2018 17th IEEE International Conference on Machine Learning and Applications

Comparison of Pre-trained Word Vectors for Arabic Text Classification using Deep Learning Approach

Ali Alwehaibi
Department of Computational Science & Engeneering
North Carolina A&T State University
Greensboro, USA
asalweha@aggies.ncat.edu

Kaushik Roy
Department of Computer Science
North Carolina A&T State University
Greensboro, USA
kroy@ncat.edu

***Abstract*— Artificial Intelligence (AI) has been used widely to extract people's opinions from social media websites. However, most of the existing works focus on eliciting the features from English text. In this paper, we describe an Arabic text sentiment analysis approach using a Deep Neural network, namely Long Short-Term Memory Recurrent Neural Network (LSTM-RNN). In this research, we investigate how the different pre-trained Word Embedding (WE) models affect our model's accuracy. The dataset includes Arabic corpus collected from Twitter. The results show significant improvement in Arabic text classification.**

Keywords—sentiment analysis, natural language processing, deep learning, long-short term memory

I. INTRODUCTION

Text classification has become a remarkable research area in Natural Language Processing (NLP) due to the increase of users' posts on various social networks. Sentiment Analysis, as an instance of text classification, aims to automatically categorize opinions within a text. While most current research is conducted on English text, Arabic text sentiment classification is still challenging due to its complex morphology, dialectal varieties, and the lack of availability of corpora and tools.

The existing techniques for opinion mining include machine learning, lexicon-based and hybrid approaches [1]. According to the results reported in [1], the machine learning approach outperformed the lexicon-based approach in terms of accuracy and precision. In terms of text granularity, sentiment analysis is studied at three levels: document level, sentence level and aspect level.

In this paper, we propose to use a Deep Learning (DL) model, namely Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) [1] for Arabic text sentiment analysis. We investigate the effect of pre-trained word vectors on a dialectal Arabic tweets dataset [2].

II. RELATED WORK

Sentiment classification aims to classify the sentiment polarity of a subjective text as positive, negative or neutral [3]. Sentiment classification has been a hot research topic due to the availability of a huge number of users' reviews and opinions about a product or topic on the social media.

While Arabic language recently became the 4th most used language on the web, few studies have addressed the sentiment analysis in Arabic[1]. This is due to the morphological complexity of Arabic language, the lack of publicly available corpora, and the utilization of Modern Standard Arabic (MSA) and various dialects. These challenges make the deep learning (DL) approach a good candidate for text classification [4].

Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) [5] are the main DL architectures to handle NLP tasks [6]. Gated Recurrent Units (GRU)[7] and Long Short-Term Memory networks (LSTM) are developed to overcome some limitations found in RNN. Each type may perform better than the other depending on NLP task [6].

LSTM-RNN networks, as a gated RNN architecture, remember longer sequences, which is more effective for text classification than regular RNN and CNN[4, 8]. Recent studies show that LSTM performs relatively well on most of the NLP tasks [9, 10].

Textual feature extraction plays a critical role in text classification, directly influencing the accuracy of text categorization [11]. Since language data come in the form of a sequence of discrete symbols, this must be presented in a meaningful way for further processing. Based on the NLP problem, features may contain letters, single words, phrases, sentences, or even documents [12]. Existing text feature extraction methods include filtration, fusion, mapping, and clustering [11]. Recently, DL approaches have been used heavily to generate distributed feature representation of words, beside classification.

Word Embedding (WE) is a useful method to extract numeric features from words. In WE, each word is presented as a vector in high-dimensional space. This is an effective method to capture relations between words by mapping nearby and similar words in the space. Word2Vec [13], GloVe [14] and FastText [15] are most popular embedding methods and have been used extensively for accurate and reliable WEs. However,

[1] https://www.internetworldstats.com/stats7.htm

978-1-5386-6805-4/18/$31.00 ©2018 IEEE
DOI 10.1109/ICMLA.2018.00239

training these WEs from the scratch requires huge text corpus and, consequently, requires a long time. Instead, various pre-trained WE vectors are publicly available for NLP tasks. Very few pre-trained WEs serve as Arabic WE techniques. In [16] FastText method, was employed to publish 249 pre-trained word vectors; two of which were for Arabic language. AraVec [17] has multiple versions based on skip-gram and continuous bag of words (CBOW) models of Word2Vec.

III. METHODOLOGY

A. Data Collection and Preparation

In our research, we implement a DL framework to evaluate the effectiveness of multiple pre-trained WE vectors in Arabic text classification. We use an AraSenTi dataset collected by [2] to test our model for 3 classes (positive, negative, neutral).

