GENDER AND AUTHORSHIP CATEGORISATION OF ARABIC TEXT FROM TWITTER USING PPM

Mohammed Altamimi[1] and William J. Teahan[2]

[1]Department of Computer Sciences And Engineering, University of Hail, Saudi Arabia
[2]Department of Computer Science, University of Bangor, Bangor, UK

ABSTRACT

*In this paper we present gender and authorship categorisationusing the Prediction by Partial Matching (PPM) compression scheme for text from Twitter written in Arabic. The PPMD variant of the compression scheme with different orders was used to perform the categorisation. We also applied different machine learning algorithms such as Multinational Naïve Bayes (MNB), K-Nearest Neighbours (KNN), and an implementation of Support Vector Machine (*LIBSVM*), applying the same processing steps for all the algorithms. PPMD shows significantly better accuracy in comparison to all the other machine learning algorithms, with order 11 PPMD working best, achieving 90 % and 96% accuracy for gender and authorship respectively.*

KEYWORDS

Arabic text categorisation, Data compression, Machine learning Algorithms.

1.INTRODUCTION

Text categorisation is the process of assigning documents to predefined categories. Recently, text categorisation has become popular due to the rapid growth of data available in the Web over the past two decades. As a result,there is large amounts of data available to be categorized in many different languages. However, text categorisation specificallyfor data obtained from Twitter in the form of tweets is considered challenging for many reasons: tweets are written in only 140 characters making it more difficult to be classified; text is written informally; and the frequency of misspelling and slang in twitter text is high[1].

The traditional (word-based) machine learning algorithms requires extensive prepossessing steps when applied to text categorisation such as stemming, tokenization, removal of stop words, and the calculation of word frequenciesto help build word vector lists. In addition, feature selection is applied subsequently to determine the most important features in a text [2]. There are number of drawbacks when applying feature selection to a text: first, the issue of defining a feature; second, the need to consider word order and context; and third, the question of weather to discard digits and non alphabetic symbols such as hashtags, underscores, and commas [3]. An alternative approach to the feature-based method is to adopt an information theoretic approach instead and to apply a character-based compression scheme such as the prediction by partial matching (PPM). By using a compression-based scheme, we sidestep issues associated with the word-based approach by adoptinga character-based approach. We also avoid the aforementioned pre-processing steps such as feature selection, stemming, and tokenization [4].

Prediction by Partial Matching (PPM) is a text compression scheme that was originally invented by Cleary and Witten in 1984 [5]. It encodes single characters one after the other using the estimated probability of the upcoming character. The basic idea of PPM is to use the last few

DOI:10.5121/ijcsit.2017.9212

characters in the input stream to predict the following one similar to a Markov based language model. Other compression techniques such as Ziv-Lempel are used for their faster speed processing rather then the effectiveness of compression[6].

In our research work, we are interested in applying compression-based language models such as PPM to problems in Natural Language Processing. The specific focus of this paper is the application of PPM to text categorisation of Arabic text. Text Categorisation in Arabic is considered difficult compared to other languages. Arabic characters have many shapes depending on the positions of the letter: isolated, initial, middle, and end. Moreover, some characters have the same shapes and can be distinguished only using the dots, zigzag, and diacritics [7].

In this paper, we present Arabic text categorisation of tweets belonging to per-selected users in Twitter (specifically for this paper gender and authorship categorisation). Our goal in gender categorisation is to classify tweets as being written by either a male or female. On the other hand, our goal on authorship categorisation is to identify the writer of the tweets depending on sample training sets having been provided for each author. Our contributions include running the same data over various machine learning algorithms along with PPM for comparison purposes.

The rest of the paper is organized as follow, section twoprovidesthe background and related work, section threedescribesthe data collection process, section fourdiscussour methodology, section fivelistsour experimental results, and section sixprovidesthe conclusion and future work.

2.BACKGROUND AND RELATED WORK

Text categorisation is used in many fields such as machine learning, text mining, natural language processing, and information retrieval. Text categorisation has many applications such as language identification [7] [9], dialect identification [10][11][12], spam filtering [13][14], and sentiment classification [15]. However, much of the above work has focused on the categorisation of standard text rather than non-standard text such as tweets. Recently, Twitter categorisation has attracted attention with researchfocussed on, for example, sentiment classification [16][17][18][19], and authorship identification [20][21].

