A framework for Arabic sentiment analysis using supervised classification

Rehab M. Duwairi*

Department of Computer Information Systems, Jordan University of Science and Technology, Irbid 22110, Jordan
Email: rehab@just.edu.jo
*Corresponding author

Islam Qarqaz

Department of Computer Science, Jordan University of Science and Technology, Irbid 22110, Jordan
Email: islam.qarqaz@hotmail.com

Abstract: Sentiment analysis aims to determine the polarity that is embedded in people comments and reviews. Sentiment analysis is important for companies and organisations which are interested in evaluating their products or services. The current paper deals with sentiment analysis in Arabic reviews. Three classifiers were applied on an in-house developed dataset of tweets/comments. In particular, the Naïve Bayes, SVM and K-nearest neighbour classifiers were employed. This paper also addresses the effects of term weighting schemes on the accuracy of the results. The binary model, term frequency and term frequency inverse document frequency were used to assign weights to the tokens of tweets/comments. The results show that alternating between the three weighting schemes slightly affects the accuracies. The results also clarify that the classifiers were able to remove false examples (high precision) but were not that successful in identifying all correct examples (low recall).

Keywords: sentiment analysis; sentiment classification; opinion mining; polarity detection; supervised learning; text mining; Arabic language.

Reference to this paper should be made as follows: Duwairi, R.M. and Qarqaz, I. (xxxx) ‘A framework for Arabic sentiment analysis using supervised classification’, Int. J. Data Mining, Modelling and Management, Vol. X, No. Y, pp.000–000.

Biographical notes: Rehab M. Duwairi received her BSc degree (1989) in Computer Science from Yarmouk University, Jordan; MSc and PhD degrees in Computer Science from the University of Wales, Cardiff, UK, in the years 1994 and 1997, respectively. She is currently working for Jordan University of Science and Technology as an associate professor of computer science. Her research interests include object oriented databases, data mining, semantic integration of structured and unstructured data, and Arabic text categorisation. She is a senior member of ACM.
1 Introduction

Sentiment analysis or opinion mining is a field of study which attempts to analyse people’s opinions, sentiments, attitudes, and emotions on entities such as products, services, and organisations. In the literature, researchers have used the phrase sentiment analysis (Nasukawa and Yi, 2003), and other researchers have used the phrase opinion mining (Dave et al., 2003). Several authors would use the phrases sentiment analysis and opinion mining interchangeably. In this work, we will also use the two phrases interchangeably.

Sentiment analysis can be viewed as a classification process that determines whether a certain document or text was written to express positive or negative opinion. Sentiment classification would be helpful in business intelligence applications and recommender systems (Glance et al., 2005). There are also potential applications to message filtering (Malouf and Mullen, 2008).

In recent years, sentiment analysis has gained considerable attention, and its applications have spread to almost every possible domain. Many systems and applications are built for the English and other Indo-European languages. However, few studies have focused on the Arabic language which is a native language for more than 450 million speakers.

This paper is concerned with detecting sentiment embedded in public Arabic tweets and Facebook comments. In general, there are two approaches for sentiment analysis: the machine learning or supervised learning approach and the lexicon-based or unsupervised learning approach. Supervised learning requires a labelled dataset – which is expensive to collect and label. Unsupervised learning, on the other hand, relies on the use of sentiment lexicons which are built with the assumption that words have prior sentiments on their own. Building such lexicons is time-consuming and difficult. For this particular work, we have used supervised learning for sentiment analysis. The K-nearest neighbour (KNN), support vector machines (SVM) and the Naïve Bayes (NB) classifiers were run on an in-house collected and annotated dataset. When dealing with text, the most common model used to represent the text is the bag of words (BOW) model. BOW is based on two assumptions. Firstly, words order is unimportant and, secondly, words are independent of each other. Hence, under BOW, tweets/comments are represented as vectors of terms or tokens. Every term or token is given a weight. Too many schemes were used as weighting functions. The common and easy to understand weighting functions are borrowed from the Information Retrieval community. These include binary model (BM), term frequency (TF) and term frequency inverse document frequency (TFIDF). This paper addresses the effects of weighting schemes on sentiment analysis in addition to effects of the classifiers. All the experimentations were designed using Rapidminer (2015) which is an open source data mining and machine learning tool.
The contributions of this paper can be summarised by:

- collecting and annotating an Arabic dataset which is suitable for sentiment analysis
- investigating the effects of several term weighting schemes or functions on the accuracy of sentiment analysis tasks
- investigating the suitability of the classifier choice on the accuracy of sentiment analysis.

