Enhancing Unsupervised Sentence Similarity Methods with Deep Contextualised Word Representations

Tharindu Ranasinghe, Constantin Orăsan and Ruslan Mitkov
Research Group in Computational Linguistics
University of Wolverhampton, UK
{t.d.ranasinghehettiarachchige, c.orasan, r.mitkov }@wlv.ac.uk

Abstract

Calculating Semantic Textual Similarity (STS) plays a significant role in many applications such as question answering, document summarisation, information retrieval and information extraction. All modern state of the art STS methods rely on word embeddings one way or another. The recently introduced contextualised word embeddings have proved more effective than standard word embeddings in many natural language processing tasks. This paper evaluates the impact of several contextualised word embeddings on unsupervised STS methods and compares it with the existing supervised/unsupervised STS methods for different datasets in different languages and different domains.

1 Introduction

Measuring Semantic Textual Similarity (STS) is calculating the degree of semantic equivalence between two snippets of text (Agirre et al., 2016). Earlier, STS tasks largely focused on similarity between short texts such as abstracts and product descriptions (Li et al., 2006; Mihalcea et al., 2006). Recently, STS tasks at the International Workshops on Semantic Evaluation (SemEval) focused on measuring STS between full sentence pairs. The introduction of competitive STS tasks led to the development of standard datasets like the SICK corpus (Bentivogli et al., 2016) and standardised the similarity score as a numerical value between 1 and 5 (Agirre et al., 2014).

Having a good STS metric is crucial for many natural language processing applications such as information retrieval (IR) (Majumder et al., 2016), text summarisation (Aliguliyev, 2009; Steinberger and Jezek, 2004), question answering (Mohler et al., 2011) and text classification (Rocchio, 1971). Semantic similarity also contributes to many semantic web applications like community extraction, ontology generation and entity disambiguation (Li et al., 2006), and it is also useful for Twitter search (Salton et al., 1997), where it is required to accurately measure semantic relatedness between concepts or entities (Xu et al., 2015). STS is not limited only to natural language processing. For example in Biomedical Informatics, it can be used to compare genes (Ferreira and Couto, 2010).

Given the growing importance of having a good STS metric and as a result of the SemEval workshops, researchers have proposed numerous STS methods. Most of the early approaches were based on traditional machine learning and involved heavy feature engineering (Béchara et al., 2015). With the advances of word embeddings, and as a result of the success neural networks have achieved in other fields, most of the methods proposed in recent years rely on neural architectures (Tai et al., 2015; Shao, 2017). Neural networks are preferred over traditional machine learning models as they generally tend to perform better than traditional machine learning models. They also do not rely on explicit linguistics features which have to be extracted before the ML model is learnt. Determining the best linguistic features for calculating STS is not an easy task as it requires a good understanding of the linguistic phenomenon and relies on researchers' intuition. In addition, calculating these features is usually not an easy task, especially for languages other than English. Therefore, in contrast to traditional ML methods, models based on neural networks can be easily applied to other languages.

However, the biggest challenge that the neural based architectures face when applied to STS tasks is the small size of datasets available to train them. As a result, in many cases the networks can-

Proceedings of Recent Advances in Natural Language Processing, pages 994–1003,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_115

not be trained properly. Given the amount of human labour required to produce datasets for STS, it is not possible to have high quality large training datasets. As a result researches working in the field have also considered unsupervised methods for STS. Recent unsupervised approaches use pretrained word/sentence embeddings directly for the similarity task without training a neural network model on them. Such approaches have used cosine similarity on sent2vec (Pagliardini et al., 2018), InferSent (Conneau et al., 2017), Word Mover's Distance (Kusner et al., 2015), Doc2Vec (Le and Mikolov, 2014) and Smooth Inverse Frequency with GloVe vectors (Arora et al., 2017). While these approaches have produced decent results in the final rankings of shared tasks, they have also provided strong baselines for the STS task.

