En-Ar Bilingual word Embeddings without Word Alignment: Factors Effects

Taghreed Alqaisi
University of York, UK
Taibah University, Saudi Arabia
ta808@york.ac.uk

Simon O'Keefe
University of York, UK
simon.okeefe@york.ac.uk

Abstract

This paper introduces the first attempt to investigate morphological segmentation on En-Ar bilingual word embeddings using bilingual word embeddings model without word alignment (BilBOWA). We investigate the effect of sentence length and embedding size on the learning process. Our experiment shows that using the D3 segmentation scheme improves the accuracy of learning bilingual word embeddings upto 10 percentage points comparing to the ATB and D0 schemes in all different training settings.

1 Introduction

In the last decade, neural networks (NN) have attracted many researchers attention and showed very promising results in many natural language processing (NLP) tasks. Many models have been introduced including: semantics and question answering (Bowman et al., 2015; Sukhbaatar et al., 2015; Hermann et al., 2015), Machine Translation (MT) (Sutskever et al., 2014; Bahdanau et al., 2015), parsing (Kong et al., 2015; Lewis et al., 2016) and many works in word embeddings have been reported. Word embedding is one of the most important NLP tasks due to its ability to capture the semantic similarities between words.

The main idea behind learning word embeddings is to transform words from discrete space into a continuous vector space of features that capture their syntactic and semantic information. In other words, words having similar meaning should have similar vectors. This similarity can be measured using different distance methods such as cosine similarity and Euclidean distance.

Now a days, many word embedding models have been introduced and show a significant improvement in different NLP tasks; language modelling (Mikolov et al., 2010; Mikolov and Zweig, 2012; Shi et al., 2013), MT (Cho et al., 2014; Bahdanau et al., 2015; Luong et al., 2015b), named entity recognition (Lample et al., 2016), document classification and sentiment analysis (dos Santos and Gatti, 2014; Kim, 2014; Severyn and Moschitti, 2015) etc. Word embeddings can be classified, based on the objective function that needs to be learnt, into two main categories. Firstly, Monolingual word embedding, which is the process of learning similar word representations for similar word meaning in the same language. Secondly, Bilingual/cross-lingual approaches, which is the process of learning similar words among languages.

In this paper, we investigate the effect of different Arabic segmentation schemes, sentence length and embedding sizes on learning Arabic-English (Ar-En) Bilingual word embeddings. The experiments show a noticeable accuracy change using different training settings. Firstly, we give an overview of some related recent works on bilingual word embeddings in Section 2. Section 3 gives a brief introduction to the Arabic language, and it describes the details of Arabic language morphological complex and preprocessing techniques. Next is the experiment section that contains a description of the model architecture, training dataset, preprocessing settings and training hyper-parameters. The evaluation section presents the evaluation methods used as well as discussing the trained models' evaluation results. Finally, we conclude this work outcomes in Section 6.

2 Related Work

Bilingual or cross-lingual word embedding is the process of learning the semantic similarity across two or more languages word embeddings using two or more corpora. Many successful models have been introduced and use different model

Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 97–107
Florence, Italy, August 1, 2019.

architectures and training corpora with different alignment levels to learn bilingual word embeddings.

Firstly, at word-level alignment, Luong et al. (2015a) extend the skip-gram model to learn efficient bilingual word embeddings. Also, at phrase-level, a bilingually-constrained phrase embeddings (BRAE) model learns source-target phrase embeddings by minimising the semantic distance between translation equivalents and maximising the semantic distance between non-translation equivalents (Zhang et al., 2014). Su et al. (2015) extend the BRAE model by introducing a "bilingual correspondence recursive autoencoder" (BCorrRAE) model, which incorporates word alignment to learn bilingual phrase embeddings by capturing different levels of their semantic relations. After that, Zhang et al. (2016) introduce a Bidimensional attention-based recursive autoencoder (BattRAE) model to learn bilingual phrase embeddings by integrating source-target interactions at different levels of granularity using attention-based models.