The AraSenTi datasets mainly consist of tweets written in MSA and Saudi dialect and has been manually annotated for sentiment. It was originally labelled as negative, positive, neutral and mixed. We ignore tweets with mixed sentiment. With a total of about 15k tweets, divided equally into 3 classes, we conduct two steps to clean text. First, we remove diacritics (tashkeel) from text by excluding Unicode[2] characters between U+0617-U+061A and between U+064B - U+0652. Next, the elongated words with repeated letters without spaces in between are corrected by keeping only one letter. The multiple forms of letters (أ,ة,ى,و) are normalized by converting multiple shapes of the letter to one shape. For example, the different forms of letter "alif" (ا,آ,إ,أ) are converted into (ا).

B. Feature Extraction

Since traditional techniques based on the *n*-gram model ignore relationships between words in text, we choose vector-space word representation models to extract numeric features from text. Word2Vec, Glove and FastText are among the most reliable and accurate methods to represent words in dimensional vectors, as stated before [13-15] . However, these methods require billions of words to build and train an effective model. Alternatively, users can use publicly available pre-trained word WE techniques. The majority of pre-trained WEs are built using CBOW or skip-gram models.

In our experiment, we tested three publicly available Arabic pre-trained WEs, reported in [17], [18] and [19], and these WEs are AraVec[3], Arabic FastTxt[4], called AraFT hereafter, and Arabic_news[5]. AraVec was trained on Wikipedia Arabic corpus using the Word2Vec method by applying CBOW for 300 dimension and with a window of size 5.

On the other hand, AraFT model was trained using the FastText method by applying skip-gram. This model uses the position-weights in 300 dimensions, character *n*-grams of length 5, and a window of size 5. Arabic_news was trained on multiple Arabic corpus sources using Word2Vec with CBOW. In this research, we prefer to select pre-trained WEs that are built with CBOW over skip-gram since literature has suggested that CBOW performed slightly better than skip-gram in learning words vectors [13].

C. Classification Using LSTM

In this research effort, we applied LSTM for classification purpose because it explicitly uses special hidden units as memory. This is particularly useful in our case because these units recall input words in the context of a long time sequence. Fig. 1 shows a single LSTM cell. The operations of the LSTM cell are specified by the following equations [20]:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2)$$

$$c_t = f_t\, c_{t-1} + i_t \tanh\,(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (4)$$

$$h_t = o_t \tanh\,(c_t) \quad (5)$$

Where σ is the logistic sigmoid function and i, f, o and c are the input gate, forget gate, output gate and cell activation vectors, respectively.

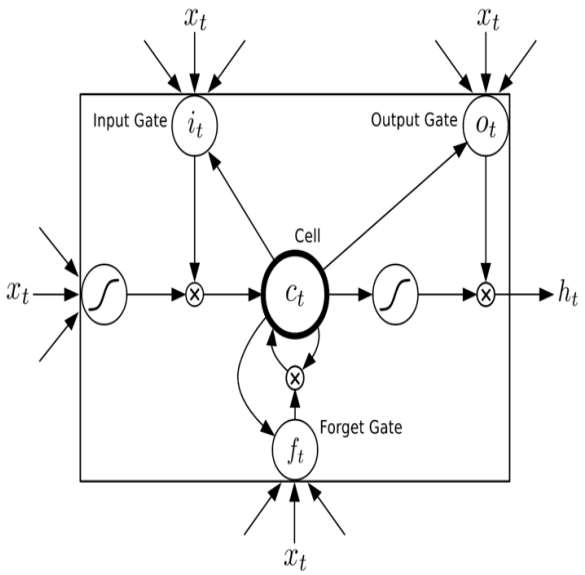

Fig. 1. LSTM-RNN cell [20]

Fig. 2 shows the sentiment classification model used in this research. It started with a pre-trained word embedding technique to generate 300-dimensional word vectors for each word in a tweet. Then, the embedding is fed to LSTM layer with a 128-dimensional hidden state that returns a batch of sequences. A dropout of 0.5 fraction rate is applied over the batch of sequences and then fed to another LSTM layer with

[2] http://www.unicode.org
[3] https://github.com/bakrianoo/aravec
[4] https://fasttext.cc/
[5] https://github.com/iamaziz/ar-embeddings

128-dimensional hidden state that returns a single hidden state. Similarly, a dropout of 0.5 fraction rate is applied to the returned sequences. Finally, a dense layer with 3 units is applied to perform 1 of 3 possible classes followed by softmax activation.

