Word Representations in Vector Space and their Applications for Arabic

Conference Paper · April 2015

DOI: 10.1007/978-3-319-18111-0_32

6 authors, including:

Mohamed A. Zahran
Purdue University
9 PUBLICATIONS **85** CITATIONS
SEE PROFILE

Ahmed Magooda
University of Pittsburgh
14 PUBLICATIONS **77** CITATIONS
SEE PROFILE

Ashraf Y Mahgoub
Purdue University
16 PUBLICATIONS **101** CITATIONS
SEE PROFILE

Hazem Raafat
Kuwait University
62 PUBLICATIONS **621** CITATIONS
SEE PROFILE

Word Representations in Vector Space and their Applications for Arabic

Mohamed A. Zahran[1], Ahmed Magooda[1], Ashraf Y. Mahgoub[1], Hazem Raafat[2], Mohsen Rashwan[3], and Amir Atyia[1]

[1] Computer Engineering Department, Cairo University, Egypt
[2] Computer Science Department, Kuwait University, Kuwait
[3] Electronics and Communications Department, Cairo University, Egypt
{moh.a.zahran,ahmed.ezzat.gawad,ashraf.youssef.mahgoub}@gmail.com,
hazem@cs.ku.edu.kw, mrashwan@rdi-eg.com,
amir@alumni.caltech.edu

Abstract. A lot of work has been done to give the individual words of a certain language adequate representations in vector space so that these representations capture semantic and syntactic properties of the language. In this paper, we compare different techniques to build vectorized space representations for Arabic, and test these models via intrinsic and extrinsic evaluations. Intrinsic evaluation assesses the quality of models using benchmark semantic and syntactic dataset, while extrinsic evaluation assesses the quality of models by their impact on two Natural Language Processing applications: Information retrieval and Short Answer Grading. Finally, we map the Arabic vector space to the English counterpart using Cosine error regression neural network and show that it outperforms standard mean square error regression neural networks in this task.

Keywords: Word Representations, Word Vectors, Word Embeddings, Arabic Natural Language Processing, Arabic Information Retrieval, Arabic Short Answer Grading, Arabic-English vector space mapping, Cosine regression neural network.

1 Introduction

Researchers proposed various techniques to leverage large amount of unlabeled data. One particular method that is adopted by many researchers is representing individual words of a language as vectors in a multidimensional space that capture semantic and syntactic properties of the language. These representations can serve as a fundamental building unit to many Natural Language Processing (NLP) applications. Word representation is a mathematical model representing a word in space, mostly a vector. Each component (dimension) is a feature to this word that can have semantic or syntactic meaning.

Collobert and Weston [1] proposed a unified architecture for natural language processing. They used a deep neural network architecture that is jointly trained using

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 430–443, 2015.
DOI: 10.1007/978-3-319-18111-0_32

back propagation for many tasks: Part of Speech tagging, Chunking, Named Entity Recognition, and Semantic Role Labeling. Individual words are embedded into a d-dimensional vector where each dimension is regarded as a feature. Words representations (embeddings) are stored in a matrix $W \in \mathbb{R}^{d \times |D|}$, where D is a dictionary of all unique words. Look up tables are used for retrieving specific features for words. Sentences are represented using the embeddings of their forming words using a window around the word of interest, which solves the problem of variable length sentences. Mnih and Hinton [2] introduced Hierarchical log-bilinear model (HLBL) that is another form of word representations. For n-gram sentences, it concatenates the embedding of the first $n-1$ words and learns a neural linear model to predict the last word. It is a probabilistic model that uses softmax layer to transform the similarity between the predicted representations with the reference representations to a probability distribution.

Mikolov et al. [3] built a neural language model using a recurrent neural network (RNN) that encode the context word by word and predict the next word. He used the trained network weights as the words representation vectors. The network architecture has input layer with recurrent feed. It has one hidden layer and an output layer. The training procedure iterates over the sentences, each sentence is broken down to words, the input layer receives the current word encoded in a one-hot vector encoding (1-of-N coding, where N is the vocabulary size). The recurrent feed is the previously encoded context history. The length of the recurrent feed depends on the length of the hidden layer. The output layer is a softmax layer that generates a probability distribution over the vocabulary, which means that the length of the output layer equals the size of the vocabulary. The RNN is trained using back propagation to maximize the likelihood of the data using this model.

Turian et al. [4] presented a survey for various work that had been done for representing words in vector space, they also presented yet another neural language model resembles the work of Collobert and Weston to represent words in vector space.

