MappSent: a Textual Mapping Approach for Question-to-Question Similarity

Amir Hazem, Basma El Amal Boussaha, Nicolas Hernandez

MappSent: a Textual Mapping Approach for Question-to-Question Similarity

Amir Hazem[1] **Basma El Amal Boussaha**[1] **Nicolas Hernandez**[1]

[1] LS2N - UMR CNRS 6004, Université de Nantes, France

{Amir.Hazem,Basma.Boussaha,Nicolas.Hernandez}@univ-nantes.fr

Abstract

Since the advent of word embedding methods, the representation of longer pieces of texts such as sentences and paragraphs is gaining more and more interest, especially for textual similarity tasks. Mikolov et al. (2013a) have demonstrated that words and phrases exhibit linear structures that allow to meaningfully combine words by an element-wise addition of their vector representations. Recently, Arora et al. (2017) have shown that removing the projections of the weighted average sum of word embedding vectors on their first principal components, outperforms sophisticated supervised methods including RNN's and LSTM's. Inspired by Mikolov et al. (2013a); Arora et al. (2017) findings and by a bilingual word mapping technique presented in Artetxe et al. (2016), we introduce MappSent, a novel approach for textual similarity. Based on a linear sentence embedding representation, its principle is to build a matrix that maps sentences in a joint-subspace where similar sets of sentences are pushed closer. We evaluate our approach on the SemEval 2016/2017 question-to-question similarity task and show that overall MappSent achieves competitive results and outperforms in most cases state-of-art methods.

1 Introduction

Since the dawn of the mass access to the Internet fostered by the availability of data, more and more community question answering (CQA) forums such as StackExchange[1] and Qatar Living[2] have been established and are gaining more and more popularity. It is not unusual to rely on such source of information to find out a correct answer to a given question. However, feeding forums with perpetual questions and answers makes this resource massive and full of duplicate posts and similar question variants. Thus, and to some extent, the search for an answer has become hard to achieve and led to the emergence of an important area of research known as Community Question Answering (CQA).

In the CQA domain, the identification of similar questions is certainly an important preliminary step for providing a correct answer to a posted question. It is necessary to figure out if a question has not already been treated in other posts, essentially for a matter of response effectiveness and to reduce as much as possible duplicate posts. To that end, question-to-question similarity task offers a key challenge while it has to deal not only with similar questions in terms of lexical similarity but also in terms of reformulation, paraphrasing, semantics, etc. It has attracted a great interest as it can be seen in the SemEval shared task where a subtask is dedicated to it since 2015.

In this paper, we propose MappSent, a novel approach for textual similarity that we evaluate on the SemEval question-to-question similarity task. The main idea is to represent questions in a joint sub-space where similar pairs are moved closer thanks to a mapping matrix. Each question is represented by the element-wise addition of its words embedding vectors (Mikolov et al., 2013a; Arora et al., 2017). Then, based on a training set of question pairs equivalence, an optimal linear transformation matrix that minimizes the distance between similar questions is learned. The mapping matrix is built according to Artetxe et al. (2016) approach that was initially introduced for mapping word embeddings of different languages.

[1] http://stackoverflow.com/

[2] http://www.qatarliving.com/forum

We adapt this approach in a monolingual scenario at the sentence level. Questions are often pieces of texts that contain the context of the question and the question itself. We do not treat separately these two information, on the contrary, we consider both the context and the question as a whole segment that we call by misuse of language: *Sentence*. Our aim is to align two pieces of texts independently of their structures, as long as they exhibit similar characteristics that we try to capture over the proposed mapping matrix. The main contributions of this work are: (i) the introduction of MappSent as a new simple and sound way of representing sentences in an optimized joint sub-space, (ii) an extensive comparison with Arora et al. (2017) approach and, (iii) an empirical study of the impact of removing the first principal components as a preliminary step to questions similarity. We evaluate our approach on the SemEval 2016/2017 question-to-question similarity task (Task3, subtaskB) and show that overall, MappSent outperforms the state-of-art approaches.