The results reveal that NB and SVM did better jobs in sentiment analysis when compared with KNN. The results also show that varying between BM, TF and TFIDF slightly affect the results and thus perhaps it is more realistic to use BM to weight terms as it is inexpensive to compute.

The rest of this paper is structured as follows: In Section 2, we describe some of the related works. In Section 3, on the other hand, we present the software that was used in this manuscript in addition to the dataset. Section 4 presents supervised classification and briefly describes the three classifiers which we have used. Section 5 discusses experimentation and result analysis. Finally, in Section 6, we discuss the conclusions of this study and highlight some future work.

2 Related work

Researchers have proposed many different approaches for sentiment analysis. In general, there are two main methods, the first one is using machine learning techniques or supervised techniques which are presented in this paper, and the other one is unsupervised techniques. Many studies have focused on the sentiment analysis for the English language and other Indo-European languages. Pang and Lee (2004) used machine learning techniques for sentiment classification. They employed three classifiers: NB, maximum entropy classification, and SVM. Their data source was the internet movie database (IMDB); they selected only reviews where the author rating was expressed either with stars or some numeral value. Dave et al. (2003) proposed an approach, which begins with training a classifier using a corpus of self-tagged reviews available from major websites. They decided to use n-grams on two tests and the result showed that this way is better than traditional machine learning.

Many researches were introduced to analyse sentiment and extracting opinions from the WWW. This proved to be important due to the large amount of data contributed by users in websites such as social networks (Facebook, Twitter, etc.). For example, Hassan et al. (2012) studied semitic sentiment analysis of Twitter. The authors used three different Twitter datasets for their experiments. They proposed the using of semitic features in Twitter sentiment classification and explored three different approaches for incorporating them into the analysis with replacement, augmentation, and interpolation. In Kumar and Sebastian (2012), the authors presented a novel approach for sentiment analysis on Twitter data. To do that, they extracted the opinion words in tweets.

There are few studies for sentiment analysis for the Arabic language. For example, Abdul-Majeed and Diab presented a newly developed manually annotated corpus of modern standard Arabic (MSA) together with a new polarity lexicon (Abdul-Mageed et al., 2011). They ran their experiments on three different pre-processing settings based on tokenised text from the Penn Arabic Treebank (PATB). They adopted two-stage
classification approach, in the first stage they built a binary classifier to sort out objective from subjective cases. For the second stage, they applied binary classification that distinguishes positive from negative cases.

In Abdul-Mageed and Diab (2012), the same researchers in Abdul-Mageed et al. (2011) reported efforts to bridge the gap between Arabic researches by presenting AWATIF; a multi-genre corpus for modern standard Arabic for subjectivity and sentiment analysis (MSA SSA). They extend their previous work by showing how annotation studies within subjectivity and sentiment analysis can both be inspired by existing linguistic theory and cater for genre nuances.

Almas and Ahmed (2007) target three languages (English, Arabic, and Urdu) in their work. They described a method for automatically extracting specialist terms called local grammar. The authors compared the behaviour of single and compound tokens in specialist and general language corpora to determine whether a token is behaving like a sentiment term or not.

Elhawary and Elfeky (2010) extract business reviews scattered on the web written in the Arabic language. They built a system that comprises two components: a reviews classifier that classifies any web page whether it contains reviews or not, and sentiment analyser that identifies the reviews’ text if it (positive, negative, neutral or mixed).