Using a sentence-aligned corpus, Gouws et al. (2015); Coulmance et al. (2015) introduce BilBOW and Trans-gram methods to learn and align word embeddings without word alignment. With a document level aligned corpus, Vulic and Moens (2015) present a model that learns bilingual word embeddings from non-parallel document-aligned data without using translation pairs. In addition, Mogadala and Rettinger (2016) introduce a Bilingual paRAgraph VEctors (BRAVE) model that learns bilingual embeddings from either a sentence-aligned parallel corpus or label-aligned non-parallel document corpus. Vulic and Moens (2015) introduce a model that learns multilingual (two or more languages) word embeddings using document-aligned comparable data.

In the literature we found three different bilingual embedding approaches: monolingual mapping, parallel corpus and joint optimisation approaches. In monolingual mapping, word representations are learnt separately for each language using large monolingual corpuses. Then, using word translation pairs, the model learns a transformation matrix that maps word representation from one language to the other (Ruder, 2017). Parallel corpus models require either word-level (Xiao and Guo, 2014) or sentence level alignments (Hermann and Blunsom, 2013; Lauly et al., 2014; Gouws et al., 2015). These models aim to have same word/sentence representations for equivalence translations.

Finally, in the joint optimisation method, the monolingual and cross-lingual objectives are optimised jointly (Gouws et al., 2015; Coulmance et al., 2015). Gouws et al. (2015) propose a bilingual bag-of-words without word alignment model (BilBOWA) that uses a skip-gram model as the monolingual objective and jointly learns the bilingual embeddings by minimising the distance between aligned sentences, by assuming that each word in the source sentence is aligned to all words in the target sentence. This model shows success in translation and document classification tasks on ES-En and En-De languages pairs.

In the context of the Arabic language, no prior work has investigated learning bilingual word embeddings to such a morphologically complex language. Thus, in this work, due to the speed and success of BilBOWA models on learning bilingual words embeddings without word alignments, we train the model on a language with a different structure namely Arabic, in order to investigate the effects of complex language morphology in learning bilingual word embeddings.

3 Arabic language

The Arabic language still presents a challenge in MT as it is the official language of 22 countries from the Arabic Gulf to Morocco and varies between countries or regions in the same country. The Arabic language has many forms including: Classical Arabic, Modern Standard Arabic (MSA) and Arabic dialects. MSA, which is based on classical Arabic syntactically, morphologically and phonologically, is written and spoken in news broadcasts, while Arabic dialects are the true native language forms for daily communications (Habash., 2010). In this research we have focused on MSA as the most accessible form.

3.1 Arabic Morphology

The Arabic language is a complex language morphologically and syntactically (Monem et al., 2008). Much work has been done in Arabic NLP but the problems that are caused by the rich morphology of Arabic still exist. We discuss some of the complexity below.

3.1.1 Arabic Language Words

As with many languages, Arabic words can have affixations (prefix, suffix) and can turn the verb to a noun and vice versa. The prefix usually indicates the tense as well as gender, while the suffix indicates plural and the gender too (Khemakhem et al., 2010). So one Arabic word can translate into up to three English words. As a result, the meaning of an Arabic word can be changed when changing its affixation. There is a lot of affixation in the Arabic language and it has been considered as an issue in many NLP tasks, researchers have handled Arabic affixes using a morphological analysis to improve the Arabic NLP (Hatem et al., 2011).

Another issue is non- or short-vowelled Arabic words. The same word can have different meanings depending on its diacritisation and these diacritisations are not usually written. However, the state of the art tool MADAMIRA (See Subsection 3.2) can handle this issue by producing a diacritised corpus.

3.1.2 Arabic Language Sentence Structure

The Arabic language has two types of sentences: nominal (starts with a name) and verbal (starts with a verb). The Arabic and English languages are very different from a structural point of view. One of the main differences between Arabic and English is the order of words. As with other languages, Arabic sentences are built of verb, subject and object. And usually, an Arabic sentence is post-verbal (VSO) so the verb comes first and then the subject is followed by the object. However, it is possible to be pre-verbal (SVO) as the English language is, but it is not always preferred (Elming and Habash, 2009). In both cases, VSO or SVO, an Arabic sentence is flexible with its verb position. However, the subject needs to come before the object, except in passive sentences in which it can be either before its subject or without its subject. Secondly, in Arabic, the adjective always comes after its noun, which is not the case in English. So a reordering rule should move the object of an Arabic sentence to the right of the adjective. Finally, indicating possession and compounding in Arabic is called Idafa. Idafa consists of one or more nouns that have been defined by the following noun (Elming and Habash, 2009).