Many researchers have already explored Arabic text categorisation using machine learning algorithms. Duwairi[22]has compared three different classifiers for Arabic text categorisation. The results show that Naïve Bayes (NB) outperforms both the K-Nearest Neighbors(KNN) and distance-based classifiers. A study by Alsaleem[23] compared Support Vector Machine (SVM) with NB, and showed that SVM outperform NB algorithm in terms of recall, precision, and the F1-measure. On the other hand, categorisation using compression based methods has been explored less than the machine learning approach. In [2], a comparison among three different compression techniques (RAR, gzip, LZW) was undertaken for Arabic text and it was found that RAR always produced more accurate classification then LZW and gzip. The RAR implementation combines both LZ and PPM compression.

Research into processing Arabic twitter text was aided when Twitter started to support Arabic hashtags. The study by Bekkali and Lachkar[24] applied twitter text categorisation based on applying rough set theory using the NB and SVM classifiers. The study shows that applying the upper approximation of rough set theory increases the F1-measure.Hussienet al.[25]applied sentiment classification to twitter text. The study showed how automatic labelling using SVM and Multinomial Naïve Bayes (MNB) produces better results than human labelling. The research by Alabdullatif et al.[26]used MNB to classify topics such as sport, religion, economy, politics, and technology and resulted in an accuracy of 90%.

Many Arabic researchers have also studied authorship identification. Alwajeeh et al. [27]explored authorship identification using both the SVM and NB classifiers, withboth classifiers preformingwellin terms of accuracy. The research by Altheneyan and Menai[28] provided an extensive study in authorship identification using NB classifiermodels: Multinational Naïve Bayes; Multivariate Bernoulli Naïve Bayes; and Multivariate Poisson Naïve Bayes. MBNB performed well among all the NB models with an average accuracy of 97%. Recently,Albadarneh et al.[29]conducted author identification in twitter involving big data analysis. They focused on many challenges such as dealing with the large-scale of Arabic tweets, the short text length of tweets, and the lack of available Arabic NLP tools. The results produced an accuracy of 61.6% using a NB classifier to perform the experiment using the Hadoop platform for the big data analysis.

However, gender identification is a less studied area compared to authorship identification for Arabic. The study by Alsmearat et al. [30] of gender identification of Arabic articles has been applied using the NB and SVM classifiers. More recently, Alsmearat et al. [31]investigated the impact of emotion analysis on finding the author gender.Their work included applying bag-of-words computing features related to sentiment. The goal was to find out if there is a specific distinct style of writing between males and females. But they could not confirm this assumption based on concrete evidence.

3.DATA COLLECTIONPROCESS

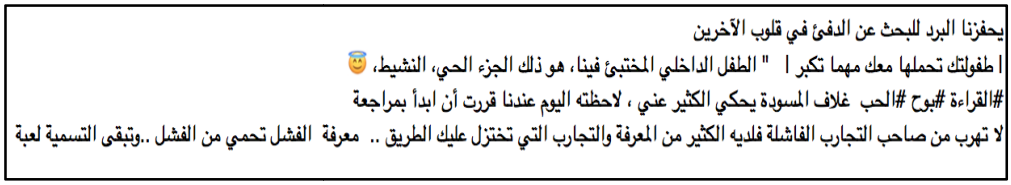
We are in the process of collecting data to create a text categorisation corpus designed specifically for Arabic text, in order to aid research in this area. This will also help with evaluating the effectiveness of the text compression-based approach we are interested in investigating. We started collecting over 200,000 tweets from multiple usernames using theTweepy library[32]. Tweepy is a Python package that interfaces to the Twitter API for collecting data. 101 usernames were selected based on their gender, genre, and country. In order to create the training data set, we collected from each username the top 2000 tweets on March 5th 2017. The tweets have gone through several processing steps:

- Break lines were removed from all the tweets. This ensures that all tweets are placed on one line.
- Retweeted tweets were removed. This is to confirm that the tweets are collected for a specific username, and are not written by another person.
- HTTP links, usernames and non Arabic tweets are removed.
- Extra lines generated from the retweet extraction are also removed.
- Special characters such as hashtags, underscores, quotes, emojis, and stop words are kept. We wish to keep this so that they can aid identification when we perform the experiments.

Table 1. Data Collected from Twitter using the Tweepy library.

File	Before Processing		After Processing	
	Number of tweets	Number of words	Number of tweets	Number of words
Training	200917	3410134	118532	1952239
Testing	2970	55181	1816	29446

Figure 1. Sample of the data after processing [33].