Mikolov et al. [5, 6] proposed two new techniques for building word representation in vector space using log linear models; continuous bag of word (CBOW) and Skipgram (SKIP-G) models. These techniques are based on a neural network architecture with the hidden layer replaced with a simple projection layer to reduce the computational requirements. Although the hidden layer is the main reason that makes neural networks a tempting choice due to their ability to represent data accurately, however by using enough training data the new models can be accurately trained in a much faster setting.

CBOW predicts a pivot word using a window of contextual words around the pivot from the same sentence. The objective of this network architecture is to classify correctly the pivot word given its context by using log linear classifiers.

While most models uses certain context to predict the current word, Skip-gram models on the other hand, uses a pivot word to predict its context by trying to maximize the probability of the contextual word given that pivot word using log linear classifier. It uses the continuous vector representation of the pivot word as an input to the classifier to predict another word in the same context within a certain window. Increasing the context window increases the model accuracy reflected in the quality of the resulting word vectors, but it increases the computation complexity.

Pennington et al. [7] presented yet another technique to learn word representations called "GloVe" for Global Vectors. While CBOW and SKIP-G models can be classified as shallow window based approaches, because they represent a word in vector space as a function of its local context controlled by a window, GloVe on the other hand utilizes the global statistics of word-word co-occurrence in the corpus to be captured by the model. The co-occurrence matrix is used to calculate the probability of $word_i$ to appear in the context of another $word_j$ $P(i|j)$, this probability is postulated to capture the relatedness between these words. For example, the word "solid" is more related to "ice" than to "steam", this can be confirmed by the ratio between P("solid"|"ice") and P("solid"|"steam") to be high. GloVe uses this ratio to encode the relationship between words and tries to find vectorized representation for words that satisfy this ratio, thus the model is built with the objective of learning vector representation for words that captures linear linguistic relationship between them.

2 Building Word Representation for Arabic

Mikolov et al. [5] compared different techniques for building word representation in vector space: Mikolov's RNN embeddings, Collobert and Weston's embeddings, Turian's embeddings and Mnih's embeddings, and showed that the CBOW and SKIP-G models are significantly faster to train with better accuracy. Pennington et al. [7] showed that GloVe performed well compared to CBOW and SKIP-G models in the semantic and syntactic analogy test presented in [5]. Accordingly, we used CBOW, SKIP-G and GloVe models to build a word representation in vector space for Modern Standard Arabic (MSA). To train these models, we collected large amount of raw Arabic texts form these sources:

- Arabic Wikipedia.
- Arabic Gigaword Corpus.
- LDC Arabic newswire.
- Arabic Wiktionary.
- The open parallel corpus [8, 9].
- Combined glosses and definitions for Arabic words in Arabase [10].
- MultiUN; which is collection of translated documents from the United Nations [11].
- OpenSubtitles 2011, 2012, and 2013. They are a collection of movie subtitles [12].
- Raw Quran text [13].
- A corpus of KDE4 localization files [14].
- A collection of translated sentences from Tatoeba [9].
- Khaleej 2004 and Watan 2004 [15].
- BBC and CNN Arabic corpus [16].
- Meedan Arabic corpus [16].
- Ksucorpus; King Saud University Corpus [18].
- A text version of Zad-Almaad book.
- Microsoft crawled Arabic Corpus.

We compiled all these sources together and performed several cleaning and normalization steps to the combined corpus:

– Cleaning noisy characters, tags and removing diacritics.
– Arabic characters normalization: we normalized (آ ,إ,أ) to (ا) and (ة) to (ه).
– Normalizing all numerical digits to the token "NUM".

We formed short phrases from individual words by attaching n-gram tokens together to be treated as a single unit [6]. To form such phrases we choose a frequency threshold above which this n-gram will be treated as a short phrase. For words w_i and w_j and their bigram $w_i w_j$:

$$score(w_i, w_j) = \frac{count(w_i w_j) - \delta}{count(w_i) \times count(w_j)} \quad (1)$$

Bigrams whose score above this threshold will be used as phrases. δ is a discounting factor to prevent the formation of phrases with infrequent words.