3.2 Arabic language Preprocessing

In pre-processing, lots of work has studied the impact of morphological pre-processing techniques on statistical machine translation (SMT) quality. Researchers agree on the importance of morphological and syntactic pre-processing in MT in terms of reducing both sparsity and the number of "out of vocabulary" words (OOV) (Khemakhem et al., 2010; El Kholy and Habash, 2012). At pre-processing level, current research focuses on two main pre-processing techniques: word segmentation and word pre-ordering. Many tools have been introduced: AMIRA (Soudi et al., 2007), MADA (Habash and Rambow, 2005), MADA+TOKAN (Habash et al., 2009), Farasa (Abdelali et al., 2016), AlKhalil Morpho (Boudchichea et al., 2017) and MADAMIRA (Pasha et al., 2014).

MADAMIRA is a tool for morphological analysis and the disambiguation of Arabic including normalisation, lemmatisation and tokenisation. It can tokenise the input text with 11 different tokenisation schemes and normalise Alif and Ya characters. MADAMIRA has been developed the same as MADA to accept two input forms: MSA and Egyptian Arabic (EGY). Pasha et al. (2014) have pointed out that MADAMIRA has outperformed both AMIRA and MADA and is the state of the art.

In this work, as word order and language modelling don't matter, we only applied segmentation and orthographic normalisation in the training datasets.

3.2.1 Word Segmentation

Word segmentation has been considered the same process as tokenisation in the Arabic language. It is one of many techniques that have been proposed to reduce morphological differences between languages such as Arabic and English (Akeel and B. Mishra, 2014). Many tokenisation schemes have been introduced for Arabic and have been successfully applied. Many researchers have studied the positive effect of morphological pre-processing on En-Ar SMT. El Kholy and Habash (2012) found that tokenisation and orthographic normalisation improves the performance on SMT, especially when translating from a rich into a poor morphological language. Their work also shows that lemma-based word alignment improves the translation quality in En-Ar SMT.

Many researchers have studied the effect of different segmentation schemes in MT quality on both En-Ar and Ar-En SMT. For example, Habash and Sadat (2006) show in their work that rule-based segmentation improves the translation qual-

ity for a medium-sized corpus but the benefit of word segmentation decreases when the corpus size is increased. Other researchers Al-Haj and Lavie (2012) believe that tokenisation schemes with more splitting lead to a decrease in the OOV rate. On the other hand, increasing the number of token types can affect word alignment, translation model and language model negatively as predicting these tokens correctly becomes more complex (El Kholy and Habash, 2012).

Researchers consider the Arabic tokenisation process one of the main solutions helping to decrease Arabic ambiguities in MT. Many researchers have introduced different rule-base segmentation schemes (See Table **??**in Appendix). Some of these schemes are used in En-Ar SMT and they show the importance of word segmentation as a pre-processing step to minimise the differences between Arabic and English as well as its effects on SMT quality. The work of (Badr et al., 2008) shows a significant improvement in En-Ar SMT performance when combining segmentation with pre-processing and post-processing steps for small training data. Al-Haj and Lavie (2012); El Kholy and Habash (2012) have studied the effect of different segmentation schemes in En-Ar phrase-based machine translation (PBMT). Al-Haj and Lavie (2012), in contrast to the previous work, investigate the effect of different segmentation schemes on a very large amount of training data of at least 150M words. Their work shows that simple segmentation performs better than complex segmentation as the complex segmentation has a negative effect by increasing the size of the phrase table.

