IMPROVED MICRO-BLOG CLASSIFICATION FOR DETECTING ABUSIVE ARABIC TWITTER ACCOUNTS

Ehab A. Abozinadah[1] and James H. Jones, Jr.[2]

[1]Department of Information System, King Abdul-Aziz University, Jeddah, Saudi Arabia
[2]Department of Electrical and Computer Engineering, George Mason University, Fairfax, Virginia, USA

ABSTRACT

The increased use of social media in Arab regions has attracted spammers seeking new victims. Spammers use accounts on Twitter to distribute adult content in Arabic-language tweets, yet this content is prohibited in these countries due to Arabic cultural norms. These spammers succeed in sending targeted spam by exploiting vulnerabilities in content-filtering and internet censorship systems, primarily by using misspelled words to bypass content filters. In this paper we propose an Arabic word correction method to address this vulnerability. Using our approach, we achieve a predictive accuracy of 96.5% for detecting abusive accounts with Arabic tweets.

KEYWORDS

Arabic, Twitter, Cyber Crime, NLP, Spelling Correction, Domain Specific Lexicon, Slang, Arabic Dialects, Text Classification, Big Data.

1. INTRODUCTION

Twitter is a micro blogging social media platform where users post short messages called *tweets*. Each tweet is limited to 140 characters[1] and can comprise text, links, symbols, videos, and, or pictures. Users in Twitter may have both a *following* and *followers,* thereby forming a social network. Followings are users that a Twitter user subscribes to, while followers are users who subscribe to an account. A Twitter account gets more influence in a social network whenever it has more followers. Each username starts with the symbol @, and occurrences of a username in a tweet are called *mentions*. A mention may include part of the original tweet, a reply, or may simply acknowledge a username. A tweet's topics start with symbol #, which it is called a *hashtag*. A hashtag can be part of the tweet and can be searched through Twitter's search engine.

Arabic is a complex morphological, syntactical, and semantical language which varies in different regions of the Middle East [1]. Arabic text is read from right to left, and the words are separated by a whitespace unlike Farsi language [2]. Arabic language does not have capitalization; however, diacritization is used on the top or the bottom of the words to emphasize their pronunciation. Arabic has two forms: formal Arabic, also called Modern Standard Arabic (MSA), and informal Arabic. Formal Arabic is used in books, newspapers, academia, and other forms of formal literature, while informal Arabic comprises local words and slang words within different regions of the Arabic-speaking world.

DOI: 10.5121/ijdkp.2016.6602

Arabic is one of the top ten languages used in Twitter, and 6% of the daily tweets are in the Arabic language [3]. Since the Arab Spring revolution in 2011, the number of Twitter users from the Middle East has increased to over five million active users. Consequently, these users post over 500 million tweets per year, which is about 17 million tweets per day [4]–[7].

There is limited research on informal Arabic. Some of the limitations facing the automated classification of Arabic tweets are:

1- Use of slang words which differ among regions in the Middle East, and the lack of a dictionary that defines meanings of words for these dialects.
2- Lack of capitalization for words to express emotions like anger or attention. Instead of capitalization, Arabic uses word elongations (repeated sequences of letters). This elongation leads to misclassification of words.
3- Deliberate misspelling of words is used to reduce the number of the letters in a Tweet in order to meet the Twitter limitation of 140 characters. Such word compressions can mislead spelling correction and word normalization tools.
4- Arabic diacritization can lead to misspelled words. Twitter users often disregard adding discretization to the word but instead use letters in the word to reflect the pronunciation sound of the diacritization.
5- Using symbols to reflect letters in the words leads to misspellings and confuses attempts to automatically determine word meaning.

Spammers are exploiting these limitations in Arabic lexical tools to bypass content filtering and internet censorship systems and send targeted Twitter spam to Arabic-speaking users. There are different kinds of spammers who target social networks, such as those who post abusive content, profanity, swear words, insulting words, harassment, child pornography, and child exploitation. Collectively, we call the accounts of such spammers *abusive accounts.*

In this work, we make three main contributions:

- We have proposed a method that uses Arabic tweet word correction to improve the detection of abusive accounts on Twitter.
- We have evaluated the results of the proposed Arabic word correction method with other well-known word correction methods using more than 800 accounts and more than one million tweets.
- We have used word stemming in each method to assess the performance of each method with and without stemming.
-

The rest of the paper is organized as follows: in Section 2 we discuss prior and related work; in Section 3 we explain the proposed method of Arabic tweet normalizing; in Section 4 we describe the dataset; in Section 5 we outline the experimental setting and results obtained; in Section 6 we present our conclusions and discuss future work.

