НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н. Г. ЧЕРНЫШЕВСКОГО

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

ИНСТИТУТ ПРОБЛЕМ УПРАВЛЕНИЯ
им. В.А. ТРАПЕЗНИКОВА РАН

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ (ИТММ-2019)

МАТЕРИАЛЫ
XVIII Международной конференции
имени А. Ф. Терпугова
26–30 июня 2019 г.

Часть 1

ТОМСК
«Издательство НТЛ»
2019

ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ И ВИЗУАЛИЗАЦИЯ ДАННЫХ

Arabic Word Embedding Method for NLP

Ghassan Khazal, Alexander Zamyatin

Computer Science Department, Tomsk State University, Tomsk, Russia

One of the most important developments in natural language processing (NLP) is word embedding representation. In this method, words are represented in a vector space by capturing the semantic and syntactic relationships between them. Despite the Arabic language being the national language in 22 countries and being spoken by more than 400 million people, few resources are available for it such as few corpora and datasets that are appropriate for computational tasks. Nowadays, research is focusing mostly on English. In this paper, we seek to build word embedding models for the Arabic language with different dimensions in an attempt to improve the performance of Arabic language processing in several machine learning algorithms and to build high-level complex representations of the Arabic language that can be used in NLP applications. We evaluate our model by experimenting with different parameters and measuring their performance using text similarity tasks.

Introduction

Researchers proposed various techniques to represent huge unstructured data, one of these methods that adopted by many researchers is representing the text data in multidimensional space vector by capturing the semantic and syntactic properties of the language to serve as necessary step in many natural language processing (NLP) applications[1]. Word embedding (representation) is the collective name for a set of language modeling and feature learning techniques in NLP where words or phrases from the vocabulary are mapped to vectors of real numbers. Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with a much lower dimension [2]. This model represents a mathematical model each component is a feature to that term that may be have a semantic or syntactic meaning. The benefits of using vector representations have been explained in many different NLP tasks including but not limited to, information retrieval, clustering, text classification, sentiment analysis, entity recognition, and part of speech tagging. These benefits were accompanied by the provision of several word representation models in English, but we cannot say the same for the Arabic [3].

This work aims to provide the Arabic NLP powerful word embedding models. The presented models were built carefully using multiple different Arabic text resources to provide wide domain coverage. Specifically, the models were built using web pages collected from World Wide Web, text harvested from social platforms and text obtained from encyclopedia entries. This paper describes the various steps followed for the creation of these models.

Background

1. Word2vec

This model was proposed by Mikolov [4]. The intuitive idea behind Word2Vec models is to train deep neural networks in order to predict the context given a word, and vice versa. The model to predict the context [..., $w(t-2)$, $w(t-1)$, $w(t+1)$, $w(t+2)$, ...] given a word $w(t)$ is known as the Skip-gram model, while the model to predict the word that goes in the middle given the context is known as the Continuous Bag of Words (CBOW) model. Figures 1 and 2 show the architecture of the both models[5].

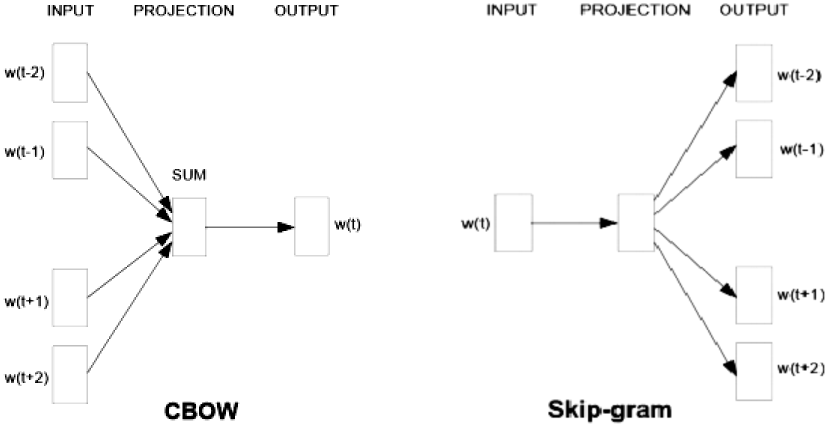

Fig. 1. Representation architecture of CBOW and skip-gram model

In CBOW process there are three layers used. First layer corresponding to the context, the second used to projection of each word from the input layer, and the last layer is output layer. As shown in the following equation [6]:

$$\frac{1}{V}\sum_{t=1}^{V}\log p(m_1 \mid m_{t-\frac{c}{2}} \ldots\ldots m_{t+\frac{c}{2}}), \tag{1}$$

where V represents the size of the vocabulary, c is the window size.