2 Related Work

With the continuous evolution of neural embedding methods, several approaches ranging from a word level embedding representation (Bengio et al., 2003; Collobert and Weston, 2008; Mikolov et al., 2013a; Pennington et al., 2014) to a longer textual level embedding representation such as phrases, sentences, paragraphs or documents (Socher et al., 2011; Mikolov et al., 2013a; Le and Mikolov, 2014; Kalchbrenner et al., 2014; Kiros et al., 2015; Wieting et al., 2016; Arora et al., 2017) have been proposed. Word embedding methods try to capture lexical and semantic word's properties by representing words in a low continuous dimensional space (Bengio et al., 2003; Mikolov et al., 2013a,b). Previous longer textual embedding methods use operations on vectors and matrices like addition or multiplication to represent phrases, sentences or paragraphs (Mitchell and Lapata, 2008, 2010; Mikolov et al., 2013a; Wieting et al., 2016; Arora et al., 2017). Other more sophisticated approaches use recurrent neural networks (RNN) (Socher et al., 2011, 2014; Kiros et al., 2015), long short-term memory (LSTM) to capture long distance dependency (Tai et al., 2015) or convolutional neural networks (CNNs) (Kalchbrenner et al., 2014) to represent sentences. Even if RNNs, LSTMs and CNNs based approaches have shown remarkable improvements in a wide range of applications, their computational cost and the need of large amount of training data, makes these approaches inefficient on small and specific datasets.

While sentence embedding representation is our main focus, it is important to mention Mikolov et al. (2013a) approach where they have shown the possibility to efficiently represent phrases by the sum of their words embedding vectors. In their Skip-Gram model, word vectors are trained to predict surrounding words and thus, to represent the distribution of the context in which a word appears. As word vectors are in a linear relationship, the sum of two word vectors can be seen as the product of the two context distributions. On the phrase analogy task Mikolov et al. (2013a) demonstrated the effectiveness of their model with the hierarchical softmax and subsampling using large amount of data. Recently, using the paraphrase pairs dataset (PPDB), Wieting et al. (2016) have shown that a simple but supervised word averaging model of sentence embeddings leads to better performance on textual similarity tasks. However, the performance of their approach is closely related to the supervision from the paraphrase dateset, while without supervision, their approach did not perform well on textual similarity tasks. More recently, Arora et al. (2017) proposed a new sentence embedding method. Its principle is to first compute a weighted average sum of the word embedding vectors of sentences, and then, to remove the projections of the average vectors on their first principal components. Like Mikolov et al. (2013a) and Wieting et al. (2016), their approach is based on word embedding sum, but the difference is remarkable on the weighted schema and on the use of principal component analysis (PCA) method to remove the correlation of sentence vectors dimensions. They significantly achieved better performance than the unweighted average on a variety of textual similarity tasks. Also, their approach outperformed sophisticated supervised methods such as RNN's and LSTM's.

SemEval question-to-question similarity task offers an appropriate environment to evaluate our approach and validate our intuition. A wide range of approaches have been proposed since the beginning of SemEval. The winners of the 2016 edition (*UH-PRHLT*) for instance (Franco-Salvador et al., 2016), combine lexical and semantic fea-

tures and representations to measure similarity between pieces of texts. Their approach take advantage of distributed representations of words, graph knowledge constructed from BabelNet and frames extracted from FrameNet. The second best system (*ConvKN*) (Barrón-Cedeño et al., 2016) used convolutional neural networks to represent sentences. They used an SVM operating on three kernels and combined convolutional tree kernels with convolutional neural networks and additional manually extracted features including text similarity and thread specific features. The third best system (*KeLP*) (Filice et al., 2016), used SVM classifier based on a linear combination of kernel functions. Different features were used such as linguistic similarities, shallow syntactic trees encoding lexical and morpho-syntactic information, feature vectors capturing task specific information, etc. Several other systems have been proposed. Wu and Lan (2016) for instance used different ranking methods such as supervised models using traditional features as well as convolutional neural network and long-short term memory. They also proposed two novel methods to improve semantic similarity estimation by integrating ranking information of question-comment pairs. Wang and Poupart (2016) explored a two-layer feedforward neural network with the average of word embedding vectors to predict the semantic similarity score of two questions. While Wu and Zhang (2016) proposed a translation based method that combines a translation model with a cosine similarity based-method to deal with question similarity. Mihaylova et al. (2016) presented a feature rich system based on various types of features: semantic, lexical, metadata and user-related. Their best results were achieved thanks to metadata features. Even if user information conveyed by metadata can be very useful, we make the choice not to exploit it, while our main focus is on text analysis only.