2. PRIOR AND RELATED WORK

This research focuses on finding the best method to normalize Arabic tweets in order to improve automated detection of abusive accounts. Prior and related work falls into two categories: Arabic normalization and spam detection in social media.

2.1 Arabic Normalization

In [8], Duwairi et al uses more than 25,000 labeled tweets to classify the tweet type. The tweets were normalized using three domain specific lexicon dictionaries that translate the informal words to MSA. This research shows the benefit of understanding the informal words in the tweets, and normalizing these informal words with Latin letters. Interestingly, this work finds that stemming the tweets would weaken the classification accuracy.

In [9], Sallam et al compares the results of using three datasets of MSA, namely: non-normalized and non-stemmed, only normalized, and only stemmed. The "only normalized" data set has the best result as it outperforms the other two data sets. Hence, normalization has higher impact on the result than the stemmer.

Shaalan et al in [10] developed an automatic spell checker for standard Arabic and Egyptian dialects. They created different lists of common Arabic spelling errors to choose corrected words from an Arabic dictionary. The first dictionary was the Reading Errors list which contained a group of letters that looked similar to each other. The second was the Hearing Errors list that contained a group of letters with similar pronunciation. The third was the Touch Typing Errors list that contained a group of letters close to each other on the keyboard. The fourth was the Morphological Errors list that contained a list of common words based on Arabic morphology. The fifth was the Editing Errors list that deal with typing mistakes such as insertion, deletion, and substitutions. This approach corrects the word based on detected error type, which may result in incorrect word correction for different dialects.

In [1], Muaidi et al divide each word into bigrams to develop an Arabic spell-checker. Each bigram is given a score. There are scores for the end of the word, anywhere in the word, or not in the word which are assigned values of 2, 1, or 0 respectively. Each word is compared against a list of words with similar bigrams. A word is considered correct if it has score of one for all the bigrams in the beginning and middle of the word, and a score of two for the last bigram, otherwise the word will be considered wrong, hence a score of zero.

In [11], Shaalan et al use a dictionary that contains more than 9 million words for Arabic spell correction. A word is considered misspelled if it is not part of the dictionary list, then the Edit Distance algorithm is applied to get a list of candidate words. Each candidate word is scored based on a noisy channel model that uses a one-million-word corpus, then the word with the highest score is chosen. This approach would apply to the MSA correction corpus, but it would not cover the informal Arabic word corpus.

2.2 Spammers in Social Media

Countries in the Middle East restrict and regulate the use of social media by the public and government employees to reduce the incidents of exploitation from spammers[12]. Spammers create and stockpile social media accounts, especially on Twitter because of its simplicity to create new accounts due to weak account opening and verification mechanisms. Spammers use these accounts to launch spamming campaigns that contain profanity, curse words, adult content, promotion of child pornography and exploitation, and harassment [13]. Spammers then disseminate targeted Twitter spam by exploiting weaknesses of the internet censorship and content filtering systems that use the blacklisted keywords, blacklisted URLs, and blacklisted spamming words [14].

Analyzing tweet content would enable better detection of the spammers. In [13], Singh et al used six filthy keywords in Twitter searches to collect a pornographic spammer dataset. The dataset contains more than 73 thousand tweets and more than 18 thousand users. The result of analyzing the tweets' content distinguished spammer accounts from non-spammer, and had a better performance than using the profile information alone. Therefore, the number of the followers and following of the spammers did not show any difference from a celebrity account, but the tweet content reflects the spammers behavior uniquely; that is, spammers are using filthy words in their tweets, screen names, and profile descriptions (bios).

In [15], Shekar et al studied pharmaceutical spammers on Twitter, where the study shows improved results by using two lists of words instead of one list. The first list is the name of the product, and the second list is the words associated with the products, for example organic, tablet, refill, etc. The classifying result had less false positives when the second list was used.

