Meta-Embedding Sentence Representation for Textual Similarity

Amir Hazem[1] **Nicolas Hernandez**[1]
[1] LS2N - UMR CNRS 6004, Université de Nantes, France
{Amir.Hazem,Nicolas.Hernandez}@univ-nantes.fr

Abstract

Word embedding models are now widely used in most NLP applications. Despite their effectiveness, there is no clear evidence about the choice of the most appropriate model. It often depends on the nature of the task and on the quality and size of the used data sets. This remains true for bottom-up sentence embedding models. However, no straightforward investigation has been conducted so far. In this paper, we propose a systematic study of the impact of the main word embedding models on sentence representation. By contrasting in-domain and pre-trained embedding models, we show under which conditions they can be jointly used for bottom-up sentence embeddings. Finally, we propose the first bottom-up meta-embedding representation at the sentence level for textual similarity. Significant improvements are observed in several tasks including question-to-question similarity, paraphrasing and next utterance ranking.

1 Introduction

According to Enkvist (1987): *"a model is a simplified representation of reality. It is simplified because it aims at reproducing a selection of relevant elements of reality rather than all of reality at once."*. If several word embedding models (Mikolov et al., 2013a; Pennington et al., 2014; Yin and Schütze, 2016; Arora et al., 2017) capture a selection of relevant features, different embedding sets can cover different characteristics which can also be complementary (Yin and Schütze, 2016). In order to capture a wide range of features, it is useful to perform models combination (ensemble models). The representation of longer pieces of texts such as sentences, by an element-wise sum of their word embeddings has recently shown promising results and outperformed sophisticated models in several textual similarity tasks (Mikolov et al., 2013a; Arora et al., 2017). This representation, also known as bottom-up sentence embeddings, is greatly affected by the choice of word embedding models. In this paper, we propose a systematic study of the impact of word embedding models on bottom-up sentence representation for textual similarity. We report the results of the main individual pre-trained embedding models that are publicly available as well as embedding models trained on in-domain data sets. Finally, we contrast multiple ensemble models and propose the first bottom-up meta-embedding sentence representation for textual similarity. We evaluate the different approaches on four tasks that is: question-to-question similarity (SemEval 2016/2017), textual entailment (SemEval 2014), paraphrasing (Sick) and next utterance ranking (NUR) and show under which conditions meta-embeddings can be beneficial to bottom-up sentence-based approaches.

2 Related Work

Embedding models at the word level representations have been widely explored in many applications (Bengio et al., 2003; Collobert and Weston, 2008; Mikolov et al., 2013a; Pennington et al., 2014; Bojanowski et al., 2016). Naturally, they have been extended to sentence, paragraph and document level representations (Socher et al., 2011; Mikolov et al., 2013a; Le and Mikolov, 2014; Kalchbrenner et al., 2014; Kiros et al., 2015b; Wieting et al., 2016; Arora et al., 2017), thanks to the continuous advances of deep neural embedding methods such as Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM) and Convolutional Neural Networks (CNN). Some sentence embedding representations can be seen as a direct inspiration from word embedding models. For instance, while the skip-gram model (Mikolov et al., 2013a) predicts the surrounding words given a source word, in the same way, *SkipThought* (Kiros et al., 2015a) and *FastSent* (Hill et al., 2016) models predict surrounding sentences given a source sentence. Also, the paragraph *DBOW* model (Le and Mikolov, 2014) learns representations for variable length pieces of texts and learns to predict the surrounding words based on contexts sampled from paragraphs. Recently, Pagliardini et al. (2018) introduced *Sent2Vec*, an approach based on word vectors along with n-gram embeddings simultaneously to represent sentences.

Another type of sentence embedding representation, also called bottom-up approach, represents sentences by a weighted sum of the embedding vectors of their individual words. This naive approach turned out to be competitive and outperformed sophisticated approaches based on RNNs and LSTMs in many natural language processing applications (Mikolov et al., 2013a; Wieting et al., 2016; Arora et al., 2017; Hazem and Morin, 2017). Mikolov et al. (2013a) for instance demonstrated the effectiveness of their model on the phrase analogy task. They used the hierarchical softmax and subsampling using large amount of data. Wieting et al. (2016) have shown that a simple but supervised word averaging model of sentence embeddings leads to better performance on the paraphrase pairs data set (PPDB). However, the performance of their approach is closely related to the supervision from the date set, 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 where they first compute a weighted average sum of the word embedding vectors of sentences, and then, 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. A noticeable remark is that their approach outperformed sophisticated supervised methods such as RNN's and LSTM's. Finally, some approaches are supervised and need labelled data, such as *DictRep* (Hill et al., 2015) which uses structured data to map dictionary definitions of words with their pre-trained embeddings. With the encouraging results and simplicity of bottom-up approaches, we focus in this paper on this type of approaches and show their potential while used jointly with meta-embeddings.