The vocabulary size of the compiled corpus is about 6.3 million entries (unigrams and bigrams), and the total number of words is about 5.8 billion. Training these models require choice of some hyper-parameters affecting the resulting vectors:

– **Word vector size**: The vector size is an input parameter. A Couple of hundreds is the recommended choice. This parameter affects the performance of the model, which means it is useful to tune this parameter if the resulting vectors will be used in a specific task.
– **Window**: For CBOW/SKIP-G it refers to the amount of context to consider around the pivot word in training the model, while for GloVe it refers to the maximum distance between two words to be considered in co-occurrence.
– **Sample**: For CBOW/SKIP-G it refers to a threshold for occurrence of words so that words appearing with frequency higher than this threshold will be randomly down-sampled.
– **Hierarchical Softmax (HS)**: For CBOW/SKIP-G, hierarchical Softmax is a computationally efficient approximation of the full softmax used to predict words during training.
– **Negative**: For CBOW/SKIP-G it refers to the number of negative examples in the training.
– **Frequency threshold**: Words appearing with frequency less than this threshold will be discarded.
– **Maximum number of iterations**: For GloVe, it is the number of iterations used to train the model.
– **X_Max**: For GloVe, this parameter is used in a weighting function whose job is to give rare and noisy co-occurrences low weights.

We built three models for Arabic (CBOW, SKIP-G and GloVe)[1]. Table 1 shows the training details for each model.

Table 1. Training configuration parameters used to build the Arabic models

	CBOW	SKIP-G	GloVe
Vector size	300	300	300
Window	5	10	10
Sample	1e-5	1e-5	N/A
HS	No	No	N/A
Negative	10	10	N/A
Freq. thresh.	10	10	10
Phrase thresh.	200	200	200
Max iterations	N/A	N/A	25
X_Max	N/A	N/A	100

3 Vector Quality Assessment

3.1 Intrinsic Evaluation

Having the individual words represented in vector space introduces new thinking strategies in using these representations in word-to-word relations. A relationship between two words can be measured by using a similarity function that maps a pair of word vectors to a real number: $F(v_1, v_2) \rightarrow \mathbb{R}$ This mapping function (similarity measure function) can be Cosine similarity, or Euclidean distance, or Manhattan distance, or any possible similarity measure techniques.

One particular interesting task, is to apply the relationship between a pair of words (e.g. singular/plural, feminization, tense change…) to a third word, this is called "analogy task". Let the first pair of words be w_1 and w_2 and the third word be w_3. Let the relationship between w_1 and w_2 be r_1 then $v(r_1) = v(w_2) - v(w_1)$ where the operator $v(\mathrm{x})$ returns a vector representation of x, and $v(r_1)$ represents a vector in space joining w_1 and w_2. We can apply r_1 to w_3 to give a fourth word w_4 such that $v(w_4) = v(w_3) + v(r_1)$ and so w_3, w_4 have the same relationship as w_1, w_2.

To test the quality of the vectors, Mikolov's analogy test for English vectors [5] is used by translating the test cases manually to Arabic. The test set contains five types of semantic questions, and nine types of syntactic questions. Examples are shown in Table 2. We compared our models to the English skip-gram model [19] and GloVe model [20] using the translated version of the test (Table 3).

The test focuses on calculating the relation between the first pair of words and apply it to a third word, then comparing the fourth word with the predicted word. The predicted word is the word with the highest Cosine similarity score to the predicted vector.

[1] Models are available at: `https://sites.google.com/site/mohazahran/data`

Table 2. A sample of Mikolov's semantic and syntactic analogy test for English and its translation to Arabic

Type of relation ship	Word pair 1				Word pair 2			
Common capital city	Athens	أثينا	Greece	اليونان	Oslo	اوسلو	Norway	النرويج
Man-Woman	brother	شقيق	sister	شقيقة	grand son	حفيد	grand daughter	حفيدة
Superlative	bad	سيء	worst	اسوأ	big	كبير	biggest	اكبر
Plural nouns	bird	طائر	birds	طيور	car	سيارة	cars	سيارات

Table 3. Total accuracy of English models and Arabic models on the test set and its Arabic translation. The first column per model shows the percentage of test cases covered by this model. The second column shows how many of the covered test cases are correct. All numbers in the table are percentages. (Cov. is short for coverage and Acc. is short for accuracy).