El-Halees (2011) presented a combined approach that extracts opinions from Arabic documents. He used a combined approach that consists of three methods: first, the lexicon-based method which classifies some documents. Second, the classified documents are used as a training set for maximum entropy model, last the KNN classifier is used to classify the rest of the documents.

Rushdi-Saleh et al. (2011) presented an opinion corpus for Arabic (OCA). It composed of Arabic reviews extracted from specialised web pages related to movies and films using Arabic language. They utilised two classifiers, namely: SVM and NB.

Al-Subaihin et al. (2011) proposed a sentiment analysis tool for modern Arabic using human-based computing. This tool will help construct and dynamically develop and maintain the tool’s lexicons. They also inspected the problem of conducting sentiment analysis on Arabic text in the WWW. The solution of the problem they proposed is a lexicon-based approach.

Kisilevich et al (2013) explore sentiments in users’ comments expressed against photos uploaded on photo sharing sites. They utilised a manually built sentiment lexicon to judge photo quality and general sentiment about objects in the photos. They modelled opinions on a real valued scale rather than on a binary scale.

The authors in Paltoglou and Thelwall (2010) experimented with several term weighting measure for sentiment analysis. The measures typically were used for information retrieval tasks. Their conclusions state that measures which scale sublinearly in relation to TF are most suitable for accurate classifications.

Pang et al. (2002) showed that binary unigrams provides the best baseline classification accuracies when compared with more sophisticated models such as bigrams and adjectives.
3 Software and dataset

3.1 Rapidminer

Rapidminer (2015) is a java-based open source data mining and machine learning software. It has a graphical user interface (GUI) where the user can design his machine learning process without having to code. The process is then transformed into an XML (extensible Markup Language) file which defines the operations that the users want to apply to the given dataset. Perhaps, one of the most valuable extensions to Rapidminer is the text processing package. This includes many operators that support text mining. For example, there are operators for tokenisation, stemming, filtering stopwords, and generating n-grams. The main reason for choosing Rapidminer is that the text processing package can deal with the Arabic language.

Designing a classification task, using Rapidminer, requires the use of the process documents from files and X-validation operators. The descriptions of these operators are found in Subsection 3.1.1 and Subsection 3.1.2.

3.1.1 Process documents from files

This is a container operator, i.e. it contains other operators related to text processing. In this work, the Tokenise, Stem(Arabic, Light), Filter Stopwords(Arabic), and Generate-n-Grams(Terms) operators were used. The Tokenise operator is responsible for splitting the text of the review into tokens or words. The Stem(Arabic, Light) operator is responsible for removing common suffixes and prefixes from tokens without necessarily reducing them to their respective roots. The Filter Stopword(Arabic) removes noise Arabic words that do not affect the classification task. When dealing with sentiment analysis, the usage of this operator is tricky, because negation words are considered stopwords for topical classification and thus removed. On the other hand, negation words are critical for sentiment analysis as they can reverse the sentiment from positive to negative and vice versa. The Generate-n-grams operator can slightly alleviate this problem by generating sequences of n-words and each sequence is considered one token. N, here, specifies the number of words or terms in a sequence. In this work, n was set to 2, i.e. we generated bi-grams. Obviously there are more sophisticated methods to deal with valence shifters such as negations. For example, one could use a parser that would search for valence shifters and attach them to the proper term and determine the sentiment of the sequence as a whole. For instance, good (ََُْد) is a positive word and not-good (ََُْد ﻦَِ) is a negative word in a given context.

The process documents from files operator takes as input folders that contain text files. In this work, two folders were fed to this operator; namely; one folder which contains the positive reviews and a second folder that contains negative reviews. This operator has a set of parameters that are generally useful when dealing with text. For example, the vector creation parameter allows the user to choose a weighting scheme for the terms from TF, TFIDF and others.
3.1.2 X-validation

Validation is an important step that allows us to test the accuracy of algorithms. The most common approaches to validation are hold out method and cross validation method. In the hold out method, part of the data or reviews is held out for testing and the remaining parts are used for training the classifier. The cross validation method, by comparison, splits the data into testing and training as in the hold out method but the data is scanned several times and each division or part of the data is get to be used in the training and testing phases. To be clear, in the ten-fold cross validation method, the data is divided into ten divisions or parts; one is used for testing and nine for training in the first run. In the second run, a different part is used for testing and nine parts, including the one that was used for testing in run one, are used for training. The runs continue until each part or division is given the chance to be part of the training data and the testing data. The final accuracy is the average of the accuracies obtained in the ten runs. In the current research, we have used ten-fold cross validation.