3 Sentence Meta-Embedding Representation

To deal with textual similarity, we propose a new approach that we refer to as meta-embedding sentence representation (MetaSentEmb). In the next sections we first recall the principle of ensemble approach from which we drawn our inspiration, then we give the details of our approach.

3.1 Ensemble Approach

The principle of the ensemble approach is to combine different models in order to catch the strength of each individual model. The main combination techniques that have shown their effectiveness are: vector addition (Garten et al., 2015) and vector concatenation (Garten et al., 2015; Yin and Schütze, 2016). For vector addition, given two embedding models, the procedure consists of applying a simple dimension-wise vector addition[1]. For vector concatenation, given two embedding models of dimensions $dim1$ and $dim2$, the resulting concatenated embedding vector will be of size $dim1 + dim2$. The vectors have to be normalized before concatenation. Usually L2 norm is performed[2]. Yin and Schütze (2016) performed a weighted concatenation of 5 embedding models. They also experienced the SVD on top of weighted concatenation vectors of dimension 950. This resulted in a reduced model of 200 dimensions.

3.2 Proposed Approach

The bottom-up sentence embedding representation consists of representing each given sentence (or piece of text of any length) by an embedding vector which is the sum of the vector embedding of each word of the sentence (Mikolov et al.,

[1] This technique can not be applied when embeddings are not of the same dimension size.

[2] L2 norm can be performed either at dimension level (as suggested by Glove authors) or at vector length level.

2013b; Wieting et al., 2016; Arora et al., 2017). This representation is illustrated in the following equation:

$$Sent_i = \sum_{j=1}^{n} (Embedding(w_j)) \quad (1)$$

with $Sent_i$ a given sentence i and n the number of words in $Sent_i$. $Embedding(w_j)$ corresponds to the embedding model used to represent each word of the sentence $Sent_i$. We refer to this baseline approach as $SentEmb$ for sentence embedding representation. A variant of this representation is the use of a weighted sum as presented in (Wieting et al., 2016) for instance.

In this work, we extend the baseline representation ($SentEmb$) and propose $MetaSentEmb$, a meta-embedding sentence representation. We aim at improving sentence representation based on the sum of its word embeddings. As we mainly operate at the word level representation, we study different word meta-embedding techniques for sentence representation. We basically use an ensemble approach to represent each word, which means that each word has its own meta-embedding. Then, we sum each meta-embedding word of a given sentence to obtain a meta-embedding sentence representation (equation 2).

$$Sent_i = \sum_{j=1}^{n} (Ensemble(w_j)) \quad (2)$$

with $Sent_i$ a given sentence i and n the number of words in $Sent_i$. $Ensemble(w_j)$ corresponds to the ensemble technique used to represent word meta-embeddings. $Ensemble(w_j)$ can be the additive or the concatenation technique. We refer to our proposed approach as $MetaSentEmb$ for sentence meta-embedding representation. Each sentence is pre-processed (Tokenization, part-of-speech tagging and lemmatization). Depending on the targeted task, stop-words can be removed and part of speech filtering can be applied (keeping only nouns, verbs and adjectives for instance).

4 Data and Tasks Description

In this section, we briefly outline the different textual resources used for our experiments, namely: (i) the Qatar Living corpus used in SemEval 2016/2017 for question similarity task, (ii) the Sick corpus used in SemEval 2014 for textual entailment and relatedness, (iii) the Microsoft Research Paraphrase Corpus (MSRPC) used for paraphrase detection and (iv) the Ubuntu Dialogue Corpus used for Next Utterance Ranking (NUR).