Word vectors are used to determine a representation of a sentence in approaches like Word Mover's Distance (Kusner et al., 2015) and Smooth Inverse Frequency (Arora et al., 2017). The main weakness of word vectors is that each word has the same unique vector regardless of the context it appears. For an example, the word "play" has several meanings, but in standard word embeddings such as Glove (Pennington et al., 2014), FastText (Mikolov et al., 2018) or Word2Vec (Mikolov et al., 2013) each instance of the word has the same representation regardless of the meaning which is used. However, contextualised word embedding models such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2018) etc. generate embeddings for a word based on the context it appears, thus generating slightly different embeddings for each of its occurrence. The recent applications in areas such as question answering and textual entailment show that contextualised word embeddings perform better than the traditional word embeddings (Devlin et al., 2018).

This paper explores the performance of several contextualised word embeddings in three unsupervised STS methods - cosine similarity using average vectors, Word Mover's Distance (Kusner et al., 2015) and cosine similarity using Smooth Inverse Frequency (Arora et al., 2016). The rest of the paper is organised as follow. Section 2 contains information about the settings of the experiments carried out in this paper including the datasets employed here and the different contextualised word embedding models explored. Each of the contextualised word embedding models against each method are evaluated in Section 4. Further experiments are conducted on Spanish sentence similarity and Bio-medical sentence similarity to observe the portability of the model to other languages and domains in section 5. Section 6 would briefly describe the related work done for STS. The paper finishes with conclusions.

2 Settings of the Experiments

2.1 Data Sets

The experiments presented in this paper were carried out using several datasets which will be explained in next subsections. In order to prove the portability of the approaches, the proposed architectures were also tested on an English Biomedical STS dataset. In addition, the language independence of the method is tested by applying it to a Spanish STS dataset.

2.1.1 English-English STS Data Set

For the experiments carried out on English STS, we used the SICK dataset. (Bentivogli et al., 2016). The SICK data contains 9927 sentence pairs with a 5,000/4,927 training/test split which were employed in the SemEval tasks. Each pair is annotated with a relatedness score between 1 and 5, corresponding to the average relatedness judged by 10 different individuals. Table 1 shows a few examples from the SICK training dataset.

Sentence Pair	Similarity
1. A little girl is looking at a woman in costume. 2. A young girl is looking at a woman in costume.	4.7
1. A person is performing tricks on a motorcycle. 2. The performer is tricking a person on a motorcycle.	2.6
1. Someone is pouring ingredients into a pot. 2. A man is removing vegetables from a pot.	2.8
1. Nobody is pouring ingredients into a pot. 2. Someone is pouring ingredients into a pot.	3.5

Table 1: Example sentence pairs from the SICK training data

2.1.2 Spanish-Spanish STS Data Set

For the Spanish STS experiments we used the dataset provided for Spanish STS subtask in SemEval 2015 Task 2 (Agirre et al., 2015). The training set has 1250 sentence pairs annotated with a relatedness score between 0 and 4. There were two sources for test set - Spanish news and Spanish Wikipedia dump having 500 and 250 sentence pairs respectively. Both datasets were annotated with a relatedness score between 0 and 4. Table 2 shows few pairs of sentences with their similarity

score. As can be seen, this dataset is significantly smaller than the English dataset presented in the previous section. The effect of this is discussed in more detail below.

Sentence Pair	Similarity
1. Ams, los misioneros apunten que los nmberos d'infectaos puen ser shasta dos o hasta cuatro veces ms grandess que los oficiales. 2. Los cadveres de personas fallecidas pueden ser hasta diez veces ms contagiosos que los infectados vivos.	0.6
1. Desde Colombia, el presidente Juan Manuel Santos dijo que convers por telfono con Humala sobre el tema y que entregara al detenido a las autoridades peruanas a ms tardar el viernes. 2. El presidente de Colombia, Juan Manuel Santos, haba anunciado horas antes que Orellana, que se encuentra detenido, ser entregado a las autoridades peruanas sentre hoy y maanas.	3.2
1. La polica abati a un canbal cuando devoraba a una mujer Matthew Williams, de 34 aos, fue sorprendido en la madrugada mordiendo el rostro de una joven a la que haba invitado a su hotel. 2. La polica de Gales del Sur mat a un canbal cuando se estaba comiendo la cara de una mujer de 22 aos en la habitacin de un hotel.	2
1. Ollanta Humala se rene maana con el Papa Francisco. 2. El Papa Francisco mantuvo hoy una audiencia privada con el presidente Ollanta Humala, en el Vaticano.	3