The X-validation operator is a nested one that consists of an operator for the classifier and another operator for calculating the performance of the classifier. The set of classifiers that we have used here are SVM, K-NN, and NB.

The performance operator is responsible for calculating the accuracy of the classifier. It has many parameters that one can choose from when deciding on a method for calculating the accuracy of the classifier. In the current research, we used precision, recall and accuracy as measures of accuracy. To calculate these we need:

- TP: the number of reviews that were correctly classified by the classifier to belong to the current class
- TN: the number of reviews that were correctly classified by the classifier not to belong to the current class
- FP: the number of reviews that were mistakenly classified by the classifier to belong to the current class
- FN: the number of reviews that were mistakenly classified by the classifier not to belong to the current class.

Therefore, Precision = TP / (TP + FP), Recall = TP / (TP + FN) and Accuracy = (TP + TN) / (TP + FP + TN + FN) for binary classification tasks. As the problem we are dealing with consists of two classes, we calculated the Micro-Precision and Micro-Recall for the two classes together.

3.2 The dataset

We generated our dataset by collecting tweets and Facebook comments from the internet. These tweets and comments address general topics such as education, sports, and political news. As far as the tweets were concerned, we have utilised the crawler and annotation tool presented in Duwairi et al. (2014).

The authors in Duwairi et al. (2014) have designed a crawler to collect tweets from Twitter. They also, relied on crowdsourcing for tweets annotation. Initially, 10,000 tweets were collected and annotated. When the collected tweets were carefully examined, we realised that they suffer from several problems. They include high number of duplicate
tweets; these may be the result of re-tweeting, also some of the automatically collected tweets are empty and contain the address of the sender only. Such tweets were removed from our dataset.

We also, manually collected 500 comments from Facebook. Many of these comments were removed either because they are written in Arabizi where Roman letters are used in writing Arabic words – a style that Arab users of social media widely use; or because the comment consists of emoticons only. Table 1 shows the number of tweets and comments that remained with their sentiment orientation.

<table>
<thead>
<tr>
<th>Tweet/comment</th>
<th>Positive</th>
<th>Negative</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1,073</td>
<td>1,518</td>
<td>2,591</td>
</tr>
</tbody>
</table>

The crowdsourcing tool presented in Duwairi et al. (2014) was used to label the tweets. Volunteers have to create a username and password to use the tool. Once they log onto the system, one tweet or comment will appear at a time. The user has the option to choose a label from (1) positive (2) negative (3) neutral and (4) other.

Positive tweets are given label ‘1’, while negative tweets are given label ‘–1’. Neutral tweets are given label ‘0’. If the tweet is empty or suffers a problem then, the option ‘other’ is used and that tweet is deleted from the dataset. Every tweet must be rated by at least three different users and majority voting is used to assign the final label for every tweet. As quality assurance measure, one of the raters was one of the researchers in Duwairi et al. (2014). After labelling was complete, we store the positive tweets in one file and the negative tweets in another file. The authors of the current paper manually labelled the Facebook comments. Note that, the crowdsourcing tool is capable of identifying neutral tweets/comments. However, in this work we have focused only on the positive and negative classes only.