4.1 Embedding Models

To study the impact of external data and context representation, we chose different embedding models. In addition to the word2vec model trained on Google News (Mikolov et al., 2013a), we used the two Glove models respectively trained on Wikipedia+GigaWord ($Glove6B$) and on Common Crawl ($Glove42B$) (Pennington et al., 2014). We also used three Wikipedia pre-trained models (Levy and Goldberg, 2014), that is, two linear bag of word contexts and one dependency-based context. The bag of word models use a context size of 5 ($Bow5_C$ corresponds to CBow and $Bow5_W$ to skipgram). The dependency-based model used syntactic relations ($Deps$). Finally, we experienced the recent proposed character n-gram model (Bojanowski et al., 2016) by using the character Skip-gram model trained on Wikipedia ($ChSG$). A summary of the pre-trained out-of-domain embedding sets is presented in Table 1. We also trained embedding models (CBOW, Skipgram, Glove and character n-gram models) on in-domain data sets (Qatar Living and Sick corpus of SemEval, MSPR for paraphrasing and Ubuntu for NUR). We respectively noted in domain trained embeddings as $CBow$, $SkipGram$, $Glove$, $CharSG$ and $CharCBOW$.

4.2 Data Sets

4.2.1 Qatar Living Corpus

The Qatar Living corpus is a community question answering data set made of original and related questions and their n corresponding answers. The training and development data sets consist of 317 original questions and 3,169 related questions[3]. The test sets of 2016 and 2017 respectively consist of 70 original/700 related questions and 88 original/880 related questions. The SemEval (2016/2017) question-to-question similarity shared task (Task3, SubtaskB) consists of identifying for each original question, its corresponding related questions over 10 candidates (Nakov et al., 2016, 2017). The question-to-question similarity task of SemEval offers an appropriate and interesting framework for evaluating our meta-

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

Model	Vocab	Dim	Training Data
word2vec	93k	300	Google News (Mikolov et al., 2013a)
Glove6B	400k	50-300	Wikipedia-Gigaword (Pennington et al., 2014)
Glove42B	1.9M	300	Common Crawl (Pennington et al., 2014)
Bow, Deps	175K	300	Wikipedia (Levy and Goldberg, 2014)
CharSG	175K	300	Wikipedia (Bojanowski et al., 2016)

Table 1: Pre-trained embedding sets (Dim: dimension size).

embedding approach since an evaluation of multiple approaches including sentence embeddings have been already performed.

4.2.2 Sick Corpus

The sick data set consists of 10,000 English sentence pairs annotated for relatedness in meaning (a score form 1 to 5) and for entailment (Neutral, Entailment or Contradiction). The SemEval 2014 shared task (Task1) consists of predicting whether two given sentences are entailed, contradictions or neutral. Using sentence embeddings as well as meta-embeddings for entailment prediction is an appropriate textual similarity task for evaluation, however, dealing with contradictions and neutral sentences is more difficult than a binary classification which consists of predicting whether sentences are entailed or not. In any case and for the sake of comparison, we perform the same evaluation as the state of the art approaches by keeping the three classes (Neutral, Entailment or Contradiction) instead of two classes (Entailment or Not Entailment). As this work is mainly dedicated to the evaluation of sentence representations in sentence similarity, we only focus on the entailment part and don't consider the relatedness (we only report the results of the accuracy).

4.2.3 Microsoft Research Paraphrase Corpus

The Microsoft Research Paraphrase (MSRP) Corpus (Dolan et al., 2004) is composed of 5,801 news paraphrase sentence pairs extracted from the web. Each sentence pair has been annotated by humans as being in paraphrase relationship (label=1) or not (label=0). 67% of sentence pairs are positive examples (in paraphrase relationship) and 33% are negative examples which make the corpus unbalanced. The corpus has been divided into 4,076 training pairs and 1,725 test pairs. The paraphrasing task consists of identifying if a paraphrase relation exists between two given sentences. By contrast to the question similarity and entailment prediction tasks, sentence embedding similarity for paraphrasing might not be appropriate for evaluation since the MSRP corpus includes many sentence pairs which are not paraphrases but contain many similar words. Sentence similarity based approaches should fail in this case to detect paraphrases. However, showing the behaviour and the performance of sentence similarity approaches including meta-embeddings on such a task may offer some clues and may constitute a baseline for more sophisticated approaches.