Model	**English SKIP-G300**		**English GloVe300**		**Arabic CBOW300**		**Arabic SKIP-G300**		**Arabic GloVe300**	
Training words	300B		840B		5.8B		5.8B		5.8B	
	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.
capital-common-countries	100	94.9	100	**100**	100	94.3	100	93.7	100	92.7
capital-world	100	93.1	100	**95.6**	100	74.7	100	77	100	80.4
currency	100	**37.8**	100	13.2	100	7.7	100	7.9	100	5.7
city-in-state	100	87.2	100	**87.4**	100	32.5	100	32.6	100	36.4
family	100	**95.3**	100	86.8	67.6	46.5	67.6	36.3	67.6	50.3
adjective-to-adverb	100	53.8	100	**65.6**	100	34.2	100	30.1	100	22
opposite	100	**57.6**	100	45.8	80	3.7	80	3.2	80	3.5
comparative	100	**97**	100	96.6	100	73.8	100	67	100	71.5
superlative	100	**95.4**	100	93.7	100	68.9	100	64.6	100	66.9
present-participle	100	96.7	100	**97.4**	93.9	46.1	93.9	42.1	93.9	30.5
nationality-adjective	100	**95.7**	100	89.2	100	49.9	100	55.5	100	44.2
past-tense	100	**93.7**	100	91.9	100	44.7	100	41.6	100	43.5
plural	100	95.8	100	**96.5**	100	56.1	100	56.9	100	57.7
plural-verbs	100	89.5	100	**90.3**	100	80.1	100	75.5	100	72.2
TOTAL	100	**87.4**	100	86.2	98	54.3	98	53.6	98	53.5

A test case is answered correctly only if one of the top five predicted words matches the fourth word, which means that synonyms and semantically close words are considered as mistakes.

By examining the results in Table 3, the Arabic models are trained on significantly smaller corpus compared to English. Having a large corpus is essential to enable the

models to represent the words more accurately, by large corpus we mean a corpus big enough such that each word in the vocabulary is repeated an adequate number of times for the models to represent accurately. However, our models show good performance on test cases whose Arabic translation is unambiguous as translating named entities (countries, capitals, and cities) and thus they are frequent in the corpus. On the other hand, they show low performance on other test cases as in the "opposite" test cases because most of the English words in this test do not have a clear Arabic translation. For example, the word "uncompetitive" has no direct (word-to-word) Arabic translation; the closest translation will be "غير_منافس". These unusual translations are either out of vocabulary or rarely found in the training corpus and thus receive a poor vectorized representation. Low frequent terms explain as well the low performance for the "currency" test cases. For example, the term "الزلوتي" (translated as zloty (Polish currency)) occurred only 63 time.

Another source of errors is the absence of diacritization, which is needed in Arabic to differentiate between words having the same form but with different meanings, however in practice the use of diacritics in Arabic is rare and the Arabic data collected is almost free of diacritics. These results suggests that there should be a tailored test set for Arabic rather than translating the English test cases in order to evaluate the Arabic vectors more accurately.

3.2 Extrinsic Evaluation

Information Retrieval

Many query expansion techniques have been proposed to enhance the performance of text retrieval task, which can be classified into semantic-based and statistical-based expansion techniques. Here, we propose using the Arabic vectors as a semantic expansion technique because the Arabic vectors capture the semantic properties of the language such that semantically close terms are clustered in close proximity in the vector space. Mahgoub et al. [23] proposed query semantic expansion techniques for Arabic information retrieval, the expansion techniques based on various language resources as Wikipedia, Google translate with WordNet, and other various Arabic linguistic resources. We compare our vector expansion technique with their techniques using TREC 2002 the cross-lingual (CLIR) track dataset [24], which contains 50 queries tested against 383,872 documents, we discarded any non-judged documents from our experiments before evaluation. The basic idea is to expand a query term such that these expansions are semantically related to the query, which means the query should act as a sense gauge to the expanded term. A term is expanded using its vector representation to retrieve all other terms in the vector space ordered descendingly by cosine similarity score, while a query is represented by adding up all the vectors of its terms together. The order of possible expansions for a term should be influenced by the query through re-ordering the terms in the expansion list using cosine similarity score with the query vector. In order to avoid bias in re-ordering the expansion list, the term being expanded is not included among the terms forming the query vector. For each query, we allow maximum 50 expansions for all of its terms, such that the number of expansions for each term is inversely

proportional to its frequency, thus allowing less frequent terms to have more expansions [23]. Figure 1 and 2 compare between the impact of using the Arabic vectors as an expansion scheme versus traditional resources as Wikipedia, WordNet translations and other resources using Indiri [25]. The following example in Table 4 shows how the query is used to disambiguate the expansion list of a term (underlined word). The query: "كيف يعامل طلاب الدين في النجف بعد اغتيال صادق الصدر؟" translated as (How are the religion students treated in Nagaf after the assassination of Sadeq Alsadr?). The expanded word (الصدر , Alsadr) has two senses, either "chest" or the name of a person "Alsadr" (which is the correct sense for this query).