Our test data sets were collected in March 25, 2017. This was three weeks after collecting our training sets from the same usernames. Testing sets were processed the same way that we processed the training set. However, this time only the top 30 tweets were collected for our testing sets. This was to ensure there was no overlap between the training and testing sets. The number of tweets and number of words that were collected for the training and testing data before and after processing is shown in Table 1. A sample of the data is also shown in Figure 1.

Our aim in the future is to collect multiple test sets over different periods of time. We wish todesignate specific training testing splitsrather than use a cross-fold validation process. This is because we believe designating specific splits is more representative of the categorisation task in this case because of the dynamic streaming nature of twitter data which changes over time. We also wish to be able to directly compare future experimental results avoiding possible inconsistent processing of the data by explicitly designating the specific training and testing splits we are using in order to aid future research. In the future, we will also investigate other categorisation tasks such as topic, dialect, genre and style categorisation and other important issues that arise with Arabic text such as code-switching. However, the specific focus for this paper, as stated, will be on gender and authorship categorisation.

4.METHODOLOGY

Prediction by partial matching (PPM) is a lossless text compression technique based on the adaptive modelling context family. PPM uses a fixed number of previous characters based on a selected maximum fixed order to predict the upcoming character. For instance, if the selected maximum order is five, the prediction of the next character will be based on the previous five characters. PPM moves from the maximum highest order down to lower orders using the escape mechanism. This process will be continued until the lowest default order of -1 is reached, where all character probability are equiprobable[34].

PPM has gone through many improvements with variants such as PPMA and PPMB [5], PPMC [35], PPMD[36], PPM* [37] PPMO [38]. For PPMC, the probability P_{PPMC} for the next character$c_i$is given by:

$$P_{PPMC}(\varphi) = \frac{c_d(\varphi)}{T_d}$$

where the currently used coding order is specified by d, the total amount of times that the current context $c_{i-5} \cdots c_{i-1}$ has occurred is indicated by $T_d(c_{i-5} \cdots c_{i-1})c_d(c_i|c_{i-5} \cdots c_{i-1})$represents the total number of occurrences for the symbol c_i in the current context. The estimation of the escape probability E by PPMC is as follows:

$$E_{PPMC} = \frac{t_d}{T_d}$$

where the total number of times that a unique character has occurred following the current context is represented by t_d.

PPMD is a slight variation of PPMC invented by Howard [36] which often results in improved compression. The formula for estimating the probability P for the next character$c_i$is given by:

$$P_{PPMD}(\varphi) = \frac{2c_d(\varphi)-1}{2T_d} \qquad (1)$$

and the escape probability is estimated as follows:

$$E_{PPMD} = \frac{t_d}{2T_d}. \qquad (2)$$

Table 2 shows an example of how the PPMC processworks which has become a benchmark version of PPM [39].

Table 2. PPMC after processing the string "المســـلم"[34].

Order k=2			**Order k=1**			**Order k=0**			**Order k=-1**		
Prediction	c	p	Prediction	c	p	Prediction	c	p	Prediction	c	p
م→ال	1	$\frac{1}{2}$	ل→ا	1	$\frac{1}{2}$	→ا	1	$\frac{1}{10}$	$\rightarrow A$	1	$\frac{1}{\|A\|}$
→Esc	1	$\frac{1}{2}$	→Esc	1	$\frac{1}{2}$	→ل	2	$\frac{2}{10}$			
						→م	2	$\frac{2}{10}$			
س→لم	1	$\frac{1}{2}$	م→ل	2	$\frac{2}{3}$	→س	1	$\frac{1}{10}$			
→Esc	1	$\frac{1}{2}$	→Esc	1	$\frac{1}{3}$	→Esc	4	$\frac{4}{10}$			
ل→مس	1	$\frac{1}{2}$	س→م	1	$\frac{1}{2}$						
→Esc	1	$\frac{1}{2}$	→Esc	1	$\frac{1}{2}$						
م→سل	1	$\frac{1}{2}$	ل→س	1	$\frac{1}{2}$						
→Esc	1	$\frac{1}{2}$	→Esc	1	$\frac{1}{2}$						

Let us imagine three scenarios where we predict three letters "ا","ل"and "ج" after we have already seen "المســـلم" (see Table 3). First, if we want to predict "ل" following "المســـلم", if we use a maximum order of 2 in this case (for illustration purposes), the probability is estimated as $\frac{1}{2}$(see the order 2 context س□→ل in Table 2). This requires 1 bit to encode ($-\log_2\left(\frac{1}{2}\right) = 1$bit).(PPM normally uses arithmetic coding to physically encode the probabilities which results in the code length being close to the theoretical optimum which is $-\log_2 p$ where p is the probability being encoded. However, when using PPM for text categorisation purposes, there is no need to physically encode the probabilities and instead we can compute the theoretical code lengths directly and use that as the categorisation measure.)