4.2.4 Ubuntu Dialogue Corpus

The Ubuntu Dialogue Corpus is a large freely available multi-turn dialogue data set (Lowe et al., 2015) constructed from the Ubuntu chat logs[4]. The corpus (Human-Human chat) consists of approximately 930,000 two person dialogues, 7,100,000 utterances[5] and 100,000,000 words. The task of NUR consists of retrieving the most probable utterance among a database of existing human productions given a similar context. This task offers a key challenge for sentence similarity approaches since the relations between dialogue utterances are more generic. Here also, evaluating sentence similarity based approaches on a different task, should give some insights about their behaviour and to what extent it might help utterance prediction.

[4] The first version can be found in http://irclogs.ubuntu.com/. A newer version has been recently released in https://github.com/rkadlec/ubuntu-ranking-dataset-creator

[5] All the replies and initial questions are referred to as utterances

Input		Tasks					
Training data	Models	SemEval16	SemEval17	MSRP	SICK	NUR	
		Map (%)	Map (%)	Accuracy	Accuracy	1 in 10 R@1	
1*Wiki/GigaWord	Glove6B	74.63 (8)	43.12 (5)	69.7 (5)	**64.3** (2)	63.6 (4)	(4)
1*Common Crawl	Glove42B	74.93 (7)	41.68 (9)	66.9 (8)	61.5 (8)	59.4 (11)	(10)
1*Google News	word2vec	74.42 (9)	42.38 (8)	67.5 (7)	63.5 (5)	62.0 (7)	(8)
4*wikipidia	Bow5C	74.08 (10)	42.93 (6)	66.4 (9)	47.1 (12)	58.8 (12)	(11)
	Bow5W	**75.64** (2)	**45.69** (2)	70.1 (4)	**63.9** (3)	62.6 (5)	(3)
	Deps	75.02 (6)	44.33 (4)	67.9 (6)	63.1 (7)	60.2 (8)	(6)
	CharSG	75.08 (5)	42.91 (7)	**71.8** (2)	**64.4** (1)	60.1 (9)	(7)
5* In-domain	Glove	73.04 (11)	41.57 (10)	64.9 (11)	54.4 (11)	62.3 (6)	(12)
	Cbow	72.78 (12)	40.12 (11)	65.1 (10)	60.1 (10)	**66.1** (2)	(12)
	SkipGram	**76.16** (1)	**45.58** (3)	**70.3** (3)	**63.9** (3)	**68.5** (1)	(1)
	CharCBOW	75.13 (4)	45.23 (4)	63.4 (12)	61.1 (9)	59.8 (10)	(9)
	CharSG	**75.21** (3)	**46.75** (1)	**72.1** (1)	63.3 (6)	**64.2** (3)	(2)

Table 2: Results of SentEmb for five distinct textual relation detection tasks (question-to-question with SemEval16 and SemEval17, paraphrase with MSRP, entailment with SICK and Next Utterance Ranking with UDC) using different pre-trained out-of-domain and in-domain embedding models. The numbers in brackets refer to the model rank in the given task, except for the last column which ranks the models regardless of the task. The score of the three best models for each task are in bold.

5 Results and Discussion

We conducted two sets of experiments. The first one aims at providing insights about the behaviour of pre-trained and in-domain embeddings used individually. The second one aims at studying the contribution of ensemble models.

Table 2 shows that, regardless of the task, the skipgram models (in-domain SkipGram, in-domain CharSG, out-of-domain Wikipedia Bow5W) outperform the other models. In addition, the two best models are in-domain and the two following are out-of-domain. The fourth position is hold by the Wikipedia/GigaWord Glove6B model. Among the worst models, we observe two CBOW models (in-domain Cbow out-of-domain Bow5C) as well as the in-domain Glove and the out-of-domain Glove42B models. Having said that, even if the differences between the extremities are notable, the coefficients of variability between two successive ranked scores are often very low. A closer look at the results shows that the out-of-domain and in-domain character skipgram models (CharSG) performed best for the paraphrase prediction task (MSRP). The entailment detection task (SICK) is the only one for which best models are largely out-of-domain. This can be explained by the very small size of the in-domain training data set. Surprisingly, the in-domain CBow which is globally one of the two worst models achieves the second best position for the next utterance ranking task (NUR). This can be explained by the large size of the in-domain Ubuntu data set while compared to other in-domain data sets.