Table 2: Example sentence pairs from the Spanish STS training data

2.1.3 Bio-medical STS Data Set

In-order to see the performance of our baseline in a complete different domain we used the biomedical English STS dataset provided in Soganciogl u et al. (2017). The dataset comprises 100 sentence pairs, which were evaluated by five different human experts that judged their similarity and gave scores ranging from [0,4]. To represent the similarity between two sentences we took the average of these scores. Table 3 shows few examples in the dataset. A dataset as small as this one can not be used by to train a supervised ML method, requiring alternative approaches such as unsupervised methods.

2.2 Contextualised Word Representations

In order to use words in machine learning models, words have to be represented with a numerical form. Over the years researches have used many word representations like bag of words, one hot encoded vectors etc. But the recent neural models like word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) provide better representations to the words considering its context too. We call them *standard word representations* in this research. Their main weakness is that every word has a unique word embedding regardless of the context it appears. As an example the word 'bank'

Sentence Pair	Similarity
1. It has recently been shown that Craf is essential for Kras G12D-induced NSCLC. 2. It has recently become evident that Craf is essential for the onset of Kras-driven non-small cell lung cancer.	4
1. Up-regulation of miR-24 has been observed in a number of cancers, including OSCC. 2. In addition, miR-24 is one of the most abundant miRNAs in cervical cancer cells, and is reportedly up-regulated in solid stomach cancers.	3
1. These cells (herein termed TLM-HMECs) are immortal but do not proliferate in the absence of extracellular matrix (ECM) 2. HMECs expressing hTERT and SV40 LT (TLM-HMECs) were cultured in mammary epithelial growth medium (MEGM, Lonza)	1.4
1.The up-regulation of miR-146a was also detected in cervical cancer tissues. 2. Similarly to PLK1, Aurora-A activity is required for the enrichment or localisation of multiple centrosomal factors which have roles in maturation, including LATS2 and CDK5RAP2/Cnn.	0.2

Table 3: Example sentence pairs from the Biomedical dataset

in two sentences - "I am walking by the river bank" and "I deposited money to the bank" would have the same embeddings which can be confusing for machine learning models. The recent introduction of contextualised word representations solved this problem by providing vectors for words considering their context too. In this way the word 'bank' in above sentences have two different embeddings. As a result, contextualised word embeddings perform better than standard word embeddings in many natural language processing tasks like question answering, textual entailment etc. (Devlin et al., 2018). The following contextualised words representation models were considered for the experiments.

2.2.1 ELMo

ELMo introduced by Peters et al. (2018) use bidirectional language model (biLM) to learn both word (e.g., syntax and semantics) and linguistic context. After pre-training, an internal state of vectors can be transferred to downstream natural language processing tasks. We used the 'original' pre-trained model provided in Peters et al. (2018) which was trained on the 1 Billion Word Benchmark (Chelba et al., 2013), approximately 800M tokens of news crawl data from WMT 2011. Using the model we represented each word as a vector with a size of 3072 values.

2.2.2 BERT

BERT was introduced in Devlin et al. (2018). It is based on a bidirectional transformer architecture rather than a unidirectional transformer used in Open AI GPT (Radford et al., 2019). In contrast to ELMo which uses a shallow concatenation layer (Devlin et al., 2018), BERT employs a deep concatenation layer. As a result BERT is considered a very powerful embedding architecture. We used pre-trained 'bert-large-uncased' model and represented each word as a 4096 lengthened vector.