With the success of the 2016 edition and the boom of neural networks, it has been noticed a jump in 2017 on the number of deep learning methods (Nakov et al., 2017). SimBow system, which did not participate in the previous year, is the winner of the 2017 edition on the question-to-question similarity task. The authors proposed a logistic regression on a combination of different unsupervised textual similarities. They introduced a variant of cosine similarity that uses semantic similarity between words to compute cosine between two bag-of-word vectors. The semantic relations were extracted using Word2Vec. LearningToQuestion system achieved the second best result using SVM and logistic regression as integrators of rich features representations (word embeddings, bidirectional LSTMs, gated recurrent unit (GRU), etc.). Kelp system which was ranked 3rd on last year edition, reached also the third place this year but with its contrastive[3] version could reach the first place. Talla system which was ranked at the fourth position, used a random forest classifier based on an ensemble of syntactic, semantic and IR-based features such as semantic word alignment, term frequency Kullback-Leibler divergence, and tree kernels (Nakov et al., 2017). A detailed description of SemEval 2016 and 2017 editions and their participants can be found in Nakov et al. (2016) and Nakov et al. (2017). Overall, the major part of SemEval state-of-art proposed approaches uses sophisticated and complex methods to deal with question-to-question similarity. One advantage of our approach is its simplicity while compared to SemEval systems.

3 MappSent Approach

In order to efficiently align similar sentences[4] and by analogy to word embedding representations, we build a sentence embedding space where sentences are represented by the sum of their word embedding vectors. Similar sentences are moved closer thanks to a mapping matrix (Artetxe et al., 2016) learned from a training dataset containing annotated similar sentences. Basically, a set of similar sentence pairs is used as seed information to build the mapping matrix. The optimal mapping is computed by minimizing the distance between the seed sentence pairs.

MappSent approach consists of the following steps:

1. We train a Skip-Gram [5] model using Gensim (Řehůřek and Sojka, 2010)[6] on a lemma-

[3] A contrastive approach refers to the non primary system. It is considered by the authors to be their second or third run.

[4] Or similar pieces of texts.

[5] CBOW model had also been experienced but it turned out to give lower results while compared to the SkipGram model.

[6] To ensure the comparability of our experiments, we fixed the python hash function that is used to generate random initialization. By doing so, we are sure to obtain the same embeddings for a given configuration.

tized training dataset. We use all the questions and answers provided by the *Qatar Living* forum (described in section 4) as training data. We consider all users interactions as a good source of information for context representation.

2. Each training and test sentence is pre-processed. We remove stopwords and only keep nouns, verbs and adjectives while computing sentence embedding vectors and the mapping matrix. This step is not applied when learning word embeddings (cf.Step 1).

3. For each given pre-processed sentence, we build its embedding vector which is the element-wise addition of its words embedding vectors (Mikolov et al., 2013a). Unlike Arora et al. (2017) we do not use any weighting procedure while computing vectors embedding sum[7].

4. We build a mapping matrix where test sentences can be projected. We adapted Artetxe et al. (2016) approach in a monolingual scenario as follows:

    - To build the mapping matrix we need a mapping dictionary which contains similar sentence pairs. To construct this dictionary, we consider pairs of sentences that are labeled as *PerfectMatch* and *Relevant* in the Qatar Living training dataset (cf section 4).
    - The mapping matrix is built by learning a linear transformation which minimizes the sum of squared Euclidean distances for the dictionary entries and using an orthogonality constraint to preserve the length normalization.
    - While in the bilingual scenario, source words are projected in the target space by using the bilingual mapping matrix, in our case, original and related questions are both projected in a similar subspace using the monolingual sentence mapping matrix. This consists of our adaptation of the bilingual mapping.

