, since up to our knowledge very little to almost non work and researches have been done on Franco-Arabic text, there are no available data sets or emotion lexicons that can be used for sentiment analysis. Another drawback is that the available preprocessing tools would not be able to recognise or understand text that is written in Franco-Arabic. Even the tools that are built to handle Arabic text would not recognise an Arabic word written in any alphabet other than the Arabic alphabet.

The approach implemented in our prototype follows the third approach with some changes to overcome some of its drawbacks. On one hand, the text is kept as Franco-Arabic and mapped to its equivalent Arabic letters. Then, it is preporocessed and given to the trained machine learning algorithm to be labelled. On the other hand, the machine learning algorithm is trained on an adjusted and preprocessed set of dialectal tweets as will be explained in details in Section 4.

4 System Architecture

The system architecture is based on the client-server model. Client-server architecture is usually one where there are servers, which are computers or processes typically used to keep the resources needed by client such as files and processing power. The clients are personal computers or workstations where the users run applications. Figure 1 gives a simple idea of how the server and the client sides are integrated together, and how data is exchanged between them.

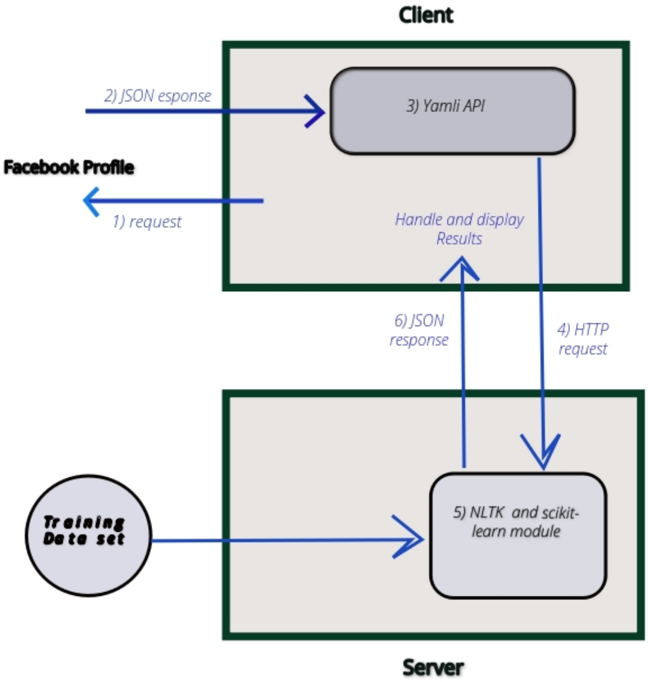


Fig. 1. Client-server interaction.

4.1 Server

The server side is mainly responsible for the sentiment analysis of the text. It stores the training data sets, negative and positive, after they have been preprocessed. On receiving text from the client side, it preprocesses the text then classifies it, into positive or negative sentiment, by applying machine learning algorithms on it. At the end, it returns back the results to the client side.

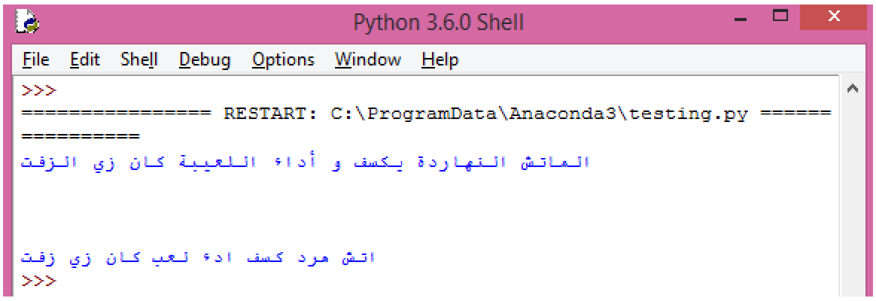
The data-set used for training the algorithm is the ASTD: Arabic Sentiment Tweets Dataset [9]. It consists of more than 10000 tweets, collected from the online social networking service Twitter. Entries are labelled in positive and negative tags, and are written in DA using the Arabic alphabet. However, since it is written in Moroccan accent, a few entries, written in Egyptian dialectal

Arabic, consisting of commonly used Egyptian positive and negative phrases and statements were manually added to the data set.

4.1.1 Preprocessing Text Before using the data to train the algorithms, it needs preprocessing first. Preprocessing is a form of data mining where real-world data is filtered and changed to some extent to result in a data set that is more suitable for the purpose it is needed for. This can be achieved with the help of Natural Language Processing (NLP) tools. Most of the preprocessing was done using Natural Language Toolkit (NLTK)[1] as it has several tools that can work on Arabic text.