In [16], Irani et al studied the top trending topic on Twitter by connecting the tweet content to the URL content on the web. One finding was that spammers were using the top trending topic in their tweet either as hashtag, or text, but the URL content was not related to the tweet topic. Also, the study used information gain measures to reduce the features with noise and the improve performance of the classifiers.

In [17], McCord et al used four different machine learning algorithms to identify spammer and non-spammer accounts on Twitter. The features were based on the user information and the content information. The most effective features were: the word weight based on average number of spamming words in the tweet, the number of the hashtags in the tweet, and the time of the tweet. The dataset showed the spammers are using more than one hashtag in each tweet, tweeting in the early morning, and they have more spamming words in their tweets.

Wang et al in [18] detected the new spammers on social media by studying suspended spammers' accounts. In their approach, they first matched the URLs, IP addresses, and hashtags with suspended account data to predict the spammers' accounts. They then classified profile data, text data, and the webpage content to determine similarities between spammer behaviour in different social networks, such as spammers use of profane words in the online community by replacing letters with symbols to bypass filtering systems.

In [19], Yoon et al determined the correct spelling of profane words that have symbols in them. Each word is checked against a list of regular words. If not identified, the word is checked against a profane word list. If not recognized in this list either, then a similarity letter process is applied. The similarity process checks for symbols in the word and replaces them with corresponding letters from a list of letters with matched symbols. After replacing the symbols, the word is again checked against the regular word list and the profanity word list.

3. PROPOSED APPROACH

We propose a domain-specific lexicon approach for choosing the correct words from candidate words based on the content of the tweet. As shown in Figure 1, the process has three main phases: identifying misspelled words with corresponding candidate words, building a list of n-gram words with their frequency, and lastly choosing the correct word.

3.1. Identifying misspelled words

Each word is compared against the Arabic Hunspell dictionary [20] that contain 300,000 words. The string matching is based on Levenshtein Distance algorithm [21] where the edit distance of 0 is an exact word match from the dictionary. The words that match with the dictionary list are considered correct and no further processing is required; otherwise, the word is considered misspelled, and it has a list of candidate corrected words from the dictionary. This approach corrects the misspelled words with a word that matches the tweet content, but does not replace a wrong word that is correctly spelled.

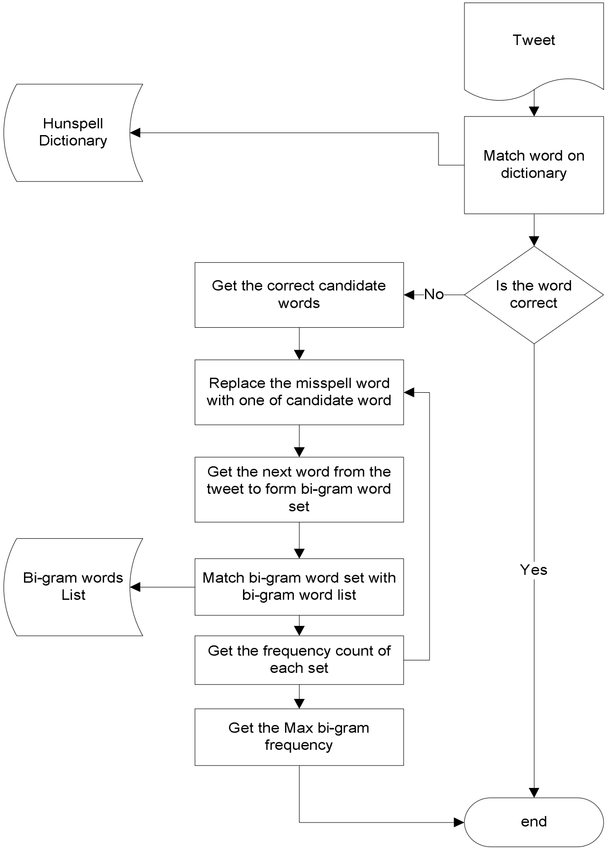
Tweet
Hunspell Dictionary
Match word on dictionary
Get the correct candidate words
No
Is the word correct
Replace the misspell word with one of candidate word
Get the next word from the tweet to form bi-gram word set
Bi-gram words List
Match bi-gram word set with bi-gram word list
Yes
Get the frequency count of each set
Get the Max bi-gram frequency
end

Figure 1. Proposed Approach

The candidate word list is based on the following operation in each letter on the word: insertion, deletion, substitution, and transposition. Insertion is adding one letter in different places to the word. Deletion is deleting a letter from the word. Substitution is replacing a letter with another letter in the word. Transposition is changing the letter’s place with another letter in the word. These operations will lead to matching words from the dictionary that are correctly spelled, but only one is the best match for the tweet.