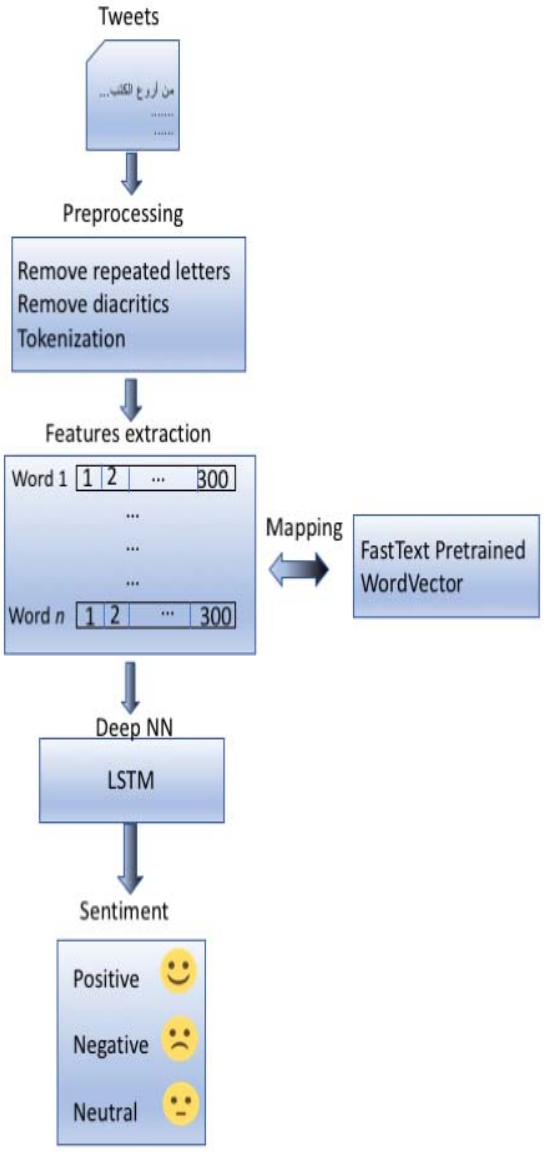

Fig. 2. Proposed sentiment clssification model

TABLE 1: COMPARISON OF DIFFERENT WES TECHNIQUES

Pre-trained WV	Dialect	# of vocab	Acc.%	F-score%
AraVec	Yes	3.3 b	88	40
ArabicNews	No	190 m	91	43
AraFT	N/A	N/A	93.5	41

D. Result and Discussion

AraSenTi dataset is divided into 2 sets: 80% for training and 20% for testing data. Table 1 shows the accuracy of three different WEs used to generate word vectors. AvaVec and ArabicNews were trained on 3.3 billion and 190 million words, respectively. AraVec achieved an accuracy of 88% and a F-score of 40%. While ArabicNews achieved an accuracy rate of 91% and F-score of 43%, AraFT scored an accuracy rate of 93.5% and F-score of 41%. To the best of our knowledge, our approach using AraFT obtained higher accuracy than the-state-of-arts. AraFT obtained the highest accuracy because it considered morphological complexity when building pre-trained WE vectors. However, no information was provided about the number of words used to train AraFT and whether or not it used MSA or dialects.

IV. CONCLUSION AND FUTURE WORK

In this paper, we implement LSTM framework to train text sentiment within dialectal Arabic tweets. We evaluate the effect of existing pre-trained WE vectors for numerical words representation. Experimental results show that LSTM-RNN model achieves a reasonable accuracy of 93.5% with AraFT. In future work, we will work on extracting morphological features in word level and grammar features in sentence level.

ACKNOWLEDGEMENTS

This research is based upon work supported by the Science & Technology Center: Bio/Computational Evolution in Action Consortium (BEACON) and the National Science Foundation.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[2] N. Al-Twairesh, H. Al-Khalifa, A. Al-Salman, and Y. Al-Ohali, "AraSenTi-Tweet: A Corpus for Arabic Sentiment Analysis of Saudi Tweets," *Procedia Computer Science,* vol. 117, pp. 63-72, 2017.

[3] B. Liu, "Sentiment analysis and opinion mining," *Synthesis lectures on human language technologies,* vol. 5, no. 1, pp. 1-167, 2012.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature,* vol. 521, no. 7553, p. 436, 2015.

[5] J. L. Elman, "Finding structure in time," *Cognitive science,* vol. 14, no. 2, pp. 179-211, 1990.

[6] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923,* 2017.

[7] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259,* 2014.

[8] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422-1432.

[9] Y. Wenpeng, K. Katharina, Y. Mo, and S. Hinrich, "Comparative study of CNN and RNN for natural language processing," *arXiv preprint arXiv: 1702.01923,* 2017.
[10] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *arXiv preprint arXiv:1511.08308,* 2015.
[11] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: a review," *EURASIP journal on wireless communications and networking,* vol. 2017, no. 1, p. 211, 2017.
[12] Y. Goldberg, "Neural network methods for natural language processing," *Synthesis Lectures on Human Language Technologies,* vol. 10, no. 1, pp. 1-309, 2017.
[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.
[14] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532-1543.
[15] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759,* 2016.
[16] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606,* 2016.
[17] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP," *Procedia Computer Science,* vol. 117, pp. 256-265, 2017.
[18] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning Word Vectors for 157 Languages," *arXiv preprint arXiv:1802.06893,* 2018.
[19] A. A. Altowayan and L. Tao, "Word embeddings for Arabic sentiment analysis," in *Big Data (Big Data), 2016 IEEE International Conference on*, 2016, pp. 3820-3825: IEEE.
[20] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, 2013, pp. 6645-6649: IEEE.