AlexU-AUX-BERT at SemEval-2020 Task 3: Improving BERT Contextual Similarity Using Multiple Auxiliary Contexts

Somaia Mahmoud
Alexandria University
Egypt
somaia.0246@gmail.com

Marwan Torki
Alexandria University
Egypt
mtorki@alexu.edu.eg

Abstract

This paper describes the system we built for SemEval-2020 task 3. That is predicting the scores of similarity for a pair of words within two different contexts. Our system is based on both BERT embeddings and WordNet. We simply use cosine similarity to find the closest synset of the target words. Our results show that using this simple approach greatly improves the system behavior. Our model is ranked 3rd in subtask-2 for SemEval-2020 task 3.

1 Introduction

A word meaning can be affected by its context. For polysemous words, the change in meaning is usually clear and evident. But even for words that are not necessarily considered polysemous, there can be subtle changes in their meaning. In SemEval-2020 Task 3 (Armendariz et al., 2020a), one goal is to predict how similar two words are in a given context, specifically given two different shared contexts for each pair of target words.

Identifying the meaning of words in context is known as Word Sense Disambiguation (WSD). It is a core task of Natural Language Processing (NLP) and has many potential applications. In (Navigli, 2009), the authors presented the motivations for solving the ambiguity of words.

Example 1	
Context 1	Small **arms** include handguns, **rifles**, machine guns, etc.
Context 2	He stretched his **arms** and **rifled** through the drawer

Example 2	
Context 1	He proposed a simple **solution** to **solve** the problem.
Context 2	He promised him to **solve** his school debt if he found the right ratios of the chemical **solution**.

Figure 1: Input example: There are two target words (boldfaced). These target words are presented in two different contexts.

We have participated in Task-3 Subtask-2. In this task, given two target words in a shared context, we want to score how similar they are. We then put the same two words in a different context and re-score their similarity. The input to our system is a pair of words within two different contexts. The output would be the two similarity scores, one for each context.

Figure 1 shows the input examples. We consider the words ‘rifle’ and ‘arm’ as our target words. In the first context, the words refer to the same meaning, a weapon. In the second context, they refer to different senses, (‘arm’ as a body part, and ‘rifle’ meaning ‘to search’). Similarly, in the first context, the words ‘solution’ and ‘solve’ have the same sense, solving a problem. In the second context, ‘solution’ means a chemical mixture and ‘solve’ means to clear a debt.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 270–274
Barcelona, Spain (Online), December 12, 2020.

In this paper, we tackle the problem of multiple senses for target words. Inspired by (Levine et al., 2019), we compose multiple embeddings for target words from sets of synonymous words using BERT (Devlin et al., 2018). To obtain multiple embeddings for the target words, we use WordNet (Fellbaum and others, 1998) to get the different senses of the target words. We put these different senses into auxiliary contexts such that we can compute different embedding using BERT again. Given the multiple word contexts, we find the closest sense of the target word and fuse it with its original contextual BERT embedding. This fusion results in a new embedding for the target word that is more representative of its meaning.

By incorporating word senses information into the word embedding, the harmonic mean of Pearson and Spearman correlations improved from 0.573 when we use the BERT embedding baseline to 0.723 when evaluated on SemEval-2020 Task3 dataset (Armendariz et al., 2020b). Our system was ranked third among the competing systems.