Table 3 reports the results of MetaSentEmb approach for the four tasks using several pre-trained embedding combinations. Overall, we observe that pre-trained embedding combination is useful in the majority of tasks (except the SICK task where no significant improvements were observed). That said, not all the combinations are efficient. It depends on the tasks and on the nature of the training data sets. For instance, in the question-to-question similarity task (Semeval) the best meta-embedding models combination were Glove6B with Glove42B (76.2% using addition and 76.4% using concatenation on 2016 edition) and CharSG with Glove42B (76.5% using addition and 76.2% using concatenation on 2016 edition), while for 2017 edition the best models were Glove6B with word2vec (47.3% using concatenation) and Deps combined with Glove42B (47.4% using addition). It is to note that Deps concatenated to Bow5W obtained similar results with

Tasks	Pre-trained embedding models																	
	Glove6B						ChSG					w2v				Deps		
	Gl42	w2c	B5C	B5W	Deps	ChSG	Gl42	w2v	B5C	B5W	Deps	Gl42	B5C	B5W	Deps	Gl42	B5C	B5W
$SE16_A$	**76.2**	75.2	75.5	75.6	74.8	74.6	**76.5**	73.9	74.8	75.8	73.57	75.0	74.1	75.4	74.8	75.4	74.2	75.1
$SE16_C$	**76.4**	76.2	75.6	76.1	74.4	75.0	76.2	74.8	74.7	75.8	74.7	75.6	74.6	76.1	74.2	75.3	74.8	75.3
$SE17_A$	44.1	44.5	43.4	46.4	45.3	45.0	43.8	42.4	44.1	45.8	45.4	44.6	43.8	46.5	46.1	**47.41**	46.1	46.5
$SE17_C$	45.9	**47.3**	45.2	46.1	45.8	45.3	45.0	43.7	44.4	46.3	46.1	45.3	43.4	46.1	45.7	45.8	45.4	**47.1**
$MSPR_A$	69.2	68.4	68.6	68.1	70.0	68.1	**72.6**	68.9	68.0	69.1	70.3	70.7	69.2	67.8	71.5	70.6	69.1	70.8
$MSPR_C$	69.4	67.9	68.6	67.0	70.4	68.0	**72.7**	69.2	69.1	68.0	71.5	71.5	69.2	68.0	71.0	70.3	69.2	70.3
$SICK_A$	64.8	64.4	64.2	64.2	64.3	64.4	64.2	63.6	62.2	63.7	63.9	63.3	62.7	63.4	63.8	64.1	63.4	63.9
$SICK_C$	64.3	64.3	64.0	63.9	64.3	64.4	64.3	63.5	62.4	63.6	64.1	63.5	63.1	63.5	63.9	64.1	63.5	63.9
NUR_A	**65.1**	65.0	61.3	63.2	62.0	64.9	62.9	61.6	57.8	60.0	58.4	64.2	59.8	61.5	60.2	60.2	60.9	62.1
NUR_C	**65.6**	65.3	61.1	63.6	62.1	65.1	64.4	61.5	57.8	60.2	58.5	64.4	59.6	62.0	60.3	62.2	61.1	62.2

Table 3: Results of the meta-embedding SentEmb, using addition ($_A$) and concatenation ($_C$) along with pre-trained embeddings. $SE16$ and $SE17$ stand respectively for SemEval16 and SemEval17. Models names were also digested to match the page setup: Glove42B (Gl42), word2vec (w2v), CharSG (ChSG), Bow5C (B5C) and Bow5W (B5W).

Tasks	In domain embedding models									
	SkipGram				CharSG			Cbow		Glove
	Cbow	Glove	CharCBOW	CharSG	Cbow	Glove	CharCBOW	Glove	CharCBOW	CharCBOW
$SE16_A$	73.5	74.2	**75.4**	**75.5**	72.7	74.0	75.3	74.5	74.7	73.6
$SE16_C$	71.4	74.1	71.4	73.5	73.7	71.0	75.1	73.5	74.2	71.0
$SE17_A$	40.9	41.6	45.1	45.4	41.3	40.4	**46.0**	41.9	41.6	40.2
$SE17_C$	42.9	43.4	44.1	43.1	42.4	41.2	44.0	42.6	40.6	42.1
$MSPR_A$	64.7	62.2	67.5	**70.7**	63.6	62.7	68.1	61.1	63.4	62.4
$MSPR_C$	63.5	62.4	66.3	69.4	61.2	63.4	67.7	60.0	64.6	61.2
$SICK_A$	62.2	62.2	62.8	**63.9**	61.7	61.3	62.3	60.0	61.4	61.4
$SICK_C$	62.3	63.4	61.3	63.1	61.2	63.1	62.1	61.1	61.6	60.4
NUR_A	**66.5**	63.9	62.5	65.6	66.2	63.4	64.7	64.8	65.8	63.9
NUR_C	**66.5**	64.2	63.1	66.1	66.4	64.4	62.7	**66.7**	65.9	63.7

Table 4: Meta-Embedding results using addition ($_A$) and concatenation ($_C$) along with in-domain embedding models. $SE16$ and $SE17$ stand respectively for SemEval16 and SemEval17.