However, let us imagine instead the letter "ا" follows "المســـلم"; the escape probability of $\frac{1}{2}$, will be encoded from order 2 because the letter "ا" was not seen in that context. Then the process will move down to order 1, the escape probability will be $\frac{1}{2}$again because the letter "ا" was also not seen in order 1. Finally, the encoded probability will be $\frac{1}{10}$since the letter "ا" was found in the order 0 context. The total probability to encode the letter "ا" is $\frac{1}{2}\times\frac{1}{2}\times\frac{1}{10}=\frac{1}{40}$, which requires only 5.32 bits to encode (see Table 3).

If we want to predict a letter that have not been seen in previous orders such as "ج", then the escape probability will be $\frac{1}{2}$ for order 2, $\frac{1}{2}$for order 1, $\frac{4}{10}$ for order 0 and finally the letter "ج" will be found in order -1. The probability of correct prediction for the letter "ج" will be $\frac{1}{A}$ where A is the alphabet size (256) for a standard byte-based encoding (8 bits). The total probability to encode the letter "ج" is $\frac{1}{2}\times\frac{1}{2}\times\frac{4}{10}\times\frac{1}{256}=\frac{1}{10240}$, which requires 13.32 bits to encode the character.

Table 3. Encoding three sample characters using PPMC.

Character	Probabilities encoded	Code lengthbeing used
ل	$\frac{1}{2}$	$-\log_2\left(\frac{1}{2}\right) = 1$bit
ا	$\frac{1}{2}, \frac{1}{2}, \frac{1}{10}$	$-\log_2\left(\frac{1}{2}\times\frac{1}{2}\times\frac{1}{10}\right) = 5.32$bits
ج	$\frac{1}{2}, \frac{1}{2}, \frac{4}{10}, \frac{1}{A}$	$-\log_2\left(\frac{1}{2}\times\frac{1}{2}\times\frac{4}{10}\times\frac{1}{256}\right) = 13.32$bits

PPM is used for classifying in the following manner – simply select the class associated with the model that compresses the text best. The main idea is to predicate the correct author/gender of text *T* using the formula:

$$\hat{\theta}(\mathrm{T}) = \mathrm{argmin}_c H(T|S_c)$$

where$H(T|S)$is some approximation of relative entropy of text *T* with respect to text *S* and the class *C* ischosen from the model with the minimum value. In this case, it is estimated using the PPM compression scheme i.e. for an order 5 model, it is calculatedusing the following formula:

$$H(T|S) = -\sum_{i=1}^{n} \log_2 P(c_i|c_{i-5}\cdots c_{i-1})$$

where*n* is the length of the text and the probabilities for each character are calculated using the PPM Markov-based modelling method which estimates the probability of the next character (see formulas (1) and (2) for PPMD) based on the context of the previous five characters.
Normally PPM is an online method with its model being dynamically updated as the text is processed sequentially. An alternative static variation is to prime the model using some representative training text, and then compression of the testing text proceeds without updating the model further. This static variation is very effective when PPM is applied to text categorization as it substantially reduces the amount of processing time needed to categorize using multiple PPM models.

5.EXPERIMENTAL RESULTS

In order to see how effective our PPM-based method was at categorising Arabic tweets, we applied our experiments to the Twitter data we collected that was described in section 3. We excluded two persons from our training sets because we found that most of their testing sets wereretweeted. Our data was annotated with manually assigned labels for both training and testing sets. We assigned one of two labels – male or female – for the gender experiments based on the person's name, whereas for authorship we assigned 99 labels using the author's username. We used PPMD(as opposed to PPMC) because it hasshown good results previously [4],which was implemented using the Text Mining Toolkit (TMT) [40].Static PPMD models were created by training on each class of text. On the other hand, Weka[41] was used to apply the other machine learning algorithms used in the experiment. We selected well-performed methods suchas Multinational Naïve Bayes, K Nearest Neighbors, and LIBSVM (an implementation of Support Vector Machine).