5. Test sentences are projected in the new subspace thanks to the mapping matrix.

6. The cosine similarity is then used to measure the similarity between the projected test sentences.

As it has been shown in Arora et al. (2017) that removing the projections of the average vectors on their first principal components improves the performance on textual similarity tasks, we apply this technique to our approach. We first compute PCA on the training dataset and then we remove the n first principal components before computing the cosine similarity between two test questions.

4 Data and Resources

In community question answering, the question-to-question similarity task (Task3, SubtaskB in SemEval) consists of reranking the related questions according to their similarity with respect to the original question. Each original question, has 10 candidates to rerank. These candidates are labeled as *PerfectMatch*, *Relevant* or *Irrelevant*. No distinction is made between *PerfectMatch* and *Relevant* labels, both are concidered as good candidates in SemEval task. The training and development datasets consist of 317 original questions and 3,169 related questions[8]. The test sets of 2016 and 2017 respectively consist of 70 original/700 related questions and 88 original/880 related questions. The official evaluation measure towards which all systems are evaluated is the mean average precision (MAP) using the 10 ranked related questions.

For building our Skip-Gram model, we used the training, development and test sets of 2015 (which is a dataset of question-comment pairs, it corresponds to the SubTask A of SemEval), in addition to the training and development sets of 2016 which contain for each original question, its related question and 10 related comments to each related question. It is to note that the training set of 2016 is the same as 2017. The size of the lemmatized training dataset is about 2 million words.

5 Experiments and Results

In this section we first present the results of $Arora$, $MappSent$ and the 3 best systems on the SemEval editions 2016 and 2017. Then, we compare MappSent and Arora approaches on the same datasets while varying different embedding

[7] We explored this direction without success.

[8] http://alt.qcri.org/semeval2016/task3/index.php?id=data-and-tools

parameters (window size, vectors dimension size, etc.) and the use or not of principal components analysis approach. Finally, we vary the number of principal components to find out the optimal configurations of PCA-based approaches. We note by $Arora$ and $Arora_{pca}$, the approaches presented in Arora et al. (2017). $Arora_{pca}$ is based on PCA removal while $Arora$ does not use PCA and is just a weighted sum of word embedding vectors of a sentence. We also propose four MappSent approaches. We note by $MappSent^{-}$ and $MappSent^{-}_{pca}$ our proposed approach that does not use the mapping matrix. It is merely the unweighted sum of word embeddings of a sentence ($MappSent^{-}$) and its PCA-based variant ($MappSent^{-}_{pca}$). We also note by $MappSent$ and $MappSent_{pca}$ our proposed approach that uses the mapping matrix ($MappSent$) and its PCA-based variant ($MappSent_{pca}$).

Tables 1 and 2 show the results of SemEval (2016/2017) of our proposed approaches (noted $MappSent^{-}$, $MappSent^{-}_{pca}$, $MappSent$ and $MappSent_{pca}$), Arora approaches (noted $Arora$ and $Arora_{pca}$) and the three best systems of the SemEval shared-task that are: $UH\text{-}PRHLT$, $ConvKN$ and $KeLP$ for the 2016 edition and *SimBow*, *LearningToQuestion* and *KeLP* for the 2017 edition. From the two Tables we see that $MappSent$ outperforms the three best systems as well as Arora approaches on both SemEval editions. The best MAP scores obtained by $MappSent$ are 79.18% (2016 edition) and 47.50% (2017 edition). We also notice that MappSent PCA-based approach ($MappSent_{pca}$) obtains the best results on 2017 with a MAP score of 49.29% while it is slightly under $MappSent$ with 79.09% of MAP score for 2016. Concerning $Arora$, $MappSent^{-}$ as well as their PCA-based variants ($Arora_{pca}$, $MappSent^{-}_{pca}$), we observe that all of them obtain competitive and sometimes better results while compared to the three best SemEval systems. This is the case for instance on 2016 where the four systems outperform the ranked first system $UH\text{-}PRHLT$. The results are more contrasted concerning the impact of PCA on the performance of $Arora$ and $MappSent$. While we observe a gain using PCA for $Arora_{pca}$ with a jump from 77.87% to 78.81% of MAP score, $MappSent^{-}_{pca}$ shows a non significant gain (a very little improvement from 78.56% to 78.66%). On the contrary, $MappSent_{pca}$ shows slightly lower results as it can be seen in Table 1. The results of Table 2 indicate opposite observations. This time $MappSent_{pca}$ shows significant improvements while $MappSent^{-}_{pca}$ and $Arora_{pca}$ don't. It is necessary to go deeper in parameters analysis to figure out their impact. This is the purpose of the next paragraphs.