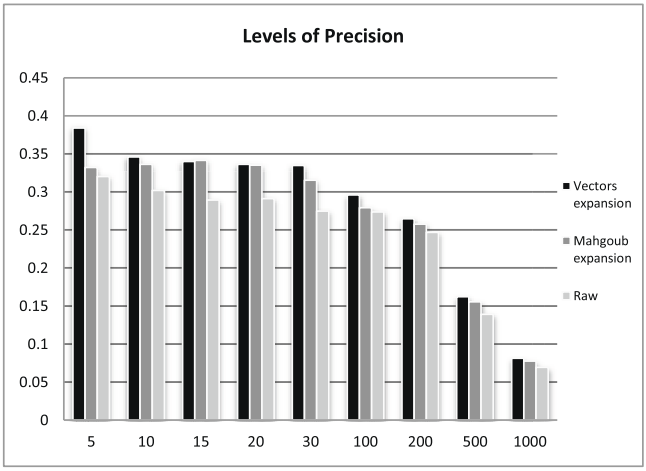

Fig. 1. Levels of precision for expansions using Arabic vectors, Mahgoub expansion using wikipedia, and other resources and raw text matching on TREC 2002

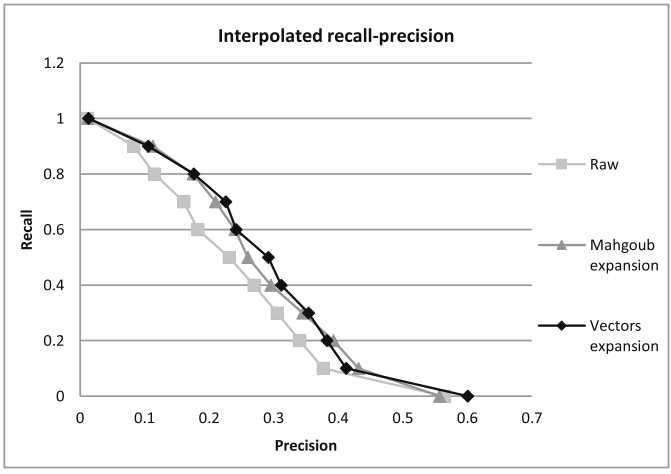

Fig. 2. Recall-precision curve for expansions using Arabic vectors, Mahgoub expansion using wikipedia, and other resources and raw text matching on TREC 2002

Table 4. A comparison between the expansion lists for the word "Alsadr" in a query before and after disambiguation using the query context vector

Before disambiguation		After disambiguation	
Arabic	**English translation**	**Arabic**	**English translation**
والصدر	And the chest	النجف	Alnagaf
للصدر	For the chest	انصار مقتدى	Supporters of Moqtada
بالصدر	By the chest	الصدر ببغداد	Alsadr in Baghdad
البطن	Stomach	مقتدي الصدر	Moqtada Alsadr

Short Answer Grading

Short answer grading is one interesting NLP application to assess how much the Arabic vectors capture semantic and syntactic properties of the language. In short answer grading, given a reference answer and a student answer; it is required to return a grade that represents the correctness of this answer. To employ the vectors in such problem it is essential to transform the grading problem into a sentence-to-sentence similarity measuring task. A sentence can be represented using a combination of the vectors of its words. For example, a simple addition of the word vectors can give a sufficient representation for a sentence in vector space especially for short sentences. Using combinations of different preprocessing steps (lemmatization, stemming …) with various vector based sentence representation schemes (CBOW, SKIP-G, GloVe) will result in a number of features relating a student answer with the reference answer via similarity measures as cosine similarity, these features are fed to an SVM regression module to scale similarity scores to a reasonable grade. Table 5 shows the impact of using the Arabic vectors on an Arabic dataset for short answer grading using root mean square error (RMSE) and Pearson's correlation against inter annotator agreement (IAA). We also report the results of Goma's system discussed in [22] on the equivalent humanly translated English data set.