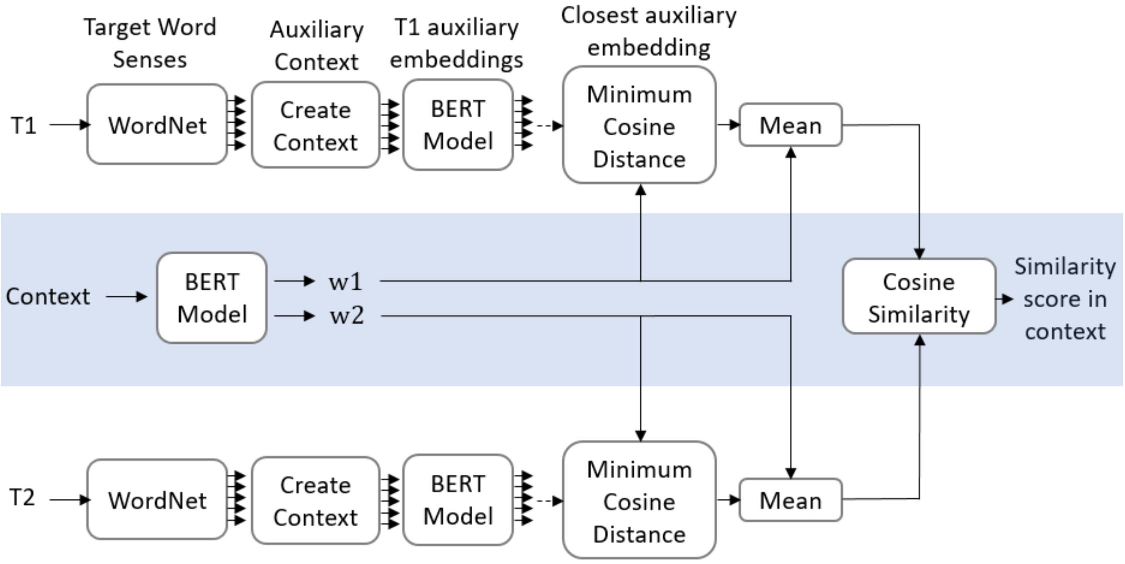

Figure 2: System Illustration: The process is repeated for each shared context. The shaded part shows the BERT baseline. Our approach generates K auxiliary contexts for each target word. Then we compute K different BERT embeddings for each target word given their corresponding auxiliary contexts. We average the target word embedding with the most similar out of the K auxiliary embeddings. Finally, we compute the cosine similarity between the target words based on the new embedding.

2 System Overview

Figure 2 illustrates how our system works, and the pipeline is described below:

1. We extract the contextualized embeddings of the target words T_1 and T_2. This is done by obtaining the embedding for the whole context through BERT (Devlin et al., 2018). Hence we obtain two vectors w_1 and w_2.

2. We use WordNet to get the senses of the target words. WordNet is a lexical database that links words into semantic relations, such as synonyms and hypernyms. Synonymous words that correspond to the same sense are grouped into synsets with short definitions and usage examples.

$$T_1.Senses = \{WordNet.synsets(T1)\}$$

3. For each target word, we select the top K synsets. We use the definition included in the synset to create an auxiliary context for the target word, as shown below.

$$T_1.AuxContext(i) = T_1 + \textbf{`is'} + T_1.Senses(i).definition() \qquad 1 < i < K$$

Example:

Target word: solution

Definition: the successful action of solving a problem

Auxiliary Context: ***Solution*** *is the successful action of solving a problem*

4. We then run the created auxiliary context through BERT to get an embedding for the target word. For the selected K senses, we get K embeddings. We select the sense that has minimum cosine distance with the embedding from the original context.

$$w_{1Aux}(i) = BERT(T_1.AuxContext(i))$$

$$T_1.Dist(i) = 1 - \frac{< w_1, w_{1Aux}(i) >}{max(\|w_1\|_2 \ \|w_{1Aux}(i)\|_2, \in)}$$

$$w_1^* = w_{1aux}(\textbf{argmin}_i(T_1.Dist))$$

5. We then adjust the new representation for the target word to the mean of its embedding from the original context and the auxiliary context.

$$w_{1Final} = Mean(w_1, w_1^*)$$

6. Steps 2-5 are repeated for the second target word. Finally, the similarity score between the two target words is the cosine similarity between their final embedding vectors.

$$Sim(T1, T2) = \frac{< w_{1Final}, w_{2Final} >}{max(\|w_{1Final}\|_2 \ \|w_{2Final}\|_2, \in)}$$

3 Experimental Setup

We use pre-trained BERT base embeddings from PyPI[1]. To get word senses, we use the WordNet interface from NLTK[2].

As an evaluation metric, we use the harmonic mean of the Pearson and Spearman correlations against the gold scores by human annotators. We evaluate our technique on SemEval-2020 Task3 evaluation data and the Stanford Contextual Word Similarity (SCWS) dataset (Huang et al., 2012). We used the (SCWS) dataset to evaluate our different models since the practice data for the task was very small in size.

We evaluated our method on English data only. SemEval-2020 Task3 evaluation data covers three other languages; Croatian, Finnish, and Slovenian. Unfortunately, the tool we used to access the wordnets does not support these languages. We can use other tools to access the wordnets[3]. It would be interesting to apply our method to other languages and see the performance.

3.1 Contextual embedding baseline

We compare our technique against a simple baseline that uses BERT. It computes the cosine similarity between BERT embeddings of the two target words and uses it as the similarity score. The highlighted part in Figure 2 is a simple illustration of the baseline.