3.2. N-gram word with frequency count list

In this phase we used the n-gram word list to choose the correct word that fit the tweet meaning. We divided each tweet into n-gram words and counted the frequency of each n-gram word as shown in Table 1.

This list is built by using 1,300,000 tweets came from 49,200 Twitter accounts. These tweets were collected by using five Arabic swearing words that are presented in [22]. We have compared the correction result of three different sizes of n-gram that include unigram (1-gram), bigram (2-gram) and trigram (3-gram) to choose the right size of n for the n-gram list. We randomly picked 300 tweets that had misspelled words and ran them against the three n-gram word lists, then we manually analyzed the word correction of each set. We analyzed the performance of each set by counting the number of misspelled words that were replaced and the number of replaced words that fit the tweet’s meaning.

Table 1. N-Gram List Comparison

	Replaced misspelled	Replaced with correct word
Uni-gram	89%	64%
Bi-gram	80%	91%
Tri-gram	10%	33%

Table 2. Tweet with Misspelling that Corrected by N-Gram Words List

TWEET	CORRECT	UNI-GRAM	BI-GRAM	TRI-GRAM
' الف مبروك فوز المنتخب العراق العراق بجميع طوائفه وقومياته يمثله هذا المنتخب'	ألف	فلا	ألف	الف
الحمد لله رجاء و رخاء و شدة و طاع و الحمد لله يوما و شهرا و عمرا	طاعه	طالعه	طاعه	طاعه

Table 3. Summary of Bi-Gram Words List

Number of tweets	1,300,000
Number of bigrams in word list	2,000,000
5% of bigram word list	100,000

As shown in Table 1, the first column represents the percentage of misspelled words that were replaced with correct words, whereas the rest of the misspelled words were not replaced because there are no matching sequences of words in the n-gram list. The results show the trigram words list is not effective on matching three words and finding the correct word, while the other two n-gram words lists replaced more than 80% of misspelled words. Moreover, the second column presents the percentage of the corrected words that fit within the tweet, as the misspelled word could be corrected by using a word that changes the tweet's meaning. For example, in Table 2, the unigram replaced the misspelled word “thousand” (الف) which is missing Hamza(أ), with word “Don’t” (فلا), but the bigram list detected the misspelled part and corrected it. The bigram word list replaced the misspelled word with the correct word more accurately than the unigram word list as shown in Table 1. Based on the results of comparing the three sizes of n-gram word lists, we selected the bigram word list for correction since it has a higher percentage of replacements that match the tweet's meaning.

The total tweets shown in Table 3 produced 2,000,000 bigram words. This list contains 5% of bigram words with frequency of five or more, and the rest of the list has a frequency less than five.

3.3. Choosing the correct word

One candidate word from the suggested candidate word list is assigned to replace each misspelled word. The replaced word will be used with the next word in the tweet to form a bigram words set. Each bigram words set is compared to the bigram word list to find the exact word match (edit distance of zero). The set with higher frequency is used as the corrected word.

Table 4. Summary of Dataset

Type of Content	Total
Accounts	350,000
Tweets	1,300,000
Hashes	530,000
Links (URLs)	1,150,000

Otherwise, if the word is not part of the bigram word list, the word is identified as unknown, and is not replaced.

The word dictionary has a limited word set that does not cover local dialects and slang. The tweets contain some informal Arabic words that cannot be found in MSA dictionary. Therefore, some of the misspelled words are not corrected and are kept without change.

4. Dataset

This section describes the dataset collection, dataset normalizing, sub-datasets, features, and classifying method.