For the gender categorisation, we investigated using different orders ofPPMDfrom order 3 to order 12. We found that the accuracy, recall and precision increased up to order 11 but then decreased subsequently as shown in Table 4. (The best result is shown in bold font). Previous

research had shown a similar effect for other languages [4] but not at such a high order. For the authorship experiment, we also found order 11 to be effective but there was also a number of other orders with similar results (see Table 5). The accuracy results from both Tables have also been graphed in Figure 2.

Table 4. Gender categorisation of Arabic tweets using PPMD.

Orders	Order 3	Order 4	Order 5	Order 6	Order 7	Order 8	Order 9	Order 10	**Order 11**	Order 12
Accuracy	0.778	0.808	0.828	0.828	0.879	0.889	0.889	0.889	**0.909**	0.899
Recall	0.767	0.801	0.822	0.818	0.869	0.868	0.868	0.863	**0.889**	0.881
Precision	0.767	0.799	0.820	0.812	0.876	0.902	0.902	0.911	**0.925**	0.909

In order to classify text documents for machine learning algorithms using Weka, training and testing sets need to run through a string-to-word-vector filter. We builtour filter using the common term frequency-inverse document frequency (tf-idf) measure. For gender categorisation we chose the top 1000 words to appear in the filter from each category (female/male). We did not do any further prepossessing to the data such as stemming, tokenization, andremoval of stop word, as we wanted to mimic the same approach as for the PPM based experiments,except for normalizing all data, as we noticed it produced better results. We applied three different algorithms –MNB, KNN, and LibSVM. We found that LibSVM outperformed the other machine learning algorithms as shown in Table6 which mirrored similar results found by previous researchers.For the authorship experiment, we applied the same processing steps to build our text models in Weka. In this case, however, we only chose the top 100 frequent words to appear in our vector list. Again, LibSVMachieved greater accuracy compared tothe other machine learning algorithms we experimented with as shown in Table 7.

Table 5. Authorship categorisationof Arabic tweets using PPMD.

Orders	Order 3	**Order 4**	Order 5	Order 6	Order 7	Order 8
Accuracy	0.939	**0.960**	0.949	0.939	0.929	0.949
Orders	**Order 9**	Order 10	**Order 11**	**Order 12**	**Order 13**	Order 14
Accuracy	**0.960**	0.949	**0.960**	**0.960**	**0.960**	0.949

Table 6. Experimental results for gender categorisation of Arabic tweets.

Measures	MNB	libSVM	KNN 1	**PPMD Order 11**
Accuracy	0.747	0.797	0.414	**0.909**
Recall	0.750	0.798	0.414	**0.889**
Precision	0.760	0.797	0.764	**0.925**

Table 7. Experimental results for authorship categorisation of Arabic tweets.

Measures	MNB	libSVM	KNN 1	**PPMD Order 11**
Accuracy	0.929	0.939	0.444	**0.960**
Recall	0.929	0.939	0.444	**0.960**
Precision	0.899	0.909	0.386	**0.939**

From both the gender categorisation and authorship categorisation experimental results in Tables 6 and 7, we see that the PPM categorisation method significantly outperforms the other machine learning methods in terms of accuracy, recall and precision.

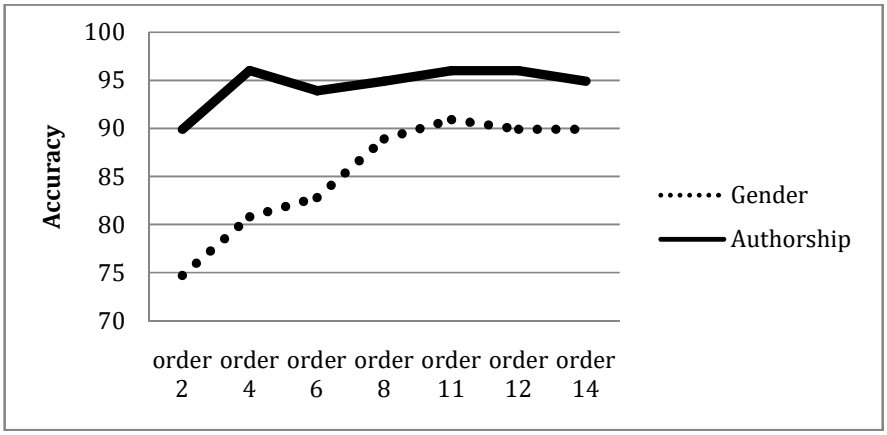
6.CONCLUSION

In this paper, we have shown how the Prediction by Partial Matching (PPM) text compression scheme is very effective when it is used for categorisation of Arabic twitter text. We have argued that character-based compression models have a number of benefits over the word-based machine learning approaches. For instance, problems such as segmentation, stemming, tokenisation, and feature extraction can be all avoided by using character-based models.