Method	MAP(%)
$UH\text{-}PRHLT$	76.70
$ConvKN$	76.02
$KeLP$	75.83
$Arora$	77.87
$Arora_{pca}$	78.81
$MappSent^{-}$	78.56
$MappSent^{-}_{pca}$	78.66
$MappSent$	**79.18**
$MappSent_{pca}$	79.09

Table 1: Results on SemEval-2016 Task3 Subtask B

Method	MAP(%)
$Simbow$	47.22
$LearningToQuestion$	46.93
$KeLP$	46.66
$Arora$	46.93
$Arora_{pca}$	46.66
$MappSent^{-}$	46.90
$MappSent^{-}_{pca}$	46.53
$MappSent$	47.50
$MappSent_{pca}$	**49.29**

Table 2: Results on SemEval-2017 Task3 Subtask B

5.1 Window and Dimension Size Comparison

Table 3 presents a comparison of MappSent and Arora approaches using different parameters. For embeddings training, we used as settings a window size of 5,10 and 20, negative sampling of 5, sampling of 1e-3 and training over 15 iterations. We applied the Skip-gram model to create vectors of 100, 300, 500 and 800 dimensions. We used hierarchical SoftMax for training the Skip-gram model. Other settings were assessed but on average the chosen ones tend to give the best results on the development data. Concerning the number of principal components, on average the best results were obtained by removing 1 or 2 principal components.

	SemEval 2016				SemEval 2017				
	Dimension size								Window size
Approach	100	300	500	800	100	300	500	800	
$Arora$	75.86	75.48	75.52	76.41	44.67	44.44	44.36	44.22	5
$Arora_{pca}$	77.47	76.98	75.45	77.07	45.07	44.85	45.55	45.26	
$MappSent^{-}$	76.16	77.01	76.44	76.62	45.39	45.43	45.41	45.15	
$MappSent^{-}_{pca}$	77.43	76.73	76.47	77.01	45.64	46.01	45.76	45.78	
$MappSent$	78.47	78.14	77.39	78.03	46.62	46.44	47.38	47.30	
$MappSent_{pca}$	77.13	77.91	76.99	77.91	47.56	48.24	48.66	48.15	
$Arora$	75.71	76.49	76.28	77.16	45.08	45.81	44.90	43.82	10
$Arora_{pca}$	76.21	77.02	77.02	77.03	44.81	46.39	**46.66**	45.33	
$MappSent^{-}$	75.94	76.47	77.37	77.86	45.54	45.60	45.83	45.48	
$MappSent^{-}_{pca}$	76.26	78.07	78.41	77.36	46.39	45.22	**46.53**	45.33	
$MappSent$	76.95	78.09	78.74	78.70	47.36	45.92	46.99	46.83	
$MappSent_{pca}$	77.12	77.19	76.55	76.33	48.57	47.22	47.89	48.15	
$Arora$	76.18	76.47	77.45	**77.87**	**46.93**	44.24	44.41	44.36	20
$Arora_{pca}$	78.03	**78.81**	78.05	78.11	45.40	44.66	44.50	44.86	
$MappSent^{-}$	76.39	77.45	77.51	**78.56**	**46.90**	44.66	45.36	45.72	
$MappSent^{-}_{pca}$	77.72	**78.66**	78.24	78.32	45.74	44.27	46.03	46.28	
$MappSent$	78.52	**79.18**	79.00	78.83	**47.50**	46.88	46.88	47.44	
$MappSent_{pca}$	78.43	78.39	**79.09**	79.02	47.80	48.03	48.00	**48.72**	