[1] https://pypi.org/project/bert-embedding/

[2] https://www.nltk.org/howto/wordnet.html

[3] http://compling.hss.ntu.edu.sg/omw/

	SCWS	SemEval
BERT (Devlin et al., 2018)	0.66	0.573
AuxBERT$_{concat}$	0.67	0.695
AuxBERT(official)	0.692	0.719
AuxBERT$_{mean}$	**0.692**	**0.723**

Table 1: Results on the SCWS dataset and SemEval evaluation dataset. Using BERT embeddings of the original context (BERT). And using BERT embeddings of both the original and auxiliary contexts (AuxBERT). The scores are the harmonic mean of the Pearson and Spearman correlations.

4 Results

As shown in Table 1, the techniques that used the information about the senses of the target words got better scores. Also, it shows that averaging the embedding vectors from the original and auxiliary contexts gives better results than concatenation.

AuxBERT(official) in Table 1 refers to the official submitted result during the evaluation phase. This result was obtained by selecting the word sense from the top-five senses. Further experiments showed that having the top-three senses to choose from is a better choice.

Table 2 compares our results on the SCWS dataset and the results reported by (Huang et al., 2012). Using our technique, the score increased from 65.7 to 68.1.

	Spearman Correlation
(Huang et al., 2012)	65.7
AuxBERT	68.1

Table 2: Spearman's correlation on the SCWS dataset

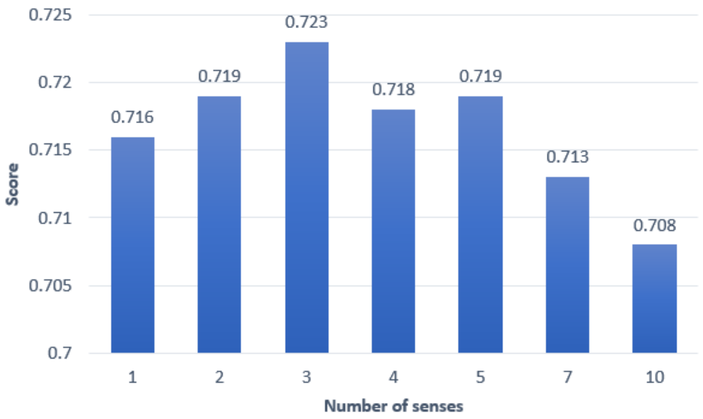

Figure 3: Harmonic mean of Pearson and Spearman correlations using different number of word senses. Setting K=3 provides the best score for our approach.

4.1 Post-evaluation

In this section, we discuss the experiments we did in the post-evaluation phase of SemEval-2020. We discuss how changing the number of word senses affect the results. And we show the results of testing our technique on subtask-1.

Selecting the closest word sense from the top-three senses is the best choice. Figure 3 shows that choosing from a larger or smaller number of senses reduces the score. This could be because the top-three senses are the most frequently used for most words. Hence, choosing from a group larger than three gives more chance for error.

	Pearson Correlation
BERT (Devlin et al., 2018)	0.713
AuxBERT	0.76

Table 3: Pearson correlation for Subtask-1: Predicting Change of Similarity scores

By subtracting the similarity scores of the two contexts, we get the change in similarity. That is the objective in subtask-1. Table 3 shows our results against the BERT baseline.

5 Conclusion

We presented a system that requires no training and uses a simple method to generate word embeddings that represent the word sense. We improved BERT embeddings by using information about the word senses. We varied the number of word senses from which we choose the closest sense and found the optimum number that reduces the error. We experimented with different aggregation methods. By using our technique, the score increased from 0.573 when we use the BERT embedding baseline to 0.723 when evaluated on SemEval dataset.

References

Carlos S. Armendariz, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Marko Robnik-Šikonja, Ivan Vulić, and Mohammad Taher Pilehvar. 2020a. SemEval-2020 task 3: Graded word similarity in context (GWSC). In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Carlos S. Armendariz, Matthew Purver, Matej Ulčar, Senja Pollak, Nikola Ljubešić, and Mark Granroth-Wilding. 2020b. CoSimLex: A resource for evaluating graded word similarity in context. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5878–5886, Marseille, France, May. European Language Resources Association.

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*.

Christiane Fellbaum et al. 1998. Wordnet: An electronic lexical database mit press. *Cambridge, Massachusetts*.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2019. Sensebert: Driving some sense into bert. *arXiv preprint arXiv:1908.05646*.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.