2.2.3 Stacked Embeddings

Stacked Embeddings are obtained by concatenating different embeddings. According to Akbik et al. (2019) stacking the embeddings can provide a powerful embeddings to represent words. We represent the stacked embeddings in section 4 with '+' between the used models. As an example if the model name says ELMo + BERT, it is a stacked embedding of ELMo and BERT. For ELMo + BERT model we used pre-trained 'bert-large-uncased' model and 'original' pre-trained ELMo model to represent each word as a 4096 + 3072 vector.

2.2.4 Flair

Flair is another type of popular contextualised word embeddings introduced in Akbik et al. (2018). It takes a different approach by using a character level language model rather than the word level language model used in ELMo and BERT. The recommended way to use Flair embeddings is to stack pre-trained 'news-forward' embeddings and pre-trained 'news-backward' embeddings with Glove (Pennington et al., 2014) word embeddings (Akbik et al., 2018). We used the stacked model to represent each word as a 4196 lengthened vector.

2.3 Standard Word Representations

In order to compare the results of contextualised word embeddings, we used a standard word representation model in each experiment as a baseline. In this research we used word2vec embeddings (Mikolov et al., 2013) pre-trained on Google news corpus. We represented each word as a 300 lengthened vector using this model.

3 Experiments

This section describes the actual methods used to calculate the STS score between a pair of sentences and their variants we used. Each experiment was conducted using all three contextualised word embedding models - ELMo, BERT and Flair and one standard word representation model - word2vec (Mikolov et al., 2013).

3.1 Cosine Similarity on Average Vectors

The first unsupervised STS method that we used to estimate the semantic similarity between a pair of sentences, takes the average of the word embeddings of all words in the two sentences, and calculates the cosine similarity between the resulting embeddings. This is a common way to acquire sentence embeddings from word embeddings. Obviously, this simple baseline leaves considerable room for variation. We have investigated the effects of ignoring stopwords and computing an average weighted by tf-idf in particular and reported them in the 4 section.

3.2 Word Mover's Distance

The second baseline that we have considered is Word Mover's Distance introduced by Kusner et al. (2015). Word Mover's Distance uses the word embeddings of the words in two texts to measure the minimum distance that the words in one text need to "travel" in semantic space to reach the words in the other text as shown in Figure 1. Kusner et al. (2015) says that this is a good approach than vector averaging since this technique keeps the word vectors as it is through out the operation. We have investigated the effects of considering/ ignoring stop words before calculating the word mover's distance.

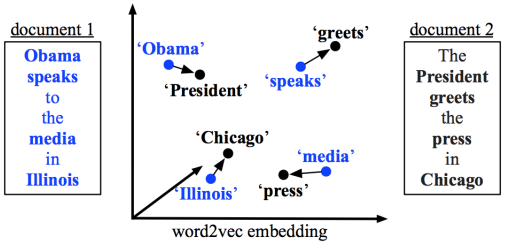

Figure 1: The Word Mover's Distance between two documents

3.3 Cosine Similarity Using Smooth Inverse Frequency

The third and the last unsupervised STS method we have considered is to acquire sentence embeddings using Smooth Inverse Frequency pro-

posed by Arora et al. (2016) and then calculate the cosine similarity between those sentence embeddings. Semantically speaking, taking the average of the word embeddings in a sentence tends to give too much weight to words that are quite irrelevant. Smooth Inverse Frequency tries to solve this problem in two steps.

1. Weighting: Smooth Inverse Frequency takes the weighted average of the word embeddings in the sentence. Every word embedding is weighted by $\frac{a}{a+p(w)}$, where a is a parameter that is typically set to 0.001 and $p(w)$ is the estimated frequency of the word in a reference corpus.

2. Common component removal: After that, Smooth Inverse Frequency computes the principal component of the resulting embeddings for a set of sentences. It then subtracts their projections on first principal component from these sentence embeddings. This should remove variation related to frequency and syntax that is less relevant semantically.

As a result, Smooth Inverse Frequency downgrades unimportant words such as *but, just*, etc., and keeps the information that contributes most to the semantics of the sentence. After acquiring the sentence embeddings for a pair of sentences, the cosine similarity between those two vectors were taken to represent the similarity between them.