Is opposite of CBOW, in skip-Gram the first layer representing the target word and the output layer is corresponding to the context [6]:

$$\frac{1}{V}\sum_{t=1}^{V}\sum_{j=t-c,j\neq t}^{t+c}\log p(m_j \mid m_t), \tag{2}$$

where V represents the size of the vocabulary, c is the window size

2. GloVe: Global Vectors

This model was proposed by Pennington [7]. This model is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. It is developed as an open-source project at Stanford. The intuitive idea behind GloVe model is to build a very big matrix, X, of co-occurrence words from a corpus. Each cell of the matrix, X_{ij} represents how many times has the row word, w_i, appeared in some context, c_j. By doing a simple normalization of the values for each row of the matrix, we can obtain the probability distribution of every context given a word [8].

Experiment setup

1. Proposed datasets

We use different Arabic datasets to construct this model; first we have downloaded the latest Arabic dump of the Wikipedia to process it. This is an XML file that contains all the information of the articles of the Arabic language in Wikipedia also we use other datasets shown in Table 1. Then we extract the all articles into a plain text file, then tokenization and remove unnecessary text like non_Arabic characters and numbers and punctuation because we focus on word embedding representation. After this processing, the resulting text file around (3GB) is composed of a single line of approximately 91 000 000 words.

Table 1

Represent statistics of all used corpuses

Corpus	# tokens
Wikipedia[1]	75000000
Alwatan2004[2]	7000000
KALIMAT a Multipurpose Arabic Corpus[3]	18000000
Arabic in Business and Management Corpora[4]	45000
Essex Arabic Summaries Corpus (EASC)[5]	56000

2. Experiments results and evaluation

We train the word2vec model using Gensim toolkit[6], and Glove model using Tensorflow toolkit[7]. For run experiments we use different window sizes (3, 5, and 7) and different embedding dimensions (100, 200, 300) also we choose 50 minimum frequency for each word, finally we train the model for 20 epoch over the entire text.

Here we use cosine as a baseline and we test an adaptation of a rank-based measure to the dense features of the word embeddings. Vector cosine computes the correlation between the entire vector dimensions, independently of their relevance for a given word pair or for a semantic cluster, and this could be a limitation for discerning different degrees of dissimilarity. The alternative rank-based measure is based on the hypothesis that similarity consists of sharing many relevant features, whereas dissimilarity can be described as either the non-sharing of relevant features or the sharing of non-relevant features [12, 14].

The similarity between any two words can be evaluated using cosine similarity, Euclidean distance, Manhattan distance or any other similarity measure functions. For example:

$$Sim\left(w_i, w_j\right) = cos\left(V\left(w_i\right), V(w_j)\right), \tag{2}$$

where w represent words and V – their vector weighting representation.

[1] https://www.kaggle.com/abedkhooli/arabic-wiki-data-dump-2018

[2] https://sourceforge.net/projects/arabiccorpus/files/

[3] http://www.lancs.ac.uk/staff/elhaj/corpora.htm

[4] http://www.lancaster.ac.uk/staff/elhaj/corpora.htm

[5] http://www.lancaster.ac.uk/staff/elhaj/corpora.htm

[6] https://radimrehurek.com/gensim/about.html

[7] https://www.tensorflow.org/

We find the top 6 closest vector to the given word in vector space using cosine similarity to see that the model have captured semantic or syntactic relationships between Arabic words. For example:

Glove (query about Cities: بعقوبة): (ديالى, 0.57272, تكريت, 0.52928, سامراء, 0.52312, واسط, 0.51728, صلاح_الدين, 0.50752, بغداد, 0.50736)
Word2vec (query about places: مطعم): (المطعم, 0.61416, مقهي, 0.58816, حانة, 0.575904, مطاعم, 0.554, فندق, 0.55048, كازينو, 0.52672)

For evaluation of the learned word embeddings, we use WordSim-353 [9], WordSim-353 contains 353 word pairs with relatedness scores assigned by 13 to 16 human subjects, and their average used as the final score.