3.2.2 Orthographic Normalization

Orthographic normalisation is an important process at the pre-processing stage. (El Kholy and Habash, 2012) have introduced two schemes of orthographic normalisation: enriched Arabic (ENR) and reduced Arabic (RED). RED is used at the pre-processing level to convert all Hamzat-Alif forms to bare Alif (taking out Hamza) and Alif-Maqsura forms to Ya (add dots). ENR selects the correct Alif and Ya form in order to generate the correct Arabic form at the post-processing level.

4 Experiments

The aim of this set of experiments is to evaluate the effect of sentence length on the process of learning bilingual embeddings using different segmentation schemes.

4.1 Model Architecture

Bilingual Bag-of-Words without Alignment (BilBOWA): BilBOWA, introduced in (Gouws et al., 2015), is a simple efficient model to learn bilingual distributed word representations without word alignment. Instead, it assumes each word in the source language sentence is aligned to every word in the target language sentence and vice versa by using a sentence level aligned corpus. This feature is an advantage of this model as the word alignment process is very time consuming.

In the BilBOWA model, as has been mentioned, both monolingual and bilingual objective functions are learnt jointly. The monolingual words representations are obtained by training word2vec using a skip-gram model using negative sampling approach by (Mikolov et al., 2013b).The bilingual objective aims to minimise the distance between source and target sentences by minimising the means of word representations in each aligned sentences pair.

4.1.1 Monolingual Features

Instead of using Softmax, Gouws et al. (2015) implemented Word2vec model using a simplified version of a noise-contrastive approach: negative sampling training objective modified by (Mikolov et al., 2013a) as:

$$\log p(w|c) = \log \sigma(v_w'^T v_{cp}) + \sum_{i=k}^{K} E_{w_i} \sim P_n(w)[\log \sigma(-v_w'^T v_{cn})] \quad (1)$$

Where v_w is word vector and v_{cp}, v_{cn} positive and negative context vectors respectively and K is the number of negative samples.

This approach learns high-quality monolingual features and speeds up the computation process in this model architecture by converting multinomial classification problem to a binary classification problem (Mikolov et al., 2013a; Gouws et al., 2015).

4.1.2 Bilingual/Cross-lingual Features

Gouws et al. (2015) believe that as with the importance of learning the relations between words in the same language, it is also very important to

learn words representations that capture the relations among languages. Therefore, the BilBOWA model learns word representations by updating the shared embeddings jointly for both monolingual and bilingual objectives. With the cross-lingual objective, this model minimises the loss between sentence representation pairs computed as the mean of bag-of-words of the parallel corpus.
The bilingual objective is defined as:

$$\Omega = ||\frac{1}{m}\sum_{i=1}^{m} r_i - \frac{1}{n}\sum_{j=1}^{n} r_j||^2 \qquad (2)$$

Where m and n are the number of words in the source and target language , and r_i and r_j is a word representation for each language respectively.

4.2 Data

In this paper, we used WIT3, Web Inventory of Transcribed and Translated Talks, plain MSA Arabic and English language parallel corpus (WIT3, 2012). The dataset has been divided into a 50K monolingual-dataset and a 24K bilingual-dataset to train the monolingual and bilingual objectives. After preprocessing (See Section 4.3), two different bilingual training datasets have been extracted based on sentence length: 5-10 and 17-80 tokens sentence length. Giving the distribution of sentence length in the corpus, these sentence length (5-10 and 17-80 tokens) give us a reasonable size of dataset and distinction between short and long sentences. For the test dataset, similarly to (Gouws et al., 2015), we created a set of 3K words by extracting the most common words in the training datasets. Then, the extracted words have been translated word by word translation using Google translator (In line with common practice in the field) to create a word-based dictionary.

Datasets	5-10	17-80	Mono50K-data
Arabic ATB	195985	901013	902307
English ATB	153111	551508	554338
Arabic D3	187612	975221	1033188
English D3	132687	520190	553414
Arabic D0	190854	773826	771512
English D0	158577	557664	553414

Table 1: Number of tokens in training Datasets with different segmentations schemes. Note that preprocessing changes sentence length, and different methods therefore produce different datasets

4.3 Preprocessing

Both sides of the dataset (English and Arabic), are tokenised, cleaned, normalised and stop-words have been removed. For Arabic, a morphological segmentation process is applied in order to minimise the differences between each En and Ar language pair.