4 Evaluation on English SemEval Data

This section describes the evaluation results of English SemEval data for all the unsupervised STS methods we described above.

All the experiments were evaluated using the three evaluation metrics normally employed in the STS tasks: Pearson correlation (τ), Spearman correlation (ρ) and Mean Squared Error (MSE). Following sub-sections will discuss the results in detail.

4.1 Cosine Similarity on Average Vectors

Vector averaging results are shown in Table 4. Since we calculated the similarity as the cosine similarity between two vectors our predicted similarity lies between $\in$ [0,1]. Since the GOLD standards are between $\in$ [1,5] we re-scaled the predictions to be $\in$ [1,5] in order to allow comparison.

Following variations were considered and reported in each sub-table.

1. All the word vectors were considered for averaging. Results are shown in table 4a.

2. All the word vectors except the vectors for stop words were considered for averaging. Table 4b shows the results.

3. All the word vectors were weighted from its tf-idf scores and considered averaging. Results are shown in table 4c

4. Stop words were removed first and remaining word vectors were weighted from its tf-idf scores and considered averaging. Table 4d shows the results.

As shown in table 4 the contextualised word vectors did not perform better than the standard word embeddings in all the variations. The only model that came close to word2vec performance was ELMo. All the contextualised word embedding models we considered have more than 3000 dimensions for the word representation which is significantly higher than the number of dimensions for the word representation we had for standard embeddings - 300. As the vector averaging model is highly dependent on the number of dimensions that a vector can have, the curse of dimensionality might be the reason for the poor performance of contextualised word embeddings.

4.2 Word Mover's Distance

The results for the Word Mover's Distance is shown in 5. Following variations were considered and reported in each sub-table.

1. Considering all the words to calculate the Word Mover's Distance. Results are shown in 5a

2. Removing stop words before calculating the Word Mover's Distance. Table 5b shows the results.

As depicted in table 5a contextualised word representations could not improve Word Mover's method too over standard word representations. Since the travelling distance is dependent on number of dimensions, the curse of dimensionality might be the reason for the poor performance of contextualised word representations in this scenario too.

Embedding	τ	ρ	MSE
Word2vec	**0.732**	**0.624**	**1.664**
ELMo	0.655	0.592	1.863
Flair	0.632	0.559	3.348
BERT	0.584	0.591	3.258
ELMo + BERT	0.654	0.612	2.789

(a) Averaging all the word vectors

Embedding	τ	ρ	MSE
Word2vec	**0.720**	**0.585**	**1.440**
ELMo	0.676	0.597	1.729
Flair	0.668	0.561	2.235
BERT	0.646	0.607	2.958
ELM0 + BERT	0.693	0.620	2.496

(b) Averaging all the word vectors removing stop words

Embedding	MSE	τ	ρ
Word2vec	**0.708**	**0.581**	**1.311**
ELMo	0.675	0.589	1.600
Flair	0.657	0.547	2.074
BERT	0.596	0.575	2.890
ELMo + BERT	0.661	0.594	2.387

(c) Averaging all the word vectors weighting them with tf-idf

Embedding	τ	ρ	MSE
Word2vec	**0.705**	**0.565**	**1.300**
ELMo	0.669	0.582	1.550
Flair	0.661	0.545	1.809
BERT	0.591	0.569	2.739
ELMo + BERT	0.656	0.587	2.250

(d) Averaging all the word vectors weighting them with tf-idf removing stop words

Table 4: Vector averaging results for SICK training set

Embedding	τ	ρ	MSE
Word2vec	**0.642**	**0.593**	**1.051**
ELMo	0.584	0.559	1.210
Flair	0.592	0.561	1.166
BERT	0.605	0.578	1.145
ELMo + BERT	0.595	0.568	1.189