4.1. Collecting the dataset

As noted above, we used 1,300,000 tweets to create the bigram words list containing misspelled words. To ensure the minimum number of misspelled words in the bigram words list, we randomly chose different sets of 500 bigram words with frequency of 1,2,3,4, and 5 respectively to reduce the number of misspelled words. We found that sets with frequencies of 1,2,3, and 4 have over 70% misspelled words in each set, but the set with a frequency of 5 has 14% misspelled words. Therefore, we used the bigrams set with frequency equal to and higher than 5, and this set is about 5% of the total bigrams set. This list was used to choose the correct the word from the candidate suggested words.

4.2. Normalizing the dataset

We normalized the tweets using the following steps:

- Removed all non-Arabic words.
- Removed all symbols.
- Removed all digits.
- Removed all the stop words by using the stop word list in [23]–[25].
- Removed all sequences of letters in the words except the name of God (Allah- الله) [22].
- Removed all diacritics.
- Removed all extra whitespaces.

4.3. Sub-dataset

The normalized data set is used to create eight sub-datasets of: clear tweets, basic normalization, edit distance, proposed approach, and four other datasets that are the same previous four sets with the light stemmer applied. The description of the first four datasets is:

- Clear tweets dataset is the normalized dataset without applying any further process of correcting misspelled words.
- Basic normalization dataset is the normalized dataset with the basic Arabic normalization process to correct the most common Arabic misspellings. This is the same as the one that was used in [9], [26]:
-
    - Converting آ – إ – أ to ا
    - Converting ى to ي
    - Converting ة to ه
    - Converting ئ – ؤ to ء

- Edit Distance dataset is the normalized dataset in which we choose the correct word by using an edit distance of 1. The edit distance is a measure of the four operations of deletion, substitution, indentation, transposition [21].
- Proposed Approach dataset is the normalized dataset in which we apply the proposed approach of correcting misspelled words as described above.

The rest of the four datasets are the first four datasets with the light stemmer applied. The light stemmer removed the prefix and suffix without dealing with infix or getting the root of the words. Arabic text mining has better performance with a light stemmer than a root stemmer [9], [27]. The two most common Arabic light stemmers are ISRI [24], [28] and Tashaphyne [29]. In our work the ISRI light stemmer was used.

4.4. Features

This paper uses bag of words (BOW) as the feature selector for the classifier because of the simplicity of this approach with our datasets [30].

4.5. Classification Method

One commonly well-performing classifier in text mining is the Support Vector Machine (SVM) algorithm [31]–[33]. SVM is a set of associated supervised learning methods that uses a maximal decision boundary between two classes. We evaluated the performance of an SVM classifier based on average precision (P), average recall (R), average F-measure (F), and accuracy (A). All four measures are computed based on the confusion matrix and provide a balanced approach to evaluating classifiers in general. The confusion matrix is presented in Table 5 where (TP, or True Positive) represents the number of non-abusers correctly classified as non-abuser, (FN, or False Negative) represents number of non-abusers incorrectly classified as abuser, (TN, or True Negative) represents number of abusers correctly classified as abuser, and (FP, or False Positive) represents number of abusers incorrectly classified as non-abuser. The precision (P), and recall (R) are measures of completeness and exactness respectively.

5. EVALUATION

In this study we have used the same dataset in [22] summarized in Table 4. We randomly selected 2,500 Twitter accounts with more than 100 tweets each. These accounts were manually analyzed by three researchers to identify the abusive accounts (spammers) and legitimate accounts (non-spammers). Each account was categorized based on the tweets' content, hashtags, and pictures. The researchers agreed on 407 abusive accounts and 581 legitimate accounts.

The researchers' consensus is based on voting that has either three out of three or two out of three voting for the same account category.

Correcting the misspelled words generally improves text classification performance and the overall result [2], [9], [34], [35]. To evaluate the performance of the proposed approach, we classify the Twitter accounts to find appropriate word correction, and compare this result with other word correction approaches to find the quality of the improved result.

We classified each of the eight datasets using two classes: abusive accounts and non-abusive accounts, and compared the performance result of each dataset. From the manually categorized twitter accounts, we randomly picked 403 Abusive Twitter accounts and 403 non-abusive Twitter accounts, each of which had at least 100 tweets. Tweets in each dataset were converted into BOW for classification.