47.1%.

For the paraphrasing task (MSPR), the best meta-embedding model was CharSG with Glove42B (72.6% using addition and 72.7% using concatenation). Other interesting combinations can be observed such as CharSG with Deps (71.5% using concatenation) and Deps with word2vec (71.5% using addition), etc. Concerning the NUR task, the best meta-embedding model is Glove6B combined with Glove42B (65.1% using addition and 65.6% using concatenation) closely followed by Glove6B combined with word2vec (65.0% using addition and 65.3% using concatenation) and CharSG (64.9% using addition and 65.1% using concatenation). Surprisingly, the majority of other models failed to improve the performance of SentEmb. One particular remark is that the best meta-embeddings always involve the Glove models. Finally, no significant improvements were observed for the entailment task (SICK). This may be due to the task itself which consists of recognizing not only the entailment relation but also the opposite and the neutral relations. In this study, no particular attention was given to opposite and neutral labels. If the combination of different embedding models is useful for 3 out of 4 tasks, the nature of the data sets also plays an important role. Embedding models trained on different data sets may provide complementary information. This can be observed for instance when combining Glove6B (trained on Wikipedia and Gigaword) and Glove42B (trained on Common Crawl). An important observation regarding Tables 2 and 3 is that best individual models are not necessary the most appropriate for combination. For instance,

Data	Tasks				
Models	SemEval16	SemEval17	MSRP	SICK	NUR
	Map (%)	Map (%)	Accuracy	Accuracy	1 in 10 R@1
Best@1	**76.7** (1)	47.2 (3)	**80.4**	**84.6**	55.2
Best@2	76.0	46.9	77.4	83.6	48.8
Best@3	75.8	46.6	76.8	83.1	37.9
SentEmb	76.16	46.75	72.2	64.4	**68.5**
MetaSentEmbADD (In)	75.5	46.0	70.7	63.9	66.5
MetaSentEmbConcat (In)	74.1	44.0	69.4	63.1	66.7
MetaSentEmbADD (Out)	76.5 (2)	**47.4** (1)	72.6	64.8	65.1
MetaSentEmbConcat (Out)	76.4 (3)	47.3 (2)	72.7	64.3	65.6

Table 5: Results obtained by the 3 best state of the art models proposed during the official competitions of each task. Results obtained by the SentEmb baseline and with our proposed MetaSentEmb using addition and concatenation techniques over pre-trained and in-domain embeddings as well as their combinations.

Bow5W (rank 3 overall the four tasks) was less efficient while combined to other models, on the contrary, Glove42b which performed poorly individually (rank 10 overall the four tasks), turned out to be very efficient while combined to other models.

To study the impact of in-domain embeddings, we report in Table 4 the results of MetaSentEmb while using embeddings trained on the in-domain data set of each task. According to the results, we observe the same tendency as for pre-trained embeddings. However, the improvements seem to be task dependent. For instance, the best obtained results were 76.5% for pre-trained versus 75.5% (Semeval 2016) and 47.4% for pre-trained versus 46.0% (Semeval 2017) while for NUR task, the in-domain embedding obtained better results with 66.5% versus 65.6% for the pre-trained models. That said for the NUR results, the different is not significant. Generally speaking, the results of Tables 3 and 4 confirm the usefulness of using external data in addition to various embedding models and also put forward the possibility to combine embeddings trained on both in-domain and external data sets.

Table 5 reports the 3 best state of the art results obtained during the official competition of each task. Also, it contrasts the SentEmb baseline with our proposed MetaSentEmb using addition and concatenation techniques over pre-trained (Out) and in-domain (In) embeddings as well as their combinations. Below the header, the first horizontal frame reports the state of the art results. The second frame depicts results for similarity measures and the last frame contains results of classification-based approaches.