(a) Considering all the word vectors

Embedding	τ	ρ	MSE
Word2vec	**0.636**	**0.573**	**1.156**
ELMo	0.600	0.549	1.416
Flair	0.615	0.557	1.254
BERT	0.639	0.580	1.177
ELMo + BERT	0.619	0.565	1.299

(b) Considering all the word vectors removing stop words

Table 5: Word moving distance results for SICK training set

4.3 Cosine Similarity Using Smooth Inverse Frequency

Table 6 shows the results for the Smooth Inverse Frequency method. As shown there, all the contextualised word representations have improved the results significantly over the standard word representations. Since the first principle component is removed in the process, curse of dimensionality has not affected this method. The stacked embeddings of ELMo and BERT provided the best results to the experiment. Also, it is important to notice that the Smooth Inverse Frequency using the stacked embeddings of ELMo and BERT showed the best results from all three methods for all three evaluation metrics.

As shown in the above tables, contextualised word embeddings did not improve the results of vector averaging and word movers distance. But contextualised word embeddings showed a great improvement over standard word embeddings in Smooth Inverse Frequency STS method which also provided the best results among the considered unsupervised STS methods.

Embedding	τ	ρ	MSE
Word2vec	0.734	0.632	0.604
ELMo	0.740	0.654	0.593
Flair	0.731	0.634	0.601
BERT	0.746	0.661	0.456
ELMo + BERT	**0.753**	**0.669**	**0.446**

Table 6: Smooth Inverse Frequency results for SICK training set

4.4 Further Experiments and Results

As shown in the above section Smooth Inverse Frequency with ELMo and BERT stacked contex-

tualised word representations provided the best result. However, since we used the cosine similarity between two vectors, the predictions of our model are constrained to follow the cosine curve and are thus not suited for these evaluation metrics. For this reason, we applied a parametric regression step to obtain better-calibrated predictions. We trained a regression model on the SICK train data and predicted on the SICK test data. This calibration step served as a minor correction for our restrictively simple similarity function. However, this regression calibration improved the Pearson correlation by 0.01 for the SICK test set.

Our unsupervised method had 0.762 Pearson correlation score, whilst the best result in the International Workshop on Semantic Evaluation 2014 Task 1 had 0.828 Pearson correlation (Marelli et al., 2014). Our approach would be ranked on the ninth position from the top results out of 18 participants, and it is the best unsupervised STS method among the results. Our method even outperformed systems that rely on additional feature generation (e.g. dependency parses) or data augmentation schemes. As an example, our method is just above the UoW system which relied on 20 linguistics features fed in to a Support Vector Machine and obtained a 0.714 Pearson correlation (Gupta et al., 2014). Compared to these complex approaches our simple approach provides a strong baseline to STS tasks.

5 Portability of the Method to Other Languages and Domains

Our approach has the advantage that it does not rely on language dependent features and it does not need a training set as the approach is unsupervised. As a result, the approach is easily portable to other languages and domains given the availability of ELMo and BERT models in that particular language or domain. In order to observe how well the method performs in other languages and domains we applied it to Spanish STS dataset and Biomedical STS dataset described in section 3.

5.1 Spanish STS

We run all the unsupervised STS methods described in section 2 on the Spanish STS dataset explained in section 2.1.2. For the ELMo embeddings we used Spanish ELMo embeddings provided in Che et al. (2018), while for the BERT embeddings we used "BERT-Base, Multilingual Cased" [1] model which has been built on the top 100 languages with the largest Wikipedias which includes Spanish language too.

The predictions from the experiment were rescaled to lie $\in$ [0,4] as the GOLD standards. Organisers have used only one evaluation metric in this Spanish STS task: Pearson correlation (τ) against the predictions and GOLD standard. They have calculated Pearson correlation for each test set: Spanish news and Spanish wiki, separately and has taken the weighted average to give the final rankings in the leader board. We took the same procedure in order to evaluate our approach with the other approaches in the task. Also we applied parametric regression step we did to English-English STS experiment to obtain better-calibrated predictions. Parametric regression step improved the Pearson correlation by 0.01 for both Wikipedia and Newswire datasets.