4 Supervised learning

4.1 Basic review

Supervised learning is a task in machine learning which aims to infer a function from labelled data. The inputs, of this task, are the objects and the outputs are the labels. This task consists of the following steps:

a Determine the type of objects/data you want to work with. In this paper, the data consist of tweets or Facebook comments. In other words, the objects are textual.

b Collect the data. This was done by using Twitter API for collecting the tweets and manually for Facebook comments.

c Annotate or label the data. This was done manually with the help of the crowds (i.e. crowdsourcing was used to label the tweets/comments).

d Determine the representation of the input objects. For textual objects, it is common to use the BOW model. BOW does not take words’ order or dependency into consideration which is unlikely accurate assumption but experimental results have shown that this model still gives excellent results and is sufficient for information
retrieval and classification. More sophisticated models have been proposed, but their associated costs outweigh their benefits. In BOW, each tweet/comment is tokenised and every token is given a weight. Several functions were used to weight the tokens or terms such as BM, TF and TFIDF.

e Determine the structure of the learned function. There is a wealth of functions or classifiers that one can use. The SVM, NB and KNN are well-known classifiers which have been used for text classification and sentiment analysis.

f Run the function on the data. The data are typically divided into training and testing. The training data are used to build the model. The testing data are used to test the resultant model or function which was built or designed using the training data. X-fold cross validation is commonly used to accomplish this task, especially with small datasets. In this paper, we have used the ten-fold cross validation.

g Calculate accuracy. Precision, recall, F1-measure, accuracy and classification error are usually used as measures to assess the goodness of the classifier.

4.2 Term weighting schemes

Assigning weights to terms or tokens is very important aspect of the BOW model. In the literature, several supervised and unsupervised weighting schemes are used. Three unsupervised weighting functions are extensively used in the literature. These are BM, TF, and TFIDF.

In the BM, binary model, the assigned weights are either equal to 0 or are equal to 1. A weight equals zero means that the current term/token is not present in the current tweet/comment. By comparison, a weight which is equal to 1 means that the token is present in the current tweet/comment one or more times.

TF, term frequency, refers to the number of times a term occurs in a tweet/comment. To avoid bias towards long comments, TF is usually normalised by dividing the raw term frequency by the maximum raw frequency of any term in the comment.

TFIDF consists of two components, namely, the term frequency and inverse document frequency. Term frequency is described above. Inverse document frequency (IDF) deemphasises the weight of terms that occur frequently in the dataset or document set and emphasises the weights of rare terms. IDF is usually calculate by \(\log(N / (1 + f(d, t)))\) where \(N\) is the total number of document/tweets/comments in the dataset. \(f(d, t)\) refers to the number of documents where term or token \(t\) appears. The addition of 1 is to avoid division by zero. The TFIDF is calculated as \(TF \times IDF\).

For classification tasks, TFIDF has proven its superiority. In the current work, we contrast these weighting schemes and investigate their suitability for sentiment analysis.

4.3 Brief description of the classifiers

4.3.1 KNN classifier

This classifier is a simple one which chooses the \(K\) number of nearest neighbours in the training documents and classifies an unannotated document based on these \(K\) neighbours. Specifically, it calculates the similarity between the unlabeled document and the remaining documents in the training dataset. After that, the labels of the most \(K\) similar documents are considered. The final label of the new document is determined using
majority voting or weighted average of the labels of these K neighbours. In Tokuhisa et al. (2008), the authors used KNN to classify emotions contained in examples, written in Japanese, extracted from the web.

4.3.2 Naïve Bayes

It is a classifier that depends on the Bayes rule written in the following formula:

\[
P(c | d) = \frac{P(c)P(d | c)}{P(d)}
\]  

The main idea of the NB classifier is to hypothesis that predictor variables are autonomous which substantially reduces the computation of probabilities. This classifier gives good results and it has been used in many research such as the work reported in Rish (2001) and McCallum and Nigam (1998).

4.3.3 Support vector machines

It is an effective traditional text categorisation framework. The main idea of SVM is to find the hyper-plane, which is represented as a vector that separates document vectors in one class from document vectors in other classes (Fung and Olvi, 2002). SVM shows very good performance and higher accuracy in many studies directed towards sentiment analysis in many languages. The work reported in Ye et al. (2009) shows that SVM did well with the English language when compared to other classifiers. Also, the work reported in Wan (2009) shows that SVM gives good results for sentiment analysis of reviews written in Chinese.