Table 3: Comparison of $Arora$ and $MappSent$ using different window and dimension size (results in bold represent the best score of each approach), the number of PCA components was fixed to 1 or 2 (MAP %)

Our first comparison concerns $Arora$ and $MappSent^{-}$ which are similar approaches in the idea of computing the sum of word embedding vectors of sentences. The difference mainly resides in the fact that $Arora$ uses a smoothed inverse frequency to weight word vectors while $MappSent^{-}$ is an unweighted approach[9]. We see that for both editions and in the majority of cases, $MappSent^{-}$ outperforms $Arora$. The best $Arora$ MAP scores are: 77.87% for 2016 (w=20 and dim=800) and 46.93% for 2017 with the same window size and 100 dimensions. $MappSent^{-}$ obtained better results on 2016 with a MAP score of 78.56% (w=20 and dim=800) and a slightly lower result on 2017 with a MAP score of 46.90% (w=20 and dim=100). It is to note that both approaches were evaluated under the same conditions that are: lemmatization, stopwords and POS-TAG filtering as well as word embeddings trained on the same corpus. Arora approach under the original conditions presented in Arora et al. (2017) was tested but the results were much lower using Wikipedia embeddings and no POS-TAG filtering.

The second comparison concerns the use of PCA in $Arora$ and $MappSent^{-}$. We measure the contribution of removing the first principal components (1 or 2) while varying window and dimension size of word embeddings. The results are obtained using $MappSent^{-}_{pca}$ and $Arora_{pca}$. We see that the use of PCA almost always improve the performance of $Arora$ approach and except few cases, it also always improve the results of $MappSent^{-}$. The best $Arora_{pca}$ MAP scores are: 78.81% for 2016 (w=20 and dim=300) and 46.66% for 2017 (w=10 and dim=500). $MappSent^{-}_{pca}$ obtained higher results on 2016 with a MAP score of 78.66% (w=20 and dim=300) and a slightly lower result on 2017 with a MAP score of 46.53% (w=10 and dim=500).

For the third comparison, we are interested in the performance of the main proposed approach which is $MappSent$ regarding $Arora$

[9]It is to note that different weighting schema have been tried, surprisingly they all degraded the results of $MappSent$.

and $Arora_{PCA}$. We notice that $MappSent$ always outperform the latter approaches (except very few cases). The best $MappSent$ MAP scores are: 79.18% for 2016 (w=20 and dim=300) and 47.50% for 2017 (w=20 and dim=100).

Interestingly, the use of PCA improves $MappSent$ performance in most cases on 2017 test set while it degrades its performance in most cases on 2016 test set. The best $MappSent_{PCA}$ MAP scores are: 79.09% for 2016 (w=20 and dim=500) and 48.78% for 2017 (w=20 and dim=800). The number of principal components was fixed to one or two depending on the approach. However it is necessary to conduct an empirical study on the impact of the number of PCA components on PCA-based approaches. This is the purpose of the next Section.

5.2 Principal Components Impact

In this section we compare $Arora$ and $MappSent$ PCA-based approaches regarding the number of principal components that were removed before the computation of sentence similarity. We vary the number of components from 0 to 10 and give an arbitrary upper bound of 20 components.

# PCA	$Arora$	$MappSent^{-}$	$MappSent$
0	76.47	77.45	**79.18**
1	**78.81**	78.66	78.39
2	77.46	77.80	77.66
3	77.20	78.35	77.63
4	77.91	**78.82**	78.02
5	78.20	78.01	77.13
6	78.59	78.14	77.34
7	78.33	78.09	77.60
8	77.64	77.69	77.51
9	77.64	77.72	78.13
10	77.16	77.14	78.19
20	76.51	75.86	77.08

Table 4: Comparison of $Arora$ and $MappSent$ on SemEval 2016 while removing different numbers of principal components (w=20 and dim= 300)

According to Tables 4 we clearly notice the positive impact of using PCA in $Arora$ and $MappSent^{-}$. The best results are obtained with one component for $Arora$ and four components for $MappSent^{-}$. Concerning $MappSent$, the use of PCA degrades its performance which is somehow surprising regarding $MappSent^{-}$. From Table 5, all the approaches benefit from PCA components removal. The best results are obtained with two components for $Arora$, seven for $MappSent^{-}$ and four components for $MappSent$. If we can observe the influence of PCA on the experiments, it is however difficult to efficiently fix the the most appropriate number of principle components to use. In addition, it is clear that a high number of principal components is in most cases inefficient.