We collected Arabic text from Twitter to create training and testing sets for our experiments. We found thatthe PPMD model with order 11 was the most effective achieving 90% and 96% accuracy for gender and authorship categorisation respectively. However, we noticed that increasing the order to higher orders would show no effect at all on the accuracy (see Figure 2).

Our aim was to investigatethe categorisation of twitter text (tweets). There are many issues when dealing with tweets compared to formal text, as they are often not grammatically well-structured, with a variety of slang and colloquial language being used, and there is also the limitation of only 140 characters being allowed in each single tweet. Our future work will continue to run a variety of experiments using test setscollected over different periods of time to investigate how well PPM performs at other categorisation tasks involving topic, genre, style and dialect and also determine the effect of code-switching a phenomenon that is common in Arabic text.

Figure 2. Accuracy of Arabic text categorisation using different PPMD orders.



ACKNOWLEDGMENT

The Saudi Arabian government supports this work and the author is grateful for their scholarship and support.

REFERENCES

[1] Ö. Çoban, B. Özyer, and G. T. Özyer, "A Comparison of Similarity Metrics for Sentiment Analysis on Turkish Twitter Feeds," in Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on, 2015, pp. 333–338.

[2] H. Ta'amneh, E. A. Keshek, M. B. Issa, M. Al-Ayyoub, and Y. Jararweh, "Compression-based Arabic text classification," in Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on, 2014, pp. 594–600.

[3] E. Frank, C. Chui, and I. H. Witten, "Text categorization using compression models,"Waikato University, 2000.

[4] W. J. Teahan and D. J. Harper, "Using compression-based language models for text categorization," in Language modeling for information retrieval, Springer, 2003, pp. 141–165.

[5] J. Cleary and I. Witten, "Data compression using adaptive coding and partial string matching," IEEE Trans. Commun., vol. 32, no. 4, pp. 396–402, 1984.

[6] T. Bell, I. H. Witten, and J. G. Cleary, "Modeling for text compression," ACM Computing Surveys, vol. 21, no. 4, pp. 557–591, 1989.

[7] M. A. Alghamdi, I. S. Alkhazi, and W. J. Teahan, "Arabic OCR evaluation tool," in Computer Science and Information Technology (CSIT), 2016 7th International Conference on, 2016, pp. 1–6.

[8] A. S. House and E. P. Neuburg, "Toward automatic identification of the language of an utterance. I. Preliminary methodological considerations," J. Acoust. Soc. Am., vol. 62, no. 3, pp. 708–713, 1977.

[9]W. B. Cavnar, J. M. Trenkle, and A. A. Mi, "N-Gram-Based Text Categorization," Ann Arbor MI 48113.2, pp. 161–175, 1994.

[10] J. Nerbonne, W. Heeringa, and P. Kleiweg, "Comparison and classification of dialects," in Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics, 1999, pp. 281–282.

[11] D. P. Branner, Problems in comparative Chinese dialectology: the classification of Miin and Hakka, vol. 123. Walter de Gruyter, 2000.

[12] O. F. Zaidan and C. Callison-Burch, "Arabic dialect identification," Computational Linguistics., vol. 40, no. 1, pp. 171–202, 2014.

[13] E. P. Sanz, J. M. G. Hidalgo, and J. C. C. Pérez, "Email spam filtering," Advances Computers, vol. 74, pp. 45–114, 2008.

[14] A. Bratko, G. V Cormack, B. Filipič, T. R. Lynam, and B. Zupan, "Spam filtering using statistical data compression models," J. Mach. Learn. Res., vol. 7, no. Dec, pp. 2673–2698, 2006.

[15] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, 2002, pp. 79–86.

[16] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in Proceedings of the workshop on languages in social media, 2011, pp. 30–38.

[17] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," CS224N Proj. Report, Stanford, vol. 1, p. 12, 2009.

[18] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach," in Proceedings of the 20th ACM international conference on Information and knowledge management, 2011, pp. 1031–1040.