Classifier accuracy is over 90% for all eight datasets, but better on all datasets without stemming. The performance of the classifier on each dataset is shown in Table 8. The classifier for the dataset of our proposed approach with word correction but not light stemming performed better across all measures than the rest of the datasets.

A comparison of the results based on the confusion matrices is shown in
Figure 2 for the basic normalized dataset and the proposed approach. The True positive (TP) rate shows improvement of 0.5% for identifying the non-abusive accounts, which implies the use of word correction improves the abusive account detection performance. In addition, the confusion matrices of
Table 6 and Table 7 shows the FP of the basic normalized dataset is higher than the proposed approach, which shows the ability of the proposed approach to detect spammers using misspelled words to evade detection.

Additionally, the basic normalization method deals with four common typing mistakes, but the proposed approach has the ability to cover all kinds of misspelled mistakes by using a larger dictionary.

The proposed approach can be improved by using a larger dictionary that contains slang and local dialect words. Also, the proposed approach has a lower false positive rate, where it identifies some spammers as non-spammers. In cyber-crime networks it is better to have a false negative than a false positive, assuming the false negative does not disproportionately affect the legitimate network.

Table 5. Confusion Matrix

	Non-Spammers	Spammers
Non-Spammers	True Positive (TP)	False Negative (FN)
Spammers	False-Positive (FP)	True-Negative (TN)

Table 6. Basic Normalized Dataset Confusion Matrix

	Non-Spammers	Spammers
Non-Spammers	394	9
Spammers	23	380

Table 7. Proposed Approach Confusion Matrix

	Non-Spammers	Spammers
Non-Spammers	395	8
Spammers	20	383

The stem datasets provided worse performance results than the non-stem datasets, which reflect the need for the full word length in Arabic text classification. The full word length reflects the gender, time, and population, which would be lost when using the stemmed word. For example, one spammer tweeting behavior is to talk about the future businesses and interactions, not about their past experience, which would be missed by stemming.

6. CONCLUSION AND FUTURE WORKS

This paper proposes a tweet-based word correction approach that uses Arabic text classification to detect abusive accounts on Twitter. Our approach outperforms other common approaches on Arabic word correction in Twitter. The performance result shows the drawback of using stemming in Arabic tweets compared to classification of tweets without stemming.

For future work, we will be collecting more tweets to refine the mechanism of choosing the correct word; we need to collect more tweets to build a larger list of bigram words with their frequencies. Also, the dictionary we used has limitations on dealing with slang and dialect words; therefore, we need to build a larger dictionary that covers all kinds of Arabic words.

Table 8. Classifier Performance of Eight Data-Sets

	P	R	F	A
Clean Data-set	0.958	0.957	0.957	0.957
Basic Normalized Data-set	0.961	0.96	0.96	0.96
Edit Distance Data-set	0.957	0.957	0.957	0.957
Propose Approach Data-Set	0.966	0.965	0.965	0.965
Stem-Clean Data-set	0.963	0.963	0.963	0.963
Stem-Basic Normalized Data-set	0.961	0.960	0.960	0.960
Stem-Edit Distance Data-set	0.963	0.963	0.963	0.963
Stem-Propose Approach Data-Set	0.963	0.963	0.963	0.963

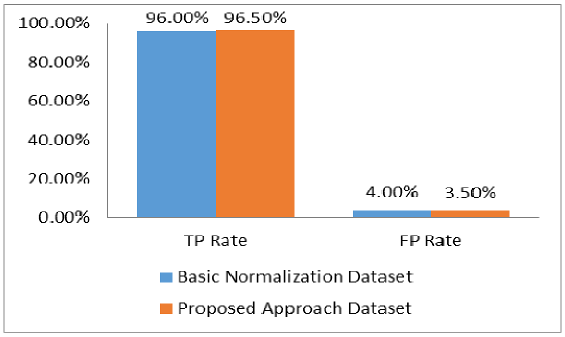

Figure 2. Confusion Matrix Result of TP and FP Rate

REFERENCES

[1] H. Muaidi and R. Al-tarawneh, “Towards Arabic Spell-Checker Based on N-Grams Scores,” Int. J. Comput. Appl., vol. 53, no. 3, 2012.