From the experiments, Smooth Inverse Frequency with ELMo and BERT stacked embeddings gave the best results, similar to the English STS experiments we conducted. Our approach had 0.660 Pearson correlation for Wikipedia dataset, 0.547 Pearson correlation for Newswire dataset and 0.570 weighted mean from both of them. The best performing model that participated in SemEval 2015 task 2, had 0.705 Pearson correlation for Wikipedia, 0.683 for Newswire and 0.690 weighted mean (Agirre et al., 2015). Our approach would rank fifth out of 17 team in the final results, which is the best result for an unsupervised approach. As with the English model, this one also surpasses other complex supervised models. As an example RTM-DCU-1stST.tree uses a supervised machine learning algorithm with Referential Translation Machines(Biici and Way, 2014) and our fairly simple unsupervised approach outperform them by a significant margin. Comparing the results we can safely assume that our approach works well with Spanish language STS too.

5.2 Bio-Medical STS

In order to evaluate our approach in a different domain, we experimented it on Bio-medical STS dataset explained in 2.1.3. As in the previous experiments we applied all unsupervised approaches mentioned. We used ELMo embeddings trained on a biomedical domain corpora (e.g., PubMed abstracts, PMC full-text articles) (Peters et al.,

[1] https://github.com/google-research/bert

2018) and BioBERT: BERT embeddings trained on biomedical domain corpora (Lee et al., 2019). We did not apply Parametric regression step to this dataset since there was not enough data for the training. The predictions from the experiment were re-scaled to lie $\in$ [0,4] as the GOLD standards. Organisers have used only one evaluation metric in this Bio-medical STS task: Pearson correlation (τ) against the predictions and GOLD standard.

Same as English and Spanish experiments, Smooth Inverse Frequency with ELMo and BERT stacked embeddings performed best with this dataset too. It had 0.680 Pearson correlation, whilst the best performing method had 0.836 Pearson correlation. This would rank our approach seventh out of 22 teams in the final results of the task (Soganciogl u et al., 2017). It should be also noted that it outperforms many complex methods that sometimes uses external tools too. As an example, the UBSM-Path approach is based ontology based similarity which uses METAMAP (Aronson, 2001) for extracting medical concepts from text and our simple unsupervised approach outperform them by a significant margin. UBSM-Path only has 0.651 Pearson correlation. Comparing the results we can safely assume that our approach works well in bio medical domain too.

6 Related Work

Given that a good STS metric is required for a variety of natural language processing fields, researchers have proposed a large number of such metrics. Before the shift of interest in neural networks, most of the proposed methods relied heavily on feature engineering. With the introduction of word embedding models, researchers focused more on neural representation for this task.

There are two main approaches which employ neural representation models: supervised and unsupervised. Unsupervised approaches use pre-trained word/sentence embeddings directly for the similarity task without training a neural network model on them while supervised approaches uses a machine learning model trained to predict the similarity using word embeddings. ConvNet (He et al., 2015), Skip Thought vectors (Kiros et al., 2015), Dependency Tree-LSTM (Tai et al., 2015) and Siamese Neural Networks (Mueller and Thyagarajan, 2016) can be considered as the most successful architectures employed for calculating STS. These supervised approaches always suffer from less training data problem which is common in STS tasks. As a result the researches have also considered unsupervised approaches.

The three unsupervised STS methods explored in this paper: Cosine similarity on average vectors, Word Mover's Distance and Cosine similarity using Smooth Inverse Frequency are the most common unsupervised methods explored in STS tasks. Apart from them cosine similarity of the output from Infersent (Conneau et al., 2017), sent2vec (Pagliardini et al., 2018) and doc2vec (Le and Mikolov, 2014) have been used to represent the similarity between two sentences. All these approaches relies on pre-trained sentence embeddings.