The first step is to tokenize the data, or in other words split it into separate words, this was done using the tokenization module of NLTK. Afterwards, the data needs to be stemmed. Stemming is applied to convert a word back to its base which is also named the stem of the word. In other words, it removes extra letters that might be added to the word due to its grammatical position. This is achieved by the help of the 'ISRIStemmer' which is developed specially to stem Arabic text. Unfortunately, not all the words are reduced perfectly because the data set is in dialectal Arabic which does not follow all the rules of modern standard Arabic.

Now that the data set is an array of separated stemmed words, the next step would be to remove the stop words. NLTK libraries and modules do not contain Arabic stop words, so a list of Arabic stop words was manually added and used to filter out those words from the data set. Afterwards, all words are stored as tuples having two pieces of data. The first part is the stemmed word itself and the second is its polarity, positive or negative. This data is then pickled. Pickling and unpickling are performed using the pickle Python module, to pickle means to convert a Python object into a stream of bytes, and to unpickle is the opposite process. The following figure shows an example of a simple sentence labeled as negative, and written in Egyptian dialectal Arabic, before and after being preprocessed.



```
>>>
================ RESTART: C:\ProgramData\Anaconda3\testing.py ======
==========
الماتش النهاردة يكسف و أداء اللعيبة كان زي الزفت



اتش مرد كسف اداء لعب كان زي زفت
>>>
```

Fig. 2. Arabic text before (top) and after (bottom) preprocessing.

[1] http://www.nltk.org/

As shown in Figure 2 and as mentioned earlier, some of the words, like the first word in the top sentence which is supposed to mean ”the match” was changed into a stem that is unrelated to the original word. This is due to the word being too far from its original corresponding word in MSA. However other words were successfully changed into their roots.

4.1.2 Classification After preparing the data, the second step would be working on the machine learning algorithms that will classify it. The classifiers used here are all Scikit learn[2] classifiers. To improve performance and increase the confidence of the classification result a voted classifier was implemented instead of using a single classifier. Voted classifier is basically a method that would run different classifiers on the data and count which sentiment or emotion was the assumption of each classifier. Based on the counts, it assumes which sentiment has the highest probability which in turn would be the output [10].

Another method was also implemented that calculates the confidence of the result or the assumption made by the voted classifier. Confidence is based on the number of votes in favour of the result in comparison with the total number of classifiers used. Since in this model there are only two possible outputs, positive and negative. Thus we can guarantee that the confidence of any result would always be above fifty percent since otherwise the other sentiment would be the output.

There are several algorithms developed for data mining such as C4.5, k-means, Support vector machines, Apriori, EM, PageRank, AdaBoost and many more. Five classifiers are being used in the voted classifier, those are: NuSVC classifier, Linear SVC classifier, Multinomial Naive Bayes classifier (MNB), Bernoulli Naive Bayes (BNB) classifier, and Logistic regression classifier [11–16]. NuSVC and Linear SVC classifiers are both optimisation of the support vector classifier (SVC) which is based on , however linear SVC scales better to large numbers of samples. MNB and BNB are both instances of the Naive Bayes classifier, the difference is that MNB uses a multinomial distribution for each of the features by calculation the occurrence and frequency at which specific words occur in the text. While BNB is more focused on modelling the absence of words from a text which makes it more suitable and gives better results for the classification of short texts.

Naive Bayes is a supervised algorithm which means it needs to be provided by a tagged training data set. In fact, Naive Bayes is not a single algorithm it is a family of classification algorithms, all those algorithms share the assumption that the features of the data to be classified do not affect one another. In other words they assume that the features are independent, this is originally the reason why the algorithm was given the name ”Naive”, as in real world applications it is very rare to find a data set were every single feature is independent and not affecting or being affected by the others. Naive Bayes is considered one of the simplest algorithms in terms of calculations, it classifies data using the following

[2] http://scikit-learn.org/stable/

equation:

$$P(ClassA|Feature1, Feature2) = \frac{P(Feature1|ClassA).P(Feature2|ClassA).P(ClassA)}{P(Feature1).P(Feature2)}$$

The equation states that the probability of classifying an item of data, having feature 1 and feature 2, as belonging to class A can be calculated by multiplying the probability of having Feature 1 given Class A, probability of having Feature 2 given class A, and the probability of Class A, then dividing the result by probability of Feature 1 multiplied by probability of Feature 2. Despite its simplicity Naive Bayes is found to be surprisingly accurate and effective in several applications such as spam filtering. There are also more accurate versions of Naive Bayes such as Bernoulli Naive Bayes and Multinomial Naive Bayes.