Table 5. Arabic vectors results in short answer grading using RMSE (the lower the better) and correlation (the higher the better)

	RMSE	**Correlation**
IAA (Arabic and English Dataset)	0.69	0.86
Arabic vectors (Arabic Dataset)	0.95	0.82
Goma's system (Manual English translations data set)	0.75	0.83

4 Arabic-English Vector Space Mapping

Mapping the Arabic vector space to the English vector space is an attractive application especially for Arabic, because Arabic suffers from poor language resources support as compared to English. Mapping the two vector spaces will allow Arabic NLP applications to use the English language support in Arabic NLP domain.

Mikolov et al. [21] used a translation matrix to learn linear transformation between the two vector spaces by minimizing the mean square error between the reference and the predicted vectors, this translation matrix can be regarded as a simple neural network with no hidden layers. Alternatively, we propose training a neural network to learn vector mapping by minimizing the Cosine error instead of minimizing the mean square error (derivations in the appendix). The intuition behind this objective function is the use of Cosine similarity score in literature as the default measure to assess the similarity between two word vectors [5, 6, 7, 21], which means there is a mismatch between the objective function (mean square error) and the similarity metric (Cosine similarity score). We used an English-Arabic dictionary to train the neural network by retrieving vectors corresponding to the dictionary's entries to form parallel training data. Each entry consists of an Arabic word with a list of possible English translations totaling 8,444 entries divided into 5,872 entries for training, 1,254 for validation and 1,318 for testing. The training and validation entries are expanded such that an Arabic word form a different entry with each of its possible English translations, resulting in 27,089 entries for training and 5,718 entries for validation. We used vectors from the best scoring models in Table 3, CBOW300 for Arabic and SKIP-G300 for English.

Two simple neural networks are constructed with 300 neurons for both input and output layers with no hidden layers. Both networks have the same architecture, parameters and initial weights, the only difference is the objective functions, the first optimizes for Cosine error, the second optimizes for mean square error and both are trained using backpropagation. Some translation examples from the test set using the Cosine neural network are shown in Table 6.

Table 7 compares between the two networks using three measures, NDCG, recall and accuracy for the test set. For each instance in the test set, we use the predicted English vector to retrieve the top k English translations using Cosine similarity. The NDCG for test sample j using k predicted translations:

$$NDCG_{\mathrm{j}} = \frac{match(1) + \sum_{\mathrm{i}=2}^{\mathrm{k}} \frac{match(i)}{log_2 i}}{1 + \sum_{\mathrm{i}=2}^{\mathrm{k}} \frac{1}{log_2 i}} \tag{2}$$

Where the function $match$ takes a predicted translation and checks if it matches one of the possible reference English translations with no accounting for synonyms, semantically close terms and even morphological variations.

$$match(\mathrm{x}) = \begin{cases} 1, & \text{if } x \text{ is a match.} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The overall NDCG for M test sentences using k predicted translations is:

$$NDCG = \frac{\sum_{\mathrm{j}=1}^{\mathrm{M}} NDCG_{\mathrm{j}}}{M} \tag{4}$$

While the NDCG accounts for the rank k at which a match happens, the recall and accuracy on the other hand takes no regard to the rank. The accuracy calculates how many test samples were translated correctly by matching at least one of the k

predicted translations with one of the reference translations, while the recall calculates how many reference translations were covered by the predicted translations. The recall for a test sample j with l reference translations and k predicted translations:

$$Recall_{\mathrm{j}} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{k}} match(i)}{Min(k,l)} \tag{5}$$

The overall recall for M test sentences using k predicted translations is:

$$Recall = \frac{\sum_{\mathrm{j}=1}^{\mathrm{M}} Recall_{\mathrm{j}}}{M} \tag{6}$$

Table 6. Translation examples using Cosine neural network

Arabic Word	English translations using Cosine neural network
عصور	epochs,epoch,millenniums,millennia,eras,era
علم_الفلك	astronomy,cosmology,quantum_physics,astrology,science
الكونجرس	parliament,congress,legislature,senate,vote,Congress
يتسلل	infiltrate,sneak,penetrate,seep,slither,creep

Table 7. A comparison between optimizing for Cosine error versus mean square error using NDCG, Recall and Accuracy

k	Cos (NDCG)	MSE (NDCG)	Cos (Recall)	MSE (Recall)	Cos (Accuracy)	MSE (Accuracy)
1	27.1%	25.9%	27.1%	25.9%	27.1%	25.9%
2	19.3%	19.1%	22%	21.7%	34.7%	34.4%
3	16.8%	16.5%	20.8%	20.1%	38.8%	38.5%
4	15.2%	14.9%	20.1%	19.4%	41.4%	40.4%
5	14%	13.8%	20.2%	19.6%	43.6%	43.1%
6	13.1%	12.8%	20.6%	19.6%	45.8%	44.8%
7	12.4%	12.2%	21.2%	20.3%	47.8%	46.7%
8	11.8%	11.5%	21.8%	20.8%	49.3%	48.1%
9	11.2%	11%	22.3%	21.1%	50.2%	49%
10	10.7%	10.5%	22.6%	21.7%	50.8%	50.4%