Literature shows many different segmentation schemes for Arabic language (See Table 2 for more details). We use MADAMIRA a state of the art Arabic morphological analyzer (Pasha et al., 2014) for Arabic tokenisation, segmentation and normalisation processes in this work. Three different training datasets with different segmentation schemes are generated: D0, ATB, And D3 (For example: See Table 3). For English, we used the Moses toolkit (Koehn et al., 2007) for tokenising the English dataset and cleaning both sides.

4.4 Training

After preprocessing, we train a BilBOWA model using six preprocessed datasets with different settings: two sentence-length (5-10 and 17-80) and three different segmentation schemes that give a range of amount of segmentations from no segmentation to more complex segmentation (D0, ATB and D3). The trained models produce different embedding sizes: (100D , 200D and 300D). As mentioned in (Gouws et al., 2015), the Asynchronous Stochastic Gradient Descent (ASGD) algorithm has been used to train the model and updating all parameters for each objective function (monolingual and bilingual threads) with a learning rate of 0.1 with linear decay. The number of negative samples is set to NS=5 for the skip-gram negative sampling objectives as we examined NS=15 and it didn't show an improvement in our language pair. All trained models has been trained on a machine that is equipped with four Quad-Core AMD Opteron processors running at 2.3 GHz and 128 GB of RAM. The training process takes up to 30 minutes depends on the model's embeddings size and sentence length.

5 Evaluation

As with word-level bilingual word embeddings (BWEs), similarly to (Gouws et al., 2015), the trained BWEs has been evaluated on a word translation task using *Edit Distance*, used by (Mikolov et al., 2013a). First, we extracted the most frequent 3K words from the Ar-En dataset

D0/UT	No tokenization.
D1	Separates the conjunction proclitics.
D2	D1 + Separates prepositional clitics and particles.
D3/S1	Separates all clitics including the definite article and the pronominal enclitics.
S0	Splitting off the conjunction proclitic w+.
S2	Same as S1 but all proclitics are put together in a single proclitics cluster.
ATB	The Arabic Treebank is splitting the word into affixes.
S3	Splits off all clitics from the (CONJ+) class and all suffixes form the (+PRON)class. In addition to splitting of all clitics of (PART+) class except s+ prefix.
S0PR	S0 + splitting off all sufixes from (+PRON) class.
S4	S3 + splitting off the s+ clitics.
S5	Splits off all possible clitics (CONJ, PART, DET and PRON) classes.
S4SF	S4 + the (+PRON) clitics.
S5SF	S5 + the (+PRON) clitics.
S5ST	S5 + prefixes concatenated into one prefix.
S3T	S3 + prefixes concatenated into one prefix.
DIAC	One of MADA features that add diactresation to Arabic text.

Table 2: Existing tokenisation schemes for Arabic (Al-Haj and Lavie, 2012)

D0	wtAvrt Tfwlty bAlryf ldrjp qd AEjz En $rHhA kmA tmyzt bAlfkr bmA yfwq twqEAtkm .
D3	wtAvrt Tfwlp +y b+ Al+ ryf l+ drjp qd AEjz En $rH +hA k+ mA tmyzt b+ Al+ fkr b+ mA yfwq twqEAt +km .
ATB	wtAvrt Tfwlp +y b+ Alryf l+ drjp qd AEjz En $rH +hA k+ mA tmyzt b+ Alfkr b+ mA yfwq twqEAt +km .

Table 3: The used Arabic tokenisation schemes examples

and preprocessed them similarly to the training dataset. Then, we translate the extracted words using Google translator to create a dictionary. After that, for Arabic as source and English as a target, we compute the distances between vectors in order to extract the embeddings of the k nearest neighbours for a given source word embedding in the target word embeddings.

After computing the similarity, the top k nearest neighbours (for k=1, 3, 5) have been selected to compute the accuracy among the test dataset, which consists of 3000 words and their translations. Then we computed the accuracy of 10 runs randomly selecting 500 source words and their k nearest neighbours as:

$$Acc = \frac{ct}{T} \tag{3}$$

Where ct is the number of correct translations and T is the number of all test samples.

