Comparing Contextualised Embeddings for Predicting the (Graded) Effect of Context in Word Similarity

Joris Albers - 6400507
Bachelor Artificial Intelligence UU, 7.5 ECTS
Yupei Du, Dong Nguyen

2 July 2021

Contents

1 Abstract

In this research I examined the differences in different contextualized word embedding models for predicting the graded effect of context in word similarity. Each model was tested on a task in which the degree and direction of change of word similarity of two word pairs had to be predicted. This was later compared to human annotated scores. I found that the BERT architecture works exceptionally well compared to other Transformer based models on the English language. I also found that the multilingual BERT models offers obtains the best score on the low resource languages Finnish, Hungarian and Slovenian. Furthermore I found that stacked embeddings, in which multiple models get combined, offer room for improvement for already performing models. At last I recommend further research in which more models can get compared and the further examination of stacked embeddings.

2 Introduction

The meaning of words can differ greatly due to its surroundings or context words, especially for polysemous words. For instance, when someone is speaking about a "prison cell", *cell* implies a room in a building. However, when *cell* occurs in a biological context, it most probably refers to the basic structural unit of organisms. Therefore, understanding the effect of context in word semantics is crucial for natural language processing (NLP) research, from upstream tools like (contextualized) word embeddings, to downstream systems including machine translation and dialog systems.

While earlier research has mainly focused on predicting the discrete difference between polysemous words depending on context (i.e. word sense disambiguation, Huang et al., 2019), there were not much research on predicting the their continuous effects on words' meaning. More research into this continuous effect on meaning can help further understand embedding models and their workings.

In recent years, transformer [Vaswani et al., 2017] based contextualized word embedding models (e.g. BERT, Devlin et al., 2018; GPT-2, Alammar, 2019) have led to substantial performance gains on various NLP tasks. They also achieve success in predicting the context effect of word similarities. For example, Hettiarachchi and Ranasinghe [2020] use BERT model to predict the contextualised word similarities, and get top-5 rankings in SemEval 2020 Task 3 [Armendariz et al., 2020].

Despite their great success, to the best of our knowledge, little effort has been made to compare the performances of different contextualized word embedding models on this task. At the same time, comparing and analyzing their differences can help us to get better understanding of not only the strengths and weakness of different models, but also the requirements of solving this task.

To this end, based on model architecture of Hettiarachchi and Ranasinghe [2020], in this work, I compare the performances of various contextualized word

embedding models, including BERT, RoBERTa, GPT-2 and stacked embedding models in this task. In experiments, I find that the BERT architecture works the best on this task, with RoBERTa scoring the highest scores on the English language and a multilingual version scoring the best on other languages. Moreover, using a stacked embedding of multilingual BERT and Flair ultimately can obtain the highest score on these languages.

My contributions can help further explain the differences between contextualised word embeddings and can be used as a starting point in the event that a new data set will be released. The information gathered in this research can be used in the upcoming years to further develop contextualised word embeddings.

In this research the main question will be: what are the differences between contextualized word embedding models when predicting the (graded) effect of context in word similarity? This question will be answered by first comparing the models to each other and ultimately testing them on a specific task.

3 Methodology

In this section, I will first introduce the different contextualised word embedding models that will be compared. Later I will explain stacked embeddings and the task the models will be tested on.

3.1 Models

Transformer Most of the models used in this research are based on the Transformer Architecture [Vaswani et al., 2017]. The Transformer architecture depends on self-attention mechanisms. Self-attention is a mechanism that relates different positions of a single sequence in order to compute a representation of the sequence. Stacking self-attention provides sufficient non-linearity and representational power to learn complicated functions. This makes lots of Transformers use self-attention as beginning for their contextualized word embedding models.

In the normal Transformer architecture a model consists of a encoder and decoder. An encoder is a stack of 6 identical layers, which each have two sublayers. The first is a multi head self-attention mechanism, and the second is a fully connected feed-forward-network. A decoder is also a stack of 6 identical layers which has the same sublayers as an encoder. In addition to these sublayers there is an third sublayer which performs multi-head attention over the output of the encoder stack.

BERT BERT is a bidirectional implementation of the Transformer architecture trained through 'masked language model' pre-training objective and next sequence prediction objective [Devlin et al., 2018]. This objective, in which random tokens of the sequence are masked and the objective is to predict the original vocabulary id of the masked word, allows BERT to be bidirectional by fusing the left and right context. This is in contrary to other (earlier) models,

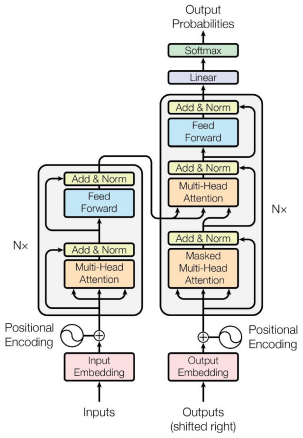

Figure 1: The Transformer - model architecture.