We trained our model with the given parameters and its results in vocabulary over then 600 000 words. The evaluation of our model using Arabic dataset developed by [9] based on the classic WordSim353 [10], as is evaluated on by [11], different experiments results are shown the Table 2.

Table 2

Similarity scores on word similarity datasets for the two models

	Glove			**Word2Vec**		
	100	**200**	**300**	**100**	**200**	**300**
3	0.50856	0.53664	**0.58016**	0.4432	0.472	0.49392
5	0.47232	0.54496	0.49248	0.49224	0.50568	0.53216
7	0.47136	0.4884	0.5304	0.51568	0.55016	**0.56304**

Our best performance was achieved with Glove model of (58 %) and with word2vec of (56 %) We also take a look at kinds of many words relationships captured in the two models.

We compared created datasets on two popular word representation models, based on Word2Vec tool and Glove tool. Results show that models are able to good meaningful word representation. This research has shown that free words order and the higher morphological complexity of Arabic language influences the quality of resulting word embeddings.

This caused by many reasons:

1. Lack of context. (Difference in mapping between source and target words (1 original word from the English dataset can be translated into two or more Arabic words) in the evaluation dataset.

2. The problem of non-standard words: some words in Arabic may have different meanings than those in the standard language.

Conclusions

In this paper, we present two large-scale word embeddings for the Arabic language from different corpuses. Moreover, neural networks have approved that it is an efficient unsupervised method for word embedding from huge amounts of text data. From these experiments, we can conclude that Arabic word embeddings are indeed a powerful tool. During this experiment, we have observed that if the representation space of the vectors is big enough, word embedding model scan be built and a correlation of 60 % can be achieved when calculating the similarity between two words. We think that these two pre-trained models can improve the performance of Arabic NLP tasks in different applications. In the future our plan is to refining this method for Arabic as well as in deep learning method for text classification.

REFERENCES

1. *Marwa Naili, Anja H. Chaibi, Henda H. Ben Ghezala.* Comparative study of word embedding methods in topic segmentation // Procedia Computer Science. September 2017. V. 112. Issue C. P. 340–349.
2. *Pan, Weike & Zhong, Erheng & Yang, Qiang.* (2012). Transfer Learning for Text Mining // Mining Text Data. Springer, 2012. P. 223–257. doi: 10.1007/978-1-4614-3223.
3. *Abu Bakr Soliman, Kareem Eissa, Samhaa R. El-Beltagy*, AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP // Procedia Computer Science. 2017. V. 117. P. 256–265. ISSN 1877-0509,.
4. *Mikolov T., Chen K., Corrado G., and Dean J.* Efficient estimation of word representations in vector space // arXiv:1301.3781. 2013.
5. *Zahran M.A., Magooda A., Mahgoub A.Y., Raafat H., Rashwan M., & Atyia A.* Word Representations in Vector Space and their Applications for Arabic // Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2015. Lecture Notes in Computer Science. 2015. V. 9041.
6. *Heuer H.* (2016). Text comparison using word vector representations and dimensionality reduction. CoRR, abs/1607.00534.
7. *Pennington J., Socher R., Manning C.* GloVe: Global Vectors for Word Representation // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014. P. 1532–1543.
8. *Rezaeinia S.M., Ghodsi A., & Rahmani R.* (2017). Improving the Accuracy of Pre-trained Word Embeddings for Sentiment Analysis. CoRR, abs/1711.08609.
9. *Santus E., Wang H., Chersoni E., Zhang Y., & Science C.* A Rank-Based Similarity Metric for Word Embeddings. 2010.
10. *Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin*. Placing search in context: The concept revisited. // Proceedings of the 10th international conference on World Wide Web. ACM, 2001. P. 406–414.

11. *Samer Hassan and Rada Mihalcea*. 2009. Crosslingual semantic relatedness using encyclopedic knowledge // Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. 2009. P. 1192–1201.
12. *Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov*. Enriching word vectors with subword information // Transactions of the Association of Computational Linguistics. 2017. No. 5. P. 135–146.
13. *Application* of Semantic Computing in Cancer on Secondary Data Analysis // International Conference on 3D Digital Imaging and Modeling. Jan. 2018. P. 407–412.
14. *Dahou A., Xiong S., Zhou J., & Haddoud M.H.* Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification // Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. 2016. P. 2418–2427.

1. *Haider S*. (1957) Urdu Word Embeddings // 11th edition of the Language Resources and Evaluation Conference, 7–12, 2018, Japan. P. 964–968.