[2] T. M. Miangah, “FarsiSpell: A spell-checking system for Persian using a large monolingual corpus,” Lit. Linguist. Comput., p. fqt008, Feb. 2013.

[3] K. Yeung, “61 Languages Found On Twitter. Here’s How They Rank In Popularity.,” The Next Web, 10-Dec-2013. [Online]. Available: http://thenextweb.com/shareables/2013/12/10/61-languages-found-twitter-heres-rank-popularity/. [Accessed: 30-Sep-2016].

[4] “Twitter in the Arab Region.” [Online]. Available: http://www.arabsocialmediareport.com/Twitter/LineChart.aspx?&PriMenuID=18&CatID=25&mnu=Cat. [Accessed: 03-Oct-2016].

[5] A. Bruns, T. Highfield, and J. Burgess, “The Arab Spring and Social Media Audiences English and Arabic Twitter Users and Their Networks,” Am. Behav. Sci., vol. 57, no. 7, pp. 871–898, Jul. 2013.

[6] C. Christensen, “Twitter Revolutions? Addressing Social Media and Dissent,” Commun. Rev., vol. 14, no. 3, pp. 155–157, Jul. 2011.

[7] Z. Harb, “Arab Revolutions and the Social Media Effect,” MC J., vol. 14, no. 2, Apr. 2011.

[8] R. M. Duwairi, R. Marji, N. Sha’ban, and S. Rushaidat, “Sentiment Analysis in Arabic tweets,” in 2014 5th International Conference on Information and Communication Systems (ICICS), 2014, pp. 1–6.

[9] R. Sallam, H. Mousa, and M. Hussein, “Improving Arabic Text Categorization using Normalization and Stemming Techniques,” Int. J. Comput. Appl., vol. 135, no. 2, pp. 38–43, Feb. 2016.

[10] K. Shaalan, A. Allam, and A. Gomah, “Towards automatic spell checking for Arabic,” in Proceedings of the 4th Conference on Language Engineering, Egyptian Society of Language Engineering (ELSE), Cairo, Egypt, 2003, pp. 21–22.

[11] K. Shaalan, M. Attia, P. Pecina, Y. Samih, and J. van Genabith, “Arabic Word Generation and Modelling for Spell Checking,” in Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12), Istanbul, Turkey, 2003.

[12] I. Elbadawi, “Social Media Usage Guidelines for the Government of the United Arab Emirates,” in Proceedings of the 6th International Conference on Theory and Practice of Electronic Governance, New York, NY, USA, 2012, pp. 508–510.

[13] M. Singh, D. Bansal, and S. Sofat, “Behavioral analysis and classification of spammers distributing pornographic content in social media,” Soc. Netw. Anal. Min., vol. 6, no. 1, p. 41, Jun. 2016.

[14] A. Chaabane, T. Chen, M. Cunche, E. De Cristofaro, A. Friedman, and M. A. Kaafar, “Censorship in the Wild: Analyzing Internet Filtering in Syria,” ArXiv14023401 Cs, Feb. 2014.

[15] R. Shekar, K. J. Liszka, and C. Chan, Twitter on Drugs: Pharmaceutical Spam in Tweets. Conference Proceedings of the 2011 International Conference on Security and Management. Las Vegas, 2011.

[16] D. Irani, S. Webb, and C. Pu, “Study of trend-stuffing on twitter through text classification,” in In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS, 2010.

[17] M. McCord and M. Chuah, “Spam Detection on Twitter Using Traditional Classifiers,” in Autonomic and Trusted Computing, J. M. A. Calero, L. T. Yang, F. G. Mármol, L. J. G. Villalba, A. X. Li, and Y. Wang, Eds. Springer Berlin Heidelberg, 2011, pp. 175–186.

[18] D. Wang, D. Irani, and C. Pu, “A Social-spam Detection Framework,” in Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference, New York, NY, USA, 2011, pp. 46–54.

[19] T. Yoon, S.-Y. Park, and H.-G. Cho, “A Smart Filtering System for Newly Coined Profanities by Using Approximate String Alignment,” in 2010 IEEE 10th International Conference on Computer and Information Technology (CIT), 2010, pp. 643–650.

[20] “Ayaspell project.” [Online]. Available: http://ayaspell.sourceforge.net/index.html. [Accessed: 06-Oct-2016].