5 Experimentation and result analysis

All the experiments that were carried out in this research were done using Rapidminer (2015) which was described in Section 3.1. All the experiments used the same settings which are summarised as follows:

a  the tweets/comments where tokenised using the Tokenise operator
b  all the tokens were stemmed using Stem(Arabic, Light) operator.
c  stopwords were filtered using Filter Stopwords(Arabic)
d  stemmed tokens are grouped into bigrams using the Generate-n-Grams(Terms) operator
e  ten-fold cross validation was used to split the dataset into training and testing.

The experiments were centred on three themes:

a  accuracy versus number of neighbours for the KNN classifier and its relationship to term weighting scheme
b  accuracy versus term weighting schemes
c  accuracy versus the classifier choice.
To investigate the accuracy of the KNN classifier with different values of K and several weighting schemes, we executed this classifier several times using 1, 3, 5, 7, 9 and 11 as values for K. Table 2 shows the accuracy of KNN when TFIDF, TF and BM were used with several values for K. One can notice from Table 2 that the overall accuracy is low. The highest accuracy was equal to 50.98 when K = 1 and BM weighting scheme is used. In general, as K increases the accuracy decreases regardless of the weighting scheme. Table 2 also clarifies that when using KNN for sentiment analysis, it is enough to take into consideration the label of the closest neighbour. It is also indicates that BM is the most suitable weighting scheme for sentiment analysis using KNN.

Table 2 KNN accuracy when TFIDF, TF and BM are employed

<table>
<thead>
<tr>
<th>Value of K</th>
<th>TFIDF accuracy</th>
<th>TF accuracy</th>
<th>BM accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>49.56</td>
<td>50.56</td>
<td>50.98</td>
</tr>
<tr>
<td>3</td>
<td>43.88</td>
<td>44.81</td>
<td>44.77</td>
</tr>
<tr>
<td>5</td>
<td>42.3</td>
<td>42.65</td>
<td>42.69</td>
</tr>
<tr>
<td>7</td>
<td>41.64</td>
<td>41.84</td>
<td>41.76</td>
</tr>
<tr>
<td>9</td>
<td>41.41</td>
<td>41.57</td>
<td>41.49</td>
</tr>
<tr>
<td>11</td>
<td>41.41</td>
<td>41.41</td>
<td>41.41</td>
</tr>
</tbody>
</table>

Table 3 shows the micro-precision and micro-recall in addition to accuracy for KNN classifier when K = 1. Table 3 asserts that KNN succeeded in eliminating false examples (relatively high precision) but failed to detect all correct examples (low recall).

Table 3 Micro-precision, micro-recall, and accuracy for KNN when K = 1

<table>
<thead>
<tr>
<th>Micro-precision</th>
<th>Micro-recall</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFIDF</td>
<td>69.58</td>
<td>56.76</td>
</tr>
<tr>
<td>TF</td>
<td>69.79</td>
<td>57.59</td>
</tr>
<tr>
<td>BM</td>
<td>69.09</td>
<td>57.87</td>
</tr>
</tbody>
</table>

Table 4 presents the micro-precision, micro-recall and accuracy for the SVM classifier when TFIDF, TF, and BM were used. As it can be seen from Table 4, the highest micro-precision was equal to 75.31, the highest value of micro-recall was equal to 59.61 and the highest value of accuracy was equal to 66.15. These results are obtained when TFIDF is used. TFIDF considers the importance of a term within a document and the importance of a term in the whole collection of documents which are distributed across multiple classes. TF slightly reduces the micro-precision, micro-recall and accuracy. BM improves the performance when compared to TF but it still lacks behind the performance of TFIDF. The SVM behaviour departs from KNN behaviour when weighting scheme is considered. However, SVM and KNN have the same behaviour when detecting correct examples (recall) and eliminating false examples (precision). The SVM has high precision and low recall but the results are better than the precision and recall given by KNN.
Table 4