Support vector machines (SVM) is also a supervised algorithm but to understand what it does, one must first be introduced to Hyperplane. hyperplane is a function that acts like the equation of a straight line, specially when used for classification tasks where the data has only two features. SVM finds out the best hyperplane to separate the data set and classify it into two classes. For example for a data set where data points either belong to Class A or Class B, SVM will find the equation for the best line that would separate the mixed data points with the highest percentage of separation, accordingly any data point can be classified into Class A or Class B by figuring out its position relative to the line. SVM can also operate at higher dimensions were there are more than two features, this is done by the use of a kernel where data is mapped from 2-dimensional surface into more dimensions, however in that case the hyperplane used will not be a line but can be a plane instead.

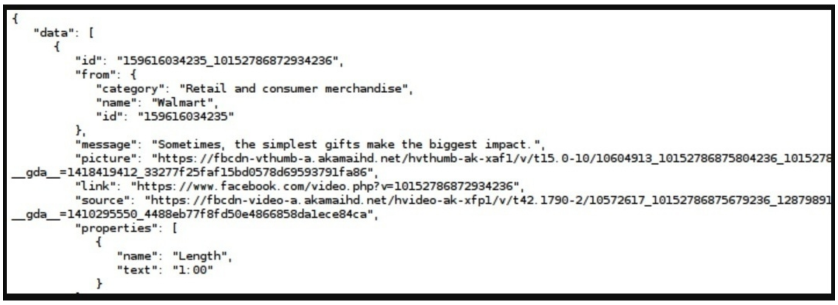
4.2 Client

The client side of the project is responsible for collecting the data to be classified from the Facebook profile of the user. In addition, preparing the collected data to be ready for classification when sent to the server side. This is done by translating the text from being written in Latin alphabet into Arabic alphabet, both being in Dialectal Arabic. The client side is implemented in the form of a Facebook application that is hosted as a website by Github Pages[3], and can be reached through the Facebook application by any Facebook user.

4.2.1 Obtaining Text from Facebook And since the main aim of this work is to extract mood from Facebook posts. the first step is using Python Facebook module and urllib modules, which leads to getting the access token needed. However, due to Facebook security hazards it is not possible to get a post, not even a public one. For this reason a Facebook application was developed using Facebook's Graph API and Facebook Login and share products were added to it. The Graph API provides the application ID and the application secret which are essential for accessing any post on Facebook. The Facebook Login product

[3] https://pages.github. com/

allows Facebook users to login to the application. Moreover, it gives the needed permissions and authorization to access their profile wall posts. The Facebook share product allows the users to share the application on their personal profiles. With all the permissions being granted, The Facebook application returns the requested posts, to the website hosted on Github Pages, in JSON objects. The client side handles those responses and obtain the needed text in the form of strings. The following Figure 3 shows an example of a simple Facebook JSON file which might be returned after a request being sent to Facebook API (Just one post).



```
{
   "data": [
      {
         "id": "159616034235_10152786872934236",
         "from": {
            "category": "Retail and consumer merchandise",
            "name": "Walmart",
            "id": "159616034235"
         },
         "message": "Sometimes, the simplest gifts make the biggest impact.",
         "picture": "https://fbcdn-vthumb-a.akamaihd.net/hvthumb-ak-xaf1/v/t15.0-10/10604913_10152786875804236_1015278
__gda__=1418419412_33277f25faf15bd0578d69593791fa86",
         "link": "https://www.facebook.com/video.php?v=10152786872934236",
         "source": "https://fbcdn-video-a.akamaihd.net/hvideo-ak-xfp1/v/t42.1790-2/10572617_10152786875679236_1287989l
__gda__=1410295550_4488eb77f8fd50e4866858da1ece84ca",
         "properties": [
            {
               "name": "Length",
               "text": "1:00"
            }
```

Fig. 3. An example of a JSON file returned by Facebook API

4.2.2 Translating Text Since classifiers are trained to classify text that is written in Arabic letters, a tool was needed to translate the text obtained from Facebook from Latin letter to Arabic letters. For translating words from Franco-Arabic written in Latin alphabet into Arabic written in Arabic letters Yamli [4] was used. Yamli is an internet start-up that offers 'the smart Arabic keyboard' product which is used by users to change Arabic text that is written in Latin letters into Arabic text written in Arabic letters. In other words it allows users to write in Arabic letters without using an Arabic keyboard as users are not so familiar with the Arabic keyboard. Yamli API[5] allows smart Arabic keyboard technology to be integrated with any website to "Yamlify" all or some of the text boxes. The code of the API can be modified according to the developers preferences regarding the settings of Yamlified text boxes such as enabling/disabling Yamli or even the fonts and the colors used.