Figure 1: The Transformer model architecture [Vaswani et al., 2017]

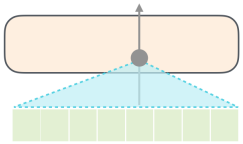

Figure 2: An illustration of normal self-attention, as used in the BERT infrastructure [Alammar, 2019]

which used autoregressive mask and only allowed attention over the previous token.

RoBERTa BERT's bidirectional architecture was also used in RoBERTa by Liu et al. [2019]. The biggest difference between BERT and RoBERTa is that RoBERTa has been trained more extensive, without the next sequence prediction objective and on another dataset. While multilingual versions exists, BERT and RoBERTa were originally released as English models.

XLM-RoBERTa XLM-RoBERTa [Conneau et al., 2019], sometimes called XLM-R, is RoBERTa's transformer architecture, combined with XLMs [Conneau and Lample, 2019] cross-lingual pretraining. In this pretraining each training sample consists of the same text in two different languages. The model still has to predict the masked tokens, but can now also use the other language as context for predicting the masked word. Another addition during pretraining is the addition of the language ID. This metadata helps the model to learn the relationship between tokens. As data set XLM-RoBERTa uses CommonCrawl instead of Wikipedia, which provides limited scale for low resource languages.

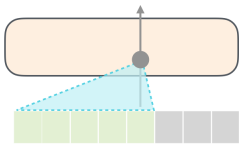

Figure 3: An illustration of masked self-attention, as used in GPT-2 [Alammar, 2019]

GPT-2 GPT2 [Radford et al., 2019] is another go on the Transformer architecture but differs in its implementation compared to BERT based models. Unlike BERT, GPT2 is auto-regressive as it can only produce estimates for the next word as output. Also, GPT2's self-attention layer differs from BERT in the way that GPT2 uses masked self-attention. BERT's self attention allows the position to see words further up in the sentence, while in masked self-attention only prior words and the current word itself.

GPT2 is only build using decoder blocks from the Transformer architecture. Instead of feeding the decoder block information made my the encoder block it is fed with zero-initialized for the first word embedding.

GPT2 is also trained on a dataset called WebText, consisting of millions of web pages

Flair The Flair model by Akbik et al. [2018] is refered to as a *contextualised string embedding*. The model is designed to handle words and context as a string of characters, instead of words. The model first works with neural character-level modeling, which is a word-level embedding. Later these embeddings get used in a sequence tagger, which retrieves the internal characters from each word and creates a contextual string embedding.

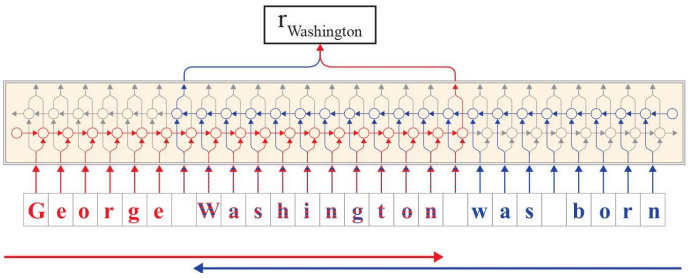

Figure 4: Extraction of a contextual string embedding for the word Washington in context [Akbik et al., 2018]

3.2 Stacked Embeddings

On top of the contextualised word embeddings, stacked embeddings will also be tested. These embeddings are generated using the earlier contextualised word embeddings by combining the vectors given by the contextualised word embeddings and concatenate them. As result a final vector is given, similarly to vectors given by the contextualised word embeddings. The idea is to combine different word embeddings to combine the characteristics of each embedding.

3.3 Task

The models are tested on their ability to identify continuous effects on meaning. During the task the models need to predict the degree and direction of change in similarity of same word pairs within two different contexts. An example can be found in Table 1.

Word 1: water Word 2 : ice	**Change: 5.56**
Context 1	**2.57**
55% to 64% **water** which comes from the milk or other ingredients. These compositions are percentage by weight. Since **ice** cream can contain as much as half air by volume, these numbers may be reduced by as much as half if cited by volume.	
Context 2	**8.13**
Water ice deposits that exist in some polar craters could serve as a source for these elements.	