SVM performance against term weighting schemes

<table>
<thead>
<tr>
<th></th>
<th>Micro-precision</th>
<th>Micro-recall</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFIDF</td>
<td>75.31</td>
<td>59.61</td>
<td>66.15</td>
</tr>
<tr>
<td>TF</td>
<td>74.41</td>
<td>59.55</td>
<td>66.11</td>
</tr>
<tr>
<td>BM</td>
<td>75.16</td>
<td>58.52</td>
<td>65.3</td>
</tr>
</tbody>
</table>

Table 5 illustrates the results obtained when the NB classifier was used. The best results were achieved when TF was used. This deviates from the behaviour of SVM where the best results were achieved when TFIDF was used and the behaviour of KNN when the best results were achieved when BM was used. Tables 2, 3 and 4 show that alternating the weight scheme between TFIDF, TF and BM has low impact on the obtained accuracies for the three classifiers.

Table 5

NB performance against term weighting schemes

<table>
<thead>
<tr>
<th></th>
<th>Micro-precision</th>
<th>Micro-recall</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFIDF</td>
<td>69.96</td>
<td>70.56</td>
<td>69.78</td>
</tr>
<tr>
<td>TF</td>
<td>70.33</td>
<td>70.92</td>
<td>69.97</td>
</tr>
<tr>
<td>BM</td>
<td>70.21</td>
<td>70.79</td>
<td>69.86</td>
</tr>
</tbody>
</table>

To summarise, the results of this work have revealed that it is enough to only consider the closest neighbour when using KNN for sentiment analysis and that the BM is more suitable when compared with TF and TFIDF (Research Theme A). The results also demonstrate that the term weighting schemes are correlated with the classifier in use (Research Theme B). For KNN, BM gave the best accuracies. For SVM, TFIDF produced the highest accuracies. Finally, for the NB, the TF gave the best results. The results show that the overall accuracy is related to the classifier as well (Research Theme C). The current work reveals that the best accuracy was achieved when NB is used.

These findings are not generalisable for every sentiment analysis task as there are other parameters and aspects one should consider. For example, here, only bi-grams were used. There is wealth of choices for n-gram models. In addition, light stemming was used. For Arabic, there is root extraction as an alternative to light stemming. Finally, one should not ignore the role of the dataset itself.

6 Conclusions and future work

This work has considered sentiment analysis in Arabic text. A dataset, which consists of 2591 tweets/comments, was collected and labelled using crowdsourcing. The NB, SVM and KNN classifiers were used to detect the polarity of a given review. Ten-fold cross validation was used to split the data into training and testing sets. All the tweets/comments were preprocessed in the following manner: tokens were light-stemmed, stopwords were removed, bi-grams were generated. TFIDF, TF, and BM were used as weighting schemes for the tokens. The highest accuracy was equal to 69.97 was achieved by the NB classifier when TF was used. The best micro-recall was achieved
by the NB classifier when TF was employed. The best micro-precision was achieved by SVM when TFIDF was used.

The results of our extensive experimentation show that the KNN, SVM and NB classifiers did better jobs in removing false examples from the displayed results (high precision). However, the same classifiers were not able to bring all correct examples into the displayed result (low recall). The experimentation also reveals that alternating the term weighting schemes between TFIDF, TF and BM slightly affects precision, recall and accuracy. The results given by the three classifiers were not consistent when different weighting schemes were used. KNN produced the best results when BM is used. SVM gave the best results when TFIDF is used. Finally, NB gave the best results when TF was used.

Certainly there are many ways that this work can and will be improved. Firstly, the size of the dataset is rather small and if we want to make solid conclusions then we definitely need big datasets. Secondly, crowdsourcing is a useful tool when labelling or annotating large amounts of data is considered. In this work we utilised crowdsourcing to label our dataset. Thirdly semi-supervised learning could be used to sentiment analysis in Arabic text as this techniques has been applied successfully to other languages as it is described in the research reported in Rao and Ravichandran (2009), Dasgupta and Ng (2009), Sindhwani and Melville (2008) and Goldberg and Zhu (2006).

References
A framework for Arabic sentiment analysis using supervised classification