# PCA	$Arora$	$MappSent^{-}$	$MappSent$
0	44.90	45.83	47.36
1	46.66	46.53	46.77
2	**47.40**	46.81	48.57
3	46.86	46.52	49.07
4	46.50	46.70	**49.29**
5	45.60	46.79	48.69
6	45.72	46.52	47.55
7	47.19	**47.21**	47.77
8	46.97	46.53	47.24
9	45.51	46.48	47.41
10	45.35	46.15	46.84
20	46.53	47.07	46.70

Table 5: Comparison of $Arora$ and $MappSent$ on SemEval 2017 while removing different numbers of principal components (w=10 and dim=500)

6 Discussion

The multiple experiments and results have clearly demonstrated the effectiveness of our approach since MappSent and its PCA variant outperformed the best SemEval systems of 2016 and 2017 editions. Hence, the idea of mapping sentences in the same sub-space suggests a better sentence representation. Two key points must however be discussed. First, sentence representation by its words embeddings sum and second, the way of building the mapping matrix and the sentence projection procedure. For sentence representation, it is unclear why a simple words embedding vectors sum performs in most cases better than a weighted sum (as in $Arora$ for instance). That said, this can be partially explained by the fact that we remove

stopwords and some POS-TAGs from each sentence. Keeping nouns, verbs and adjectives only, makes sentences smaller and this probably reduce the impact of a weighting schema. The mapping matrix has been built and optimized on a small training dataset using orthogonal constraint, unit normalization and mean centering reduction. The set of training similar sentence pairs was small (about 2000 question pairs). A question remains on how our approach could perform if the mapping matrix was trained on a large sentence database such as the paraphrasing database (PPTB) for instance? We let this question for future work. In addition, one important adaptation of Artetxe et al. (2016) approach is the projection phase. While in a bilingual scenario source words are mapped into the target language, in our monolingual case, we map both source questions (the original questions) and target questions (the related questions). It wouldn't make sense to only map the source questions as we need to represent both pairs in the same sub-space.

In most cases, $MappSent$ and $Arora$ perform better on higher window size (10 or 20). For vector dimensions the results are more contrasted (300, 500 or 800). While it is difficult to clearly pinpoint the reasons of such observation, it is well established that smaller windows capture syntactic/semantic dependencies, while larger windows capture topical structures (Mikolov et al., 2013b). As our datasets treat different topics of the Qatar daily life, one can suppose that topical information maybe more discriminant than the one provided by syntactic information, at least in these experiments.

An important phase is certainly embedding models. Word embedding vectors have been trained using the Skip-Gram model[10]. Here also, and as it has been already shown (Mikolov et al., 2013a,b), the Skip-Gram model performs better than CBOW model on small datasets. Another interesting information is the fact that training word embeddings on a specific dataset (here Qatar living) performs better than using pretrained embeddings such as wikipedia or other bigger size corpora. This can also be explained by the general representation of such embeddings which maybe inappropriate when dealing with specific domains. One interesting direction which we also let for future work is to contrast different domains corpora and also use data selection before training our embedding models.

Finally, we could notice the positive impact of PCA in most cases except for $MappSent$ [11] on the 2016 test set. Removing principal components from a sum of word embeddings is useful while the resulting sentence embedding vectors are uncorrelated. Hence, similar information is removed which makes sentence comparison more efficient. However, one drawback of PCA among other mathematical transforms is its sensitivity to the original data. One possible reason that can explain PCA performance is probably the correlation between the training and the test datasets. Another PCA drawback is the empirical way to fix the number of principal components. It would be interesting to explore other discriminant mathematical transformations such as canonical correlation analysis (CCA) or independent component analysis (ICA).