Another way to assess the quality of the neural network is to compare the translated vectors with the native English vectors in a practical NLP task as in short answer grading. The idea is having an Arabic datasets for short answer grading and its human translation to English. Given an Arabic student answer a_{s} and its Arabic reference answer a_{r} and their corresponding student and reference English answers e_{s} and e_{r} respectively. First, we assign a grade to the English answers using the English vectors following the same ideas discussed in the previous section. Second, using the proposed neural network we translate the two Arabic answer vectors to English vectors (e'_{s} and e'_{r}) and assign a grade to them using the translated vectors. Finally, we compare between the Arabic vectors on the Arabic data, the translated vectors on the English data, and the English vectors on English data using the Pearson's

correlation between the predicted grades of each system with the reference grade as shown in Table 8.

Table 8. Correlation between predicted grades and reference grades for Native Arabic, Translated English and Native English for short answer grading using word vectors

Model	Correlation
Arabic vectors & Arabic data	0.79
Translated English Vectors & Human translated English data	0.75
English vectors & Human translated English data	0.76

These results show that the translated english vectors using the neural network performs remarkably closer to the native english vectors on the same data set (Human translated english data set) than to Arabic vectors on Arabic data.

5 Conclusion and Future Work

In this paper, we compared between different models for building continuous representation in vector space for Arabic and tested these vectors via intrinsic and extrinsic evaluations. In intrinsic evaluation, we used the analogy task to test the vectors' capability to capture semantic and syntactic properties for Arabic. While in extrinsic evaluations, we employed the vectors in two NLP applications: query expansion for information retrieval and short answer grading. For the query expansion, the Arabic vectors enhanced the retrieval process slightly better than other semantic expansion techniques, while for short answer grading, Arabic vectors made it possible to evaluate short answer grades for Arabic dataset without the need for Arabic-English translations.

We built a neural network to map Arabic vectors to English vectors and showed that minimizing for cosine error outperforms the standard mean square error minimization for word-to-word similarity using cosine score, which means that the objective function of the training procedure should match the similarity measure used. Using the proposed neural network, we succeed to achieve humanly translated-like results for short answer grading task. Many extensions can be related to the work presented here, starting with increasing the raw Arabic data used to train the vectorized representations. It will also be useful to have an analogy test built specifically for Arabic rather than the manual translation of the English test. We showed a simple technique for word sense disambiguation in the context of query expansion using the Arabic vectors, yet even more sophisticated techniques can be developed. Using deep neural networks, it can be possible to learn complex relations to map the Arabic and English vector spaces more accurately.

Acknowledgments. We would like to thank Microsoft Advanced Technology Lab in Cairo, Egypt for supporting this work with data and computing resources.