[21] V. Levenshtein, “Binary Codes Capable of Correcting Deletions, Insertions and Reversals,” vol. 10, no. 707, 1966.

[22] E. A. Abozinadah, A. V. Mbaziira, and J. H. J. Jones, “Detection of Abusive Accounts with Arabic Tweets,” Int. J. Knowl. Eng.-IACSIT, vol. 1, no. 2, pp. 113–119, 2015.

[23] “stop-words - Stop words - Google Project Hosting.” [Online]. Available: https://code.google.com/p/stop-words/. [Accessed: 22-Nov-2014].

[24] "nltk.stem.isri — NLTK 3.0 documentation." [Online]. Available: http://www.nltk.org/_modules/nltk/stem/isri.html. [Accessed: 07-Oct-2016].

[25] "PyArabic 0.5 : Python Package Index." [Online]. Available: https://pypi.python.org/pypi/PyArabic/0.5. [Accessed: 07-Oct-2016].

[26] K. Darwish, W. Magdy, and A. Mourad, "Language processing for arabic microblog retrieval," in Proceedings of the 21st ACM international conference on Information and knowledge management, 2012, pp. 2427–2430.

[27] M. K. Saad and W. Ashour, "Arabic morphological tools for text mining," Corpora, vol. 18, p. 19, 2010.

[28] K. Taghva, R. Elkhoury, and J. Coombs, "Arabic stemming without a root dictionary," in International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II, 2005, vol. 1, pp. 152–157 Vol. 1.

[29] "Tashaphyne 0.2 : Python Package Index." [Online]. Available: https://pypi.python.org/pypi/Tashaphyne/. [Accessed: 07-Oct-2016].

[30] W. Pu, N. Liu, S. Yan, J. Yan, K. Xie, and Z. Chen, "Local Word Bag Model for Text Categorization," in Seventh IEEE International Conference on Data Mining (ICDM 2007), 2007, pp. 625–630.

[31] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," in Machine Learning: ECML-98, C. Nédellec and C. Rouveirol, Eds. Springer Berlin Heidelberg, 1998, pp. 137–142.

[32] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," J. Mach. Learn. Res., vol. 2, no. Nov, pp. 45–66, 2001.

[33] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," J Mach Learn Res, vol. 2, pp. 45–66, Mar. 2002.

[34] Y. Bassil, "Parallel Spell-Checking Algorithm Based on Yahoo! N-Grams Dataset," ArXiv12040184 Cs, Apr. 2012.

[35] V. H. Nguyen, H. T. Nguyen, and V. Snasel, "Normalization of vietnamese tweets on twitter," in Intelligent Data Analysis and Applications, Springer, 2015, pp. 179–189.

Authors

Ehab A Abozinadah earned a MSc in Information Systems from George Mason University, Fairfax VA in 2013, Graduate Certificate in Information Security Assurance from George Mason University in 2012, a MEd Information Technology at Western Oregon University 2008 and a BSc in Computer Science in 2004. He is currently pursuing a PhD in Information Technology (Cyber Crime) at George Mason University, Fairfax, VA. Also, he is a lecturer in the School of Information Technology at King Abdul-Aziz University. He was previously Director of quality assurance of e-learning systems in King Abdul-Aziz University, Saudi Arabia. His research interests are cybercrime, machine learning and social networks.

James H. Jones, Jr. earned a PhD in Computational Sciences and Informatics from George Mason University in Fairfax, Virginia, USA, in 2008, a MS in Mathematical Sciences from Clemson University in Clemson, South Carolina, USA, in 1995, and a BS in Industrial and Systems Engineering from Georgia Tech in Atlanta, Georgia, USA, in 1989. He is currently an Associate Professor at George Mason University in Fairfax, Virginia, USA. He was previously on the faculty of Ferris State University, served as a Principal Scientist, Chief Scientist, and Chief Technology Officer for business units of SAIC, was a Research Scientist at the Georgia Tech Research Institute, and was a Scientist for the US Department of Defense. His research interests are focused on digital artifact extraction, analysis, and manipulation, and on offensive cyber deception in adversarial environments. Dr. Jones is a member of IAFIE, ACM, and IEEE.