The accuracy is computed for all experiments with all different settings: sentence-length, embeddings size and segmentation schemes and the results are discussed below. We also took into account the observed variance in considering significance of the observed differences in performance.

5.1 Results And Discussion

After computing each run accuracy, we computed the model final performance by computing the mean of the output values for each experiment as shown in Tables 4, 5 and 6. Based on the observed accuracies and using sample/population standard deviation (SSD and PSD) to indicate significant differences (See Tables 4, 5 and 6), our results cover three aspects of the problem:

- **Embeddings size:**
  Training the model on different embeddings sizes (100D, 200D and 300D) showed that, for more complex language pairs, increasing the vector size allowed the model to capture more information and lead to learn better Ar-En BWEs. Both Figures 1 and 2 show an increase in accuracy when the size of word representation is increased.

En-Ar 100D	k=1			k=3			k=5		
5-10	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	17.86	1.82	1.73	23.45	1.89	1.79	28.31	2.01	1.91
D0	15.32	0.97	0.92	18.82	3.85	3.65	20.99	2.44	2.31
D3	18.98	1.87	1.78	26.04	2.28	2.17	28.32	2.62	2.49
17-80	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	17.88	1.32	1.25	23.85	1.86	1.77	27.49	1.24	1.17
D0	16.14	1.76	1.67	19.99	1.74	1.65	21.94	2.37	2.25
D3	22.92	1.09	1.04	31.59	2.6	2.5	33.82	1.9	1.8

Table 4: 100D Models' Results

En-Ar 200D	k=1			k=3			k=5		
5-10	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	25.86	1.23	1.16	33.14	1.53	1.46	37.6	2.46	2.33
D0	21.19	1.65	1.56	27.71	2.12	2.01	30.28	1.81	1.72
D3	26.34	2.58	2.44	34.74	1.53	1.45	37.02	2.03	1.92
17-80	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	22.89	2.18	2.07	30.19	2.66	2.52	31.6	1.38	1.31
D0	22.22	2.17	2.06	28.87	1.67	1.58	31.32	1.55	1.47
D3	32.83	1.48	1.41	41.06	2.35	2.23	43.9	1.39	1.32

Table 5: 200D Models' Results

En-Ar 300D	k=1			k=3			k=5		
5-10	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	31.12	1.96	1.86	39.94	3.4	3.29	42.72	1.63	1.55
D0	26.88	1.65	1.56	33.99	1.10	1.04	37.67	2.63	2.50
D3	31.8	1.86	1.77	42.48	1.93	1.84	44.74	1.61	1.53
17-80	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	33.81	3.29	3.12	43.73	2.76	2.62	46.04	1.92	1.83
D0	30.38	2.09	1.98	37.09	1.73	1.64	40.39	1.98	1.88
D3	40.38	1.99	1.89	49.16	1.54	1.46	51.25	2.94	2.79

Table 6: 300D Models' Results

- **Sentence length:**
  Comparing results from using short and long sentences, our results shows that long sentences (which increase the number of words "tokens") outperformed the short sentences in 300D embeddings size models using all three different segmentation schemes. While short sentences perform better only with 200D embeddings size and ATB segmentation scheme trained model. Thus, long sentences with 300D embeddings size allow trained models to capture more information and learn better bilingual word representations.

- **Segmentation schemes:**
  Different segmentation schemes show different levels of learning BWEs. D3, which is more segmentation (breaking the word into more tokens: split all clitics), has a significant effect on the model learning process as it outperforms both D0 and ATB segmentation schemes (See Tables: 4, 5 and 6). In other words, increasing the number of tokens in training dataset using D3 segmentation scheme, as shown in Table 1, leads to better word alignment and consequently improve the model performance.