References

1. Collobert, R., Weston, J.: A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, pp. 160–167 (2008)
2. Mnih, A., Hinton, G.: A Scalable Hierarchical Distributed Language Model. In: NIPS: Proceedings of Neural Information Processing Systems, Vancouver, B.C, Canada, pp. 1081–1088 (2009)
3. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: NAACL-HLT: Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751 (2013)
4. Turian, J., Ratinov, L., Bengio, Y.: Word representations: A simple and general method for semi-supervised learning. In: ACL: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394 (2010)
5. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: ICLR: Proceeding of the International Conference on Learning Representations Workshop Track, Arizona, USA, pp. 1301–3781 (2013)
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representation of Words and Phrases and their Compositionality. In: NIPS: Proceedings of Neural Information Processing Systems Nevada, United States, pp. 3111–3119 (2013)
7. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: EMNLP: Proceeding of the Empirical Methods in Natural Language Processing, Doha, Qatar, pp. 1532–1543 (2014)
8. http://opus.lingfil.uu.se/ (accessed January 29, 2015)
9. Tiedemann, J.: Parallel Data, Tools and Interfaces in OPUS. In: LREC: Proceedings of the 8th International Conference on Language Resources and Evaluation, Istanbul, Turkey, pp. 2214–2218 (2012)
10. Raafat, H., Zahran, M., Rashwan, M.: Arabase A Database Combining Different Arabic Resources with Lexical and Semantic Information. In: Proceeding of KDIR is part of IC3K, The International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Portugal, pp. 233–240 (2013)
11. Eisele, A., Chen, Y.: MultiUN: A Multilingual corpus from United Nation Documents. In: LREC: Proceeding of the International Conference on Language Resources and Evaluation, Valletta, Malta, pp. 17–23 (2010)
12. http://www.opensubtitles.org/ (accessed January 29, 2015)
13. http://tanzil.net/download/ (accessed January 29, 2015)
14. Tiedemann, J.: News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In (RANLP): Recent Advances in Natural Language Processing, pp. 237–248. John Benjamins, Amsterdam (2009)
15. https://sites.google.com/site/mouradabbas9/corpora (accessed January 29, 2015)
16. Saad, M.K., Ashour, W.: OSAC: Open Source Arabic Corpus. In: EEECS: the 6th International Symposium on Electrical and Electronics Engineering and Computer Science, European University of Lefke, Cyprus, vol. 10 (2010)
17. https://github.com/anastaw/Meedan-Memory (accessed January 29, 2015)
18. http://ksucorpus.ksu.edu.sa/ar/ (accessed January 29, 2015)
19. https://code.google.com/p/word2vec/ (accessed January 29, 2015)
20. http://nlp.stanford.edu/projects/glove/ (accessed January 29, 2015)

21. Mikolov, T., Le, V.Q., Sutskever, I.: Exploiting Similarities among Languages for Machine Translation. In: arXiv, 1309-4168 (2013)
22. Gomaa, W.H., Fahmy, A.A.: Automatic scoring for answers to Arabic test questions. Computer Speech & Language, 833–857 (2014)
23. Mahgoub, Y.A., Rashwan, A.M., Raafat, H., Zahran, A.M., Fayek, B.M.: Semantic Query Expansion for Arabic Information Retrieval. In: EMNLP: The Arabic Natural Language Processing Workshop, Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, pp. 87–92 (2014)
24. Oard, D.W., Gey, F.C.: The TREC 2002 Arabic/English CLIR Track. In: TREC (2002)
25. `http://sourceforge.net/p/lemur/wiki/Indri/` (accessed January 31, 2015)

Appendix

The objective function is to maximize the cosine similarity between the predicted vector (y) and the reference vector (d). This is equivalent to:

$$Minimize\ E = 1 - \cos(y, d) = 1 - \frac{y.d}{|y||d|}$$

Let the activation function be $y_i^z = F(x_i^z)$. The superscript denotes the layer number, and the subscript denotes the input number. The notation $l(z)$ refers to the number of neurons in the layer z. The derivative of the error function E with respect to weights at layer z for the training sample m:

$$\frac{\partial E}{\partial w_{IJ}^z} = \frac{\partial E}{\partial y_I^{z+1}} \times \frac{\partial y_I^{z+1}}{\partial x_I^{z+1}} \times \frac{\partial x_I^{z+1}}{\partial w_{IJ}^z} = \delta_I^{z+1} \times \frac{\partial x_I^{z+1}}{\partial w_{IJ}^z} \tag{7}$$

$$where,\ \delta_I^{z+1} = \frac{\partial E}{\partial y_I^{z+1}} \times \frac{\partial y_I^{z+1}}{\partial x_I^{z+1}} \tag{8}$$

$$\frac{\partial E}{\partial y_I^{z+1}} = \frac{(y.d)y_I^{z+1} - d_I^{z+1}|y|^2}{|d||y|^3} \tag{9}$$

$$y_I^{z+1} = f(x_I^{z+1})$$

$$f(x) = 1.7159 \tanh\left(\frac{2}{3}x\right)$$
$$\frac{\partial y_I^{z+1}}{\partial x_I^{z+1}} = f'(x_I^{z+1}) = \left(1.7159 \times \frac{2}{3}\right)\left(1 - \left(\frac{y_I^{z+1}}{1.7159}\right)^2\right) \tag{10}$$

$$x_I^{z+1} = \sum_{j=1}^{l(z)} w_{ij}^z\, y_j^z \quad \Rightarrow \quad \frac{\partial x_I^{z+1}}{\partial w_{IJ}^z} = y_J^z \tag{11}$$

Finally $\frac{\partial E}{\partial w_{IJ}^z}$ is calculated by substituting from (8), (9), (10) and (11) in (7).