Globally, except for the NUR task, the meta-embedding configurations using pre-trained models are slightly better than the ones using in-domain models and enhance the performance of the SentEmb model. In particular, they outperform the SentEmb baseline and are ranked among the three best models for the SemEval tasks. Concerning the NUR task, the SentEmb baseline and all the meta-embedding models outperform the three best state of the art models. For this specific task, the combination of in-domain models give better results than out-of-domain models. Concerning the MSRP and the SICK tasks, while meta-embedding models build on out-of-domain corpora achieve better results than in-domain models, none of them succeeded in beating the state of the art models.

If additional efforts are certainly needed to understand the weak results on the entailment task and the different errors over all the evaluations, our observations through an error analysis showed different findings, depending on the task of course but also on the proposed method itself which is quite naive, especially for tasks like paraphrasing or entailment. First, MetaSentEmb performed well on the question-to-question similarity task and was competitive with regards to the best SemEval systems. This is certainly due to the adequacy of the task with our way of measuring sentence similarity. The questions in the Qatar Living corpus contain few ambiguities and the main errors were due to the specific forum vocabulary and mistakes that can be done by users. Also, one notable remark is the size of the original and related questions which is very important. Our way to deal with that was to filter stop-words and keep only nouns, verbs and adjectives to limit the impact of long sentences. Using POS-tagging also provided tagging errors that introduced some errors of our system. Second, MetaSentEmb did not compete with the three best systems on the paraphrasing task (MSPR), however, if we compare the results of MetaSentEmb with state of art sentence embedding representations such as FastSent (72.2%) and Skipthough (73.0%) (Hill et al., 2016), our approach obtained similar results (72.7%) with much simpler training. This finding is encouraging while no particular attention was given to the characteristics of paraphrase. Concerning textual entailment, MetaSentEmb failed to improve the performance of SentEmb. Here also the particularities of the small data set as well as the prediction of three classes including contradiction and neutral sentences may explain the low results. Finally, for the NUR task, our approach turned out to be very efficient. Utterance characteristics, at least for the Ubuntu corpus exhibit strong similarities which are certainly better captured by our meta-embedding approach.

6 Conclusion

In this paper we introduced the first bottom-up meta-embedding sentence representation for textual similarity. We have explored a variety of pre-trained and in-domain embedding models and there impact on question-to-question similarity, paraphrasing, textual entailment and next utterance ranking tasks. We have also proposed meta-embedding sentence representations based on vectors addition and concatenation and have shown under which conditions they can be jointly used for better performance. If further investigations are needed, the preliminary results lend support the idea that using meta-embeddings improve the performance of bottom-up sentence-based embedding approaches and offer an appropriate way to deal with textual similarity. One notable advantage of our approach is its simplicity, especially when using pre-trained embeddings since no computational cost is incurred.

7 Acknowledgments

The current work was supported by the ANR 2016 PASTEL (ANR-16-CE33-0007) project[6].

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

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.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR* abs/1607.04606. http://arxiv.org/abs/1607.04606.

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*.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '04.

N Enkvist. 1987. *Text linguistics for the applier: An orientation.*. In U. Connor R. Kaplan, Writing across languages: Analysis of L2 Text (pp. 23-44), Reading, MA: Addison-Wesley.

Justin Garten, Kenji Sagae, Volkan Ustun, and Morteza Dehghani. 2015. Combining distributed vector representations for words. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Denver, CO, USA, pages 95–101. http://www.aclweb.org/anthology/W15-1513.

Amir Hazem and Emmanuel Morin. 2017. Bilingual word embeddings for bilingual terminology extraction from specialized comparable corpora. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Asian Federation of Natural Language Processing, pages 685–693. http://aclweb.org/anthology/I17-1069.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *CoRR* abs/1602.03483. http://arxiv.org/abs/1602.03483.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2015. Learning to understand phrases by embedding the dictionary. *CoRR* abs/1504.00548. http://arxiv.org/abs/1504.00548.

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

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015a. Skip-thought vectors. *arXiv preprint arXiv:1506.06726* .

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015b. 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.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*. pages 302–308. http://aclweb.org/anthology/P/P14/P14-2050.pdf.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909* .

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*, pages 3111–3119.

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.

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 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.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*.

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.

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*.

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

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *ACL (1)*. The Association for Computer Linguistics. http://dblp.uni-trier.de/db/conf/acl/acl2016-1.htmlYinS16.