7 Conclusion

In this paper we have proposed MappSent, a novel approach for textual similarity. Our approach allows to map sentences in a joint more representative sub-space. Thanks to questions mapping matrix, similar questions are pushed closer suggesting that the new sub-space is more discriminant. The experimental results confirm our intuition while MappSent and its PCA-based variant obtain the best results on SemEval (2016/2017) question-to-question similarity task over state-of-art approaches. One remarkable advantage of MappSent is its simplicity while neither intensive computation nor external resources or metadata are needed. In addition, MappSent can be applied to pieces of text of any length as long as a training set of similar texts is available. That said, no attention has been given to linguistic information and questions were treated as bags-of-words. For future work, we intend to explore linguistic features as well as exploiting the context of a question and the question itself differently. Another exciting challenge is to apply our approach to questions and answers. The use of metadata is also another interesting direction that we leave for the future.

[10] It is to note that some experiments using the CBOW model have been conducted but the performance were much lower than using the Skip-Gram model.

[11] Normalization and mean centering embedding vectors as well as a weak correlation between training and test data may explain this behaviour.

Acknowledgments

The current work was both supported by the Unique Interministerial Fund (FUI) No. 17 as part of the ODISAE[12] project and the ANR 2016 PASTEL[13].

References

Sanjeev Arora, Liang Yingyu, and Ma Tengyu. 2017. A simple but tough to beat baseline for sentence embeddings. In *Proceedings of the 17th International Conference on Learning Representations (ICLR'17)*. pages 1–11.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*. Austin, TX, USA, pages 2289–2294. https://aclweb.org/anthology/D16-1250.

Alberto Barrón-Cedeño, Giovanni Da San Martino, Shafiq Joty, Alessandro Moschitti, Fahad Al-Obaidli, Salvatore Romeo, Kateryna Tymoshenko, and Antonio Uva. 2016. Convkn at semeval-2016 task 3: Answer and question selection for question answering on arabic and english fora. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 896–903. http://www.aclweb.org/anthology/S16-1138.

Yoshua Bengio, Rjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *JOURNAL OF MACHINE LEARNING RESEARCH* 3:1137–1155.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning, ICML*.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 1116–1123. http://www.aclweb.org/anthology/S16-1172.

Marc Franco-Salvador, Sudipta Kar, Thamar Solorio, and Paolo Rosso. 2016. UH-PRHLT at semeval-2016 task 3: Combining lexical and semantic-based features for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. pages 814–821.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *CoRR* abs/1404.2188.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., pages 3294–3302.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *CoRR* abs/1405.4053.

Tsvetomila Mihaylova, Pepa Gencheva, Martin Boyanov, Ivana Yovcheva, Todor Mihaylov, Momchil Hardalov, Yasen Kiprov, Daniel Balchev, Ivan Koychev, Preslav Nakov, Ivelina Nikolova, and Galia Angelova. 2016. Super team at semeval-2016 task 3: Building a feature-rich system for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. pages 836–843.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pages 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Tomas Mikolov, Scott Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*. Columbus, Ohio, pages 236–244.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science* 34(8):1388–1439.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic*

[12] http://www.odisae.com

[13] http://www.agence-nationale-recherche.fr/?Projet=ANR-16-CE33-0007

Evaluation. Association for Computational Linguistics, Vancouver, Canada, SemEval '17.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, San Diego, California, SemEval '16.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, pages 45–50. `http://is.muni.cz/publication/884893/en`.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In *Advances in Neural Information Processing Systems 24*.

Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *TACL* 2:207–218.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *CoRR* abs/1503.00075.

Hujie Wang and Pascal Poupart. 2016. Overfitting at semeval-2016 task 3: Detecting semantically similar questions in community question answering forums with word embeddings. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. pages 861–865.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. *Internationa Conference on Learning Representations, CoRR* abs/1511.08198.

GuoShun Wu and Man Lan. 2016. ECNU at semeval-2016 task 3: Exploring traditional method and deep learning method for question retrieval and answer ranking in community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. pages 872–878.

Yunfang Wu and Minghua Zhang. 2016. ICL00 at semeval-2016 task 3: Translation-based method for CQA system. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. pages 857–860.