7 Conclusions

This paper experimented three unsupervised STS methods namely cosine similarity using average vectors, Word Mover's Distance and cosine similarity using Smooth Inverse Frequency with contextualised word embeddings for calculating semantic similarity between pairs of texts and compared them with other unsupervised/ supervised approaches. Contextualised word embeddings could not improve cosine similarity using average vectors and Word Mover's Distance methods, but the results when using Smooth Inverse Frequency method were improved significantly with contextualised word embeddings, instead of standard word embeddings. Further more we learned that stacking ELMo and BERT provides a strong word representation rather than individual representations of ELMo and BERT. The results indicated that calculating cosine similarity using Smooth Inverse Frequency with stacked embeddings of ELMo and BERT is the best unsupervised method from the available approaches. Also, our approach finished on the top half of the final results list surpassing many complex and supervised approaches.

Our approach was also applied in the Spanish STS and Bio-medical STS tasks, where our simple unsupervised approach finished on the top half of the final result list in both cases. Therefore, given our results we can safely assume that regardless of the language or the domain cosine similarity using Smooth Inverse Frequency with stacked embeddings of ELMo and BERT will provide a simple but strong unsupervised method for STS tasks.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *SemEval@NAACL-HLT*.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *SemEval@COLING*.

Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval@NAACL-HLT*.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. page to appear.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*. pages 1638–1649.

Ramiz M. Aliguliyev. 2009. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Syst. Appl.* 36:7764–7772.

Alan R. Aronson. 2001. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. *Proceedings. AMIA Symposium* pages 17–21.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings .

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR 2017*.

Hanna Béchara, Hernani Costa, Shiva Taslimipoor, Rohit Gupta, Constantin Orasan, Gloria Corpas Pastor, and Ruslan Mitkov. 2015. Miniexperts: An svm approach for measuring semantic textual similarity. In *SemEval@NAACL-HLT*.

Luisa Bentivogli, Raffaella Bernardi, Marco Marelli, Stefano Menini, Marco Baroni, and Roberto Zamparelli. 2016. Sick through the semeval glasses. lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *Language Resources and Evaluation* 50:95–124.

Ergun Biici and Andy Way. 2014. Rtm-dcu: Referential translation machines for semantic similarity. In *SemEval@COLING*.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, Brussels, Belgium, pages 55–64. http://www.aclweb.org/anthology/K18-2005.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

João D. Ferreira and Francisco M. Couto. 2010. Semantic similarity for automatic classification of chemical compounds. In *PLoS Computational Biology*.

Rohit Gupta, Hanna Béchara, Ismaïl El Maarouf, and Constantin Orasan. 2014. Uow: Nlp techniques developed at the university of wolverhampton for semantic similarity and textual entailment. In *SemEval@COLING*.

Hua He, Kevin Gimpel, and Jimmy J. Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *EMNLP*.

Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *ICML*.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR* abs/1901.08746.

Yuhua Li, David McLean, Zuhair Bandar, James O'Shea, and Keeley A. Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering* 18:1138–1150.

Goutam Majumder, Partha Pakray, Alexander F. Gelbukh, and David Pinto. 2016. Semantic textual similarity methods, tools, and applications: A survey. *Computación y Sistemas* 20.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *SemEval@COLING*.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Michael Mohler, Razvan C. Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *ACL*.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI*.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *NAACL-HLT*.

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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners .

J. J. Rocchio. 1971. Relevance feedback in information retrieval. In G. Salton, editor, *The Smart retrieval system - experiments in automatic document processing*, Englewood Cliffs, NJ: Prentice-Hall, pages 313–323.

Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. Automatic text structuring and summarization. *Inf. Process. Manage.* 33(2):193–207. https://doi.org/10.1016/S0306-4573(96)00062-3.

Yang Shao. 2017. Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *SemEval@ACL*.

Gizem Soganciogl u, Hakime Öztürk, and Arzucan Özgür. 2017. Biosses: a semantic sentence similarity estimation system for the biomedical domain. In *Bioinformatics*.

Josef Steinberger and Karel Jezek. 2004. Using latent semantic analysis in text summarization and summary evaluation.

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

Wei Xu, Chris Callison-Burch, and William B. Dolan. 2015. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *SemEval@NAACL-HLT*.