[19] A. M. Qamar, S. A. Alsuhibany, and S. S. Ahmed, "Sentiment Classification of Twitter Data Belonging to Saudi Arabian Telecommunication Companies," Int. J. Adv. Comput. Sci. Appl., vol. 1, no. 8, pp. 395–401, 2017.

[20] A. Castro and B. Lindauer, "Author Identification on Twitter." 2012.

[21] R. Layton, P. Watters, and R. Dazeley, "Authorship attribution for twitter in 140 characters or less," in Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second, 2010, pp. 1–8.

[22] R. M. Duwairi, "Machine learning for Arabic text categorization," J. Am. Soc. Inf. Sci. Technol., vol. 57, no. 8, pp. 1005–1010, 2006.

[23] S. Alsaleem, "Automated Arabic Text Categorization Using SVM and NB," vol. 2, no. 2, pp. 124–128, 2011.
[24] M. Bekkali and A. Lachkar, "ARABIC TWEETS CATEGORIZATION BASED ON ROUGH SET THEORY," Int. J. Comput. Sci. Inf. Technol., vol. 6, 2014.
[25] W. A. Hussien, Y. M. Tashtoush, M. Al-Ayyoub, and M. N. Al-Kabi, "Are emoticons good enough to train emotion classifiers of arabic tweets?," in Computer Science and Information Technology (CSIT), 2016 7th International Conference on, 2016, pp. 1–6.
[26] A. Alabdullatif, B. Shahzad, and E. Alwagait, "Classification of Arabic Twitter Users: A Study Based on User Behaviour and Interests," Mob. Inf. Syst., vol. 2016, 2016.
[27] A. Alwajeeh, M. Al-Ayyoub, and I. Hmeidi, "On authorship authentication of arabic articles," in Information and Communication Systems (ICICS), 2014 5th International Conference on, 2014, pp. 1–6.
[28] A. S. Altheneyan and M. E. B. Menai, "Naïve Bayes classifiers for authorship attribution of Arabic texts," J. King Saud Univ. Inf. Sci., vol. 26, no. 4, pp. 473–484, 2014.
[29] J. Albadarneh, B. Talafha, M. Al-Ayyoub, B. Zaqaibeh, M. Al-Smadi, Y. Jararweh, and E. Benkhelifa, "Using big data analytics for authorship authentication of arabic tweets," in Utility and Cloud Computing (UCC), 2015 IEEE/ACM 8th International Conference on, 2015, pp. 448–452.
[30] K. Alsmearat, M. Al-Ayyoub, and R. Al-Shalabi, "An extensive study of the bag-of-words approach for gender identification of arabic articles," in Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on, 2014, pp. 601–608.
[31] K. Alsmearat, M. Shehab, M. Al-Ayyoub, R. Al-Shalabi, and G. Kanaan, "Emotion analysis of arabic articles and its impact on identifying the author's gender," in Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of, 2015, pp. 1–6.
[32] Tweepy, "Tweepy," Tweepy.org. [Online]. Available: Tweepy.org. [Accessed: 05-Mar-2016].
[33] Majeed Timraz, "kotobji," Twitter, 2012. [Online]. Available: https://twitter.com/majeedtimraz0. [Accessed: 07-Apr-2017].
[34] K. M. Alhwaiti, "Adaptive Models of Arabic Text," Bangor University, 2014.
[35] A. Moffat, "Implementing the PPM data compression scheme," IEEE Trans. Commun., vol. 38, no. 11, pp. 1917–1921, 1990.
[36] P. G. Howard, "The Design and Analysis of E cient Lossless Data Compression Systems." Diss. PhD thesis, Brown University, 1993.
[37] J. G. Cleary and W. J. Teahan, "Unbounded length contexts for PPM," Comput. J., vol. 40, no. 2 and 3, pp. 67–75, 1997.
[38] P. Wu and W. J. Teahan, "A new PPM variant for Chinese text compression," Nat. Lang. Eng., vol. 14, no. 3, pp. 417–430, 2008.
[39] W. J. Teahan, "Adaptive Models of English Text," Waikato University, 1998.
[40] W. J. Teahan and D. J. Harper, "Combining PPM models using a text mining approach," in Data Compression Conference, 2001. Proceedings. DCC 2001., 2001, pp. 153–162.
[41] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," ACM SIGKDD Explor. Newsl., vol. 11, no. 1, pp. 10–18, 2009.