Unfortunately, when using Yamli API there was an obstacle which is the fact that the API yamlifies text areas and not strings. This means that the API was build to yamlify words at the moment they are being typed by a user into a text area that has been yamlified, and not a whole sentence. In other words, Yamli API mainly depends on event listeners from the keyboard and mouse of

[4] http://www.yamli.com/

[5] http://www.yamli.com/api/

the user, where as the user types and presses "space" for example a single word gets yamlified. However if the user "pastes" a whole sentence without typing it word by word, the sentence remains as it is. This is because Yamli offers several variants or translations for each word, and those are displayed for users as they type so that they can choose the most suitable one, or just hit space and in this case the first option will be automatically chosen.

Consequently, some changes were applied to the Yamli open source code [6]. Upon receiving the sign to translate, which is done by the help of listeners, those changes in the source code cause it to loop on the whole text and create objects for each word. Those objects are then send to methods that yamlify the words, and each word gets yamlified on its own. This sets the translation of every word to the first option in the list of possible translations then returns the whole sentence in Arabic letters after it has all been yamlified. The yamlified sentence is preprocessed in the same way the training dataset was preprocessed like in Section 4.1.1.

Now, that the text is ready to be categorized the client side sends an HTTP request to the server side, where the text to be categorized is passed through the url. The server side handles the request and obtains the text to be categorized, and applies the machine learning algorithms as explained before. The result, being 'positive' or 'negative' sentiment is then returned back to the client side in the form of a JSON response. The client side then handles the response and displays the result to the user.

5 Conclusion

Work on sentiment analysis on Arabic text is needed and specially text written in Franco-Arabic as it is increasingly being used on social networking tools. In addition, classifying it can help enhance both the users' and suppliers' experience. In the prototype of the system presented here most of the work was mainly focused on two parts. The first one is natural data processing to be able to get roots of words that were written in dialectal Arabic. Moreover, to be able to recognize a word even when written in different spelling. The second one would be classifying the text using machine learning algorithms and classifiers. Using just one classifier was not enough for classifying Franco-Arabic text due to the several challenges and limitations so five classifiers were used, each having different features, to help reach more accurate classification. The prototype implemented was tested on 50 different Franco-Arabic Facebook posts. It achieved a result of 85% correctly classified posts. The 15% wrongly classified posts are due to the fact of having different dialects in the training dataset.

Future improvements would preferably be focused on the machine learning algorithms and classification approaches used. The model can be enhanced from a uni-gram model into a bi-gram one, which would classify text according to relation between every two consecutive words not one by one, also classification

[6] `http://api.yamli.com/js/yamli_api.js`

can be made into two levels, one which is implemented in this model, and another that uses lexicons. In addition, the system should be widely used and tested by a big number of users.

References

1. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta, European Language Resources Association (2010)
2. El-Halees, A.: Arabic opinion mining using combined classification approach. (2011)
3. Bird, S., Klein, E., Loper, E.: 5. In: Categorizing and Tagging Words. O'Reilly (2009)
4. Dictionary, O.E.: The Oxford Dictionary and Thesaurus. Oxford: Oxford University Press (1999)
5. Aman, S., Szpakowicz, S.: Identifying expressions of emotion in text. In: International Conference on Text, Speech and Dialogue, Springer (2007) 196–205
6. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word-emotion association lexicon. **29** (2013) 436–465
7. Saif M. Mohammad, M.S., Kiritchenko, S.: Sentiment lexicons for arabic social media. In: Proceedings of 10th edition of the the Language Resources and Evaluation Conference (LREC), Portorož, Slovenia (2016)
8. Diab, M.T., Al-Badrashiny, M., Aminian, M., Attia, M., Elfardy, H., Habash, N., Hawwari, A., Salloum, W., Dasigi, P., Eskander, R.: Tharwa: A large scale dialectal arabic-standard arabic-english lexicon. In: LREC. (2014) 3782–3789
9. Nabil, M., Aly, M.A., Atiya, A.F.: ASTD: arabic sentiment tweets dataset. In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., Marton, Y., eds.: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, The Association for Computational Linguistics (2015) 2515–2519
10. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: a new explanation for the effectiveness of voting methods. Ann. Statist. **26** (1998) 1651–1686
11. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y., et al.: Top 10 algorithms in data mining. Knowledge and information systems **14** (2008) 1–37
12. El-Manzalawy, Y., Honavar, V.: Wlsvm: integrating libsvm into weka environment. Software available at http://www. cs. iastate. edu/yasser/wlsvm (2005)
13. Schapire, R.E., Freund, Y., Barlett, P., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. In Fisher, D.H., ed.: Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997, Morgan Kaufmann (1997) 322–330
14. McCallum, A., Nigam, K.: A comparison of event models for naive Bayes text classification. In: Learning for Text Categorization: Papers from the 1998 AAAI Workshop. (1998) 41–48
15. McLachlan, G.: Discriminant analysis and statistical pattern recognition. Volume 544. John Wiley & Sons (2004)

16. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their applications **13** (1998) 18–28