The main conclusion is that, for Arabic-English

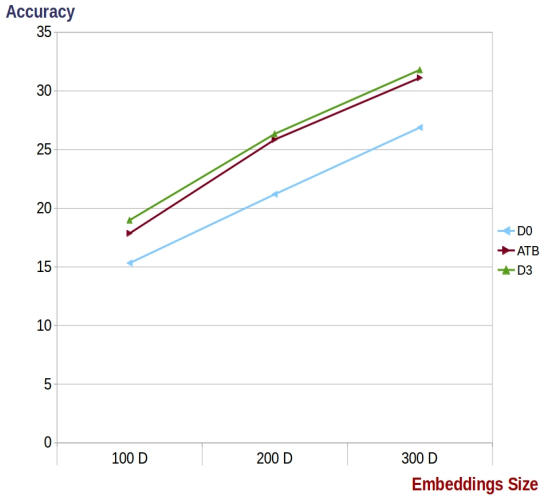

Figure 1: 5-10 sentence length training data results

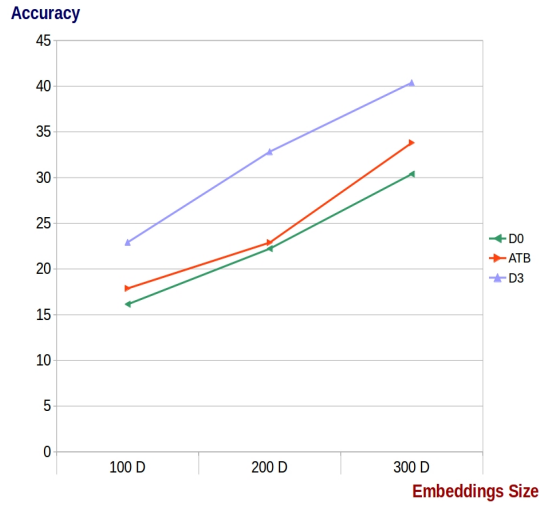

Figure 2: 17-80 sentence length training data results

in contrast to MT task, increasing embedding size, sentence length and more Arabic segmentation allow the model to capture more information and leads to learn better BWEs. See Figures 1 and 2. For Figure 1, short sentences training dataset shows that both segmented datasets: ATB and D3 give better results compared to D0 (No segmentation). D3 outperforms ATB slightly. In Figure 2, using the long sentence training dataset, D3 gives a much better performance compared to both other segmentation schemes, and increases the accuracy dramatically up to 10 %.

6 Conclusion

In this work, we have trained a BilBOWA model to investigate the effect of different morphological segmentations and different training settings (sentence-length and embeddings size) on learning BWE for Ar-En language pair. Our results show that increasing the word embedding size leads to improvement in the learning process of Arabic-English bilingual word embeddings.

For Arabic, as a morphological segmentation process is essential in many Arabic NLP tasks, segmentation also has a positive effect in this work as it leads to learning a better bilingual word embeddings. Going from D0 (full word form) to D3 (more segmentation, which increases the number of tokens in training dataset), decreases the distance between Ar-En pairs and increases the similarity more than 10 percentage points.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California. Association for Computational Linguistics.

Marwan Akeel and R. B. Mishra. 2014. A statistical method for english to arabic machine translation. *International Journal of Computer Applications*, 86(2):13–19.

Hassan Al-Haj and Alon Lavie. 2012. The impact of arabic morphological segmentation on broad-coverage english-to-arabic statistical machine translation. *Machine translation*, 26(1/2):3–24.

Ibrahim Badr, Rabih Zbib, and James Glass. 2008. Segmentation for english-to-arabic statistical machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, number 4 in HLT-Short '08, pages 153–156, Stroudsburg, PA, USA.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mohamed Boudchichea, Azzeddine Mazrouia, Mohamed Ould Abdallahi Ould Bebahb, Abdelhak Lakhouajaa, and Abderrahim Boudlalc. 2017. Alkhalil morpho sys 2: A robust arabic morpho-syntactic analyzer. *Journal of King Saud University*

Computer and Information Sciences, 29(2):141–146.

Samuel R. Bowman, Christopher Potts, and Christopher D. Manning. 2015. Recursive neural networks can learn logical semantics. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 12–21, Beijing, China. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. 2015. Trans-gram, fast cross-lingual word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1113, Lisbon, Portugal. Association for Computational Linguistics.

Ahmed El Kholy and Nizar Habash. 2012. Orthographic and morphological processing for english–arabic statistical machine translation. *Machine Translation*, 26(1-2):25–45.

Jakob Elming and Nizar Habash. 2009. Syntactic reordering for English-Arabic phrase-based machine translation. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 69–77, Athens, Greece. Association for Computational Linguistics.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 748–756. JMLR.org.

Nizar Habash. 2010. Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 573–580, Ann Arbor, Michigan. Association for Computational Linguistics.

Nizar Habash, Owen Rambow, and Ryan Roth. 2009. Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. In *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt*.

Nizar Habash and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 49–52, New York City, USA. Association for Computational Linguistics.

Arwa Hatem, Nazlia Omar, and Khalid Shaker. 2011. Morphological analysis for rule based machine translation. In *International Conference on Semantic Technology and Information Retrieval (STAIR)*, pages 260–263.

Karl Moritz Hermann and Phil Blunsom. 2013. The role of syntax in vector space models of compositional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 894–904, Sofia, Bulgaria. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340.

Ines Khemakhem, Salma Jamoussi, and Abdelmajid Ben Hamadou. 2010. The miracl arabic-english statistical machine translation system for iwslt 2010. In *Proceedings of IWSLT International Workshop on Spoken Language Translation*, pages 119–125.

Yoon. Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* ,, page 17461751, Doha, Qatar.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Lingpeng Kong, Chris Dyer, and Noah A Smith. 2015. Segmental recurrent neural networks. In *ICLR*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Stanislas Lauly, Alex Boulanger, and Hugo Larochelle. 2014. Learning multilingual word representations using a bag-of-words autoencoder. *CoRR*, abs/1401.1803.

Mike Lewis, Kenton Lee, and Luke Zettlemoyer. 2016. LSTM CCG parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 221–231, San Diego, California. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Tomas Mikolov, Martin Karafit, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2, pages 1045–1048.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2 of *NIPS'13*, pages 3111–3119, USA. Curran Associates Inc.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. *IEEE Workshop on Spoken Language Technology, SLT 2012 - Proceedings*.

Aditya Mogadala and Achim Rettinger. 2016. Bilingual word embeddings from parallel and non-parallel corpora for cross-language text classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 692–702, San Diego, California. Association for Computational Linguistics.

Azza Abdel Monem, Khaled Shaalan, Ahmed Rafea, and Hoda Baraka. 2008. Generating arabic text in multilingual speech-to-speech machine translation framework. *Machine translation*, 22(4):205–258.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland. European Language Resources Association (ELRA).

Sebastian Ruder. 2017. A survey of cross-lingual embedding models. *CoRR*, abs/1706.04902.

Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 959–962, New York, NY, USA. ACM.

Yongzhe Shi, Wei-Qiang Zhang, Jia Liu, and Michael T. Johnson. 2013. RNN language model with word clustering and class-based output layer. *EURASIP Journal on Audio, Speech, and Music Processing*, 2013(1):22.

Abdelhadi Soudi, Günter Neumann, and Antal Van den Bosch. 2007. *Arabic computational morphology: knowledge-based and empirical methods*. Springer.

Jinsong Su, Deyi Xiong, Biao Zhang, Yang Liu, Junfeng Yao, and Min Zhang. 2015. Bilingual correspondence recursive autoencoder for statistical machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1248–1258, Lisbon, Portugal. Association for Computational Linguistics.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-to-End memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Curran Associates, Inc.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence learning with neural networks. *CoRR*, abs/1409.3215.

Ivan Vulic and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, page 719725.

WIT3. 2012. Plain training and development sets for the mt track. https://wit3.fbk.eu/mt.php?release=2012-02-plain.

Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *CoNLL*, pages 119–129.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2016. BattRAE: Bidimensional attention-based recursive autoencoders for learning bilingual phrase embeddings. *CoRR*, abs/1605.07874.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 111–121, Baltimore, Maryland. Association for Computational Linguistics.