Table 1: Example of the English set. Context 1 is different from context 2. When both scores get substracted that leaves a positive change of 5.56. This is the score that the models have to predict.

4 Experiment

4.1 Setup

The experiment will be run using Google Collaborations in Python 3.7. All the models will be implemented using the Flair [Akbik et al., 2019] plugin in Python.

The evaluation of the task is done using by measuring the Pearson correlation between the model predicted values and gold standards. The gold standards are the averages of the scores produced by human annotators. It seems that the BERT architecture works well for predicting the effect of context. The small difference between BERT and RoBERTa

As dataset CoSimLex [Armendariz et al., 2019] is used. This set is created using human annotaters. The data set consists of word pairs and their context.

The data set contains 340 English pairs, 112 Hungarian pairs, 111 Slovenian pairs and 24 Finnish word pairs. Among these pairs, 10 English pairs, 5 Croatian pairs en 5 Slovenian pairs were released including the annotated scores as practice data to evaluate the models.

Table 2: The models and their respective languages

Model	Languages
bert-base-miltilingual-cased	All languages
bert-large-cased	English
roberta-base	English
roberta-large	English
xlm-roberta-base	All languages
xlm-roberta-large	All languages
gpt2	English
flair	All languages

4.2 Results

The results I obtained are shown in Table3. As you can see the highest score for English is obtained using the the large version of RoBERTa, closely followed by the BERT model only trained on English and the base model of RoBERTa. Stacked embeddings are marked in italics with a + sign between the models. The highest score for each language is bold.

Table 3: Results models

Model	English	Finnish	Hungarian	Slovenian
bert-base-multilingual-cased	0.660	0.441	0.553	0.573
bert-large-cased	0.724	0.020	0.321	0.196
roberta-base	0.728	0.117	0.444	0.218
roberta-large	**0.746**	0.094	0.315	0.248
xlm-roberta-large	0.558	0.243	0.440	0.431
gpt2	0.546	0.237	0.350	0.191
flair	0.239	0.375	0.294	0.344
flair+bert-base-multilingual-cased	0.653	**0.463**	**0.557**	**0.585**
gpt2 + bert-large-cased	0.562	0.234	0.350	0.129

4.3 Results Discussion

As viewable in Table 3 the large version of RoBERTa obtained the highest score on the English data set. It seems that all the models based on the Transformer architecture work the best on this specific task in the English data set, as the best 3 scoring models inherit this structure. The base model of RoBERTa outperforms the large version of BERT with a narrow margin. Following this we could say that the difference in training makes RoBERTa the better model for this specific task. This is also supported by the fact that the large RoBERTa model obtained the highest score out of all the models. RoBERTa, based on BERT, but trained differently on a larger data set is improved compared to other BERT models. As these three models are all trained exclusively on a English data set they score low on the other languages in the task. This is expected, as well as GPT2 scoring low on these languages.

The other model based on the BERT architecture is the multilingual BERT model. This model scores worse than the both RoBERTa models and the English BERT model on the English task, but still obtains the fourth highest score. Unlike the other BERT based models, the multilingual BERT scores very high on low resource languages. Without counting stacked embeddings, multilingual BERT obtained the highest score for these low resource languages. The reason that this model scores higher than other BERT based models is due to that this model has been trained on more than a hundred languages. With a score of 0.660, multilingual BERT obtained a higher score on English than on the low resource languages, while the model was trained on all these languages. The reason for this difference could be the size of the training. Wikipedia contains more English articles than Finnish for example. This could have led to multilingual BERT being trained better on English compared to other languages.

XLM-RoBERTa does not perform outstanding on all of the languages in the task, but still obtains the second best score on the Slovenian words and the third best on the Hungarian. On the Finnish data, XLM-R performs the worst out of all the multilingual models. The reason that XLM-RoBERTa scores lower than other multilingual models may lie in the data that has been used in pre-training. The CommonCrawl data set is designed to be more diverse than other data sets, which mainly use Wikipedia and books. CoSimLex has been build using Wikipedia, which can lead to higher scores by models that have also been trained on Wikipedia. XLM-RoBERTa scoring very low on Finnish comes as a surprise. One of the upsides of the CommonCrawl data set should be that there is more data available of low resource languages. Finnish scoring very low comes a bit as a surprise, as the language is not underrepresented in the CommonCrawl data set. There might be a bigger difference between the Finnish Wikipedia and CommonCrawl data, but this will need further research.

As for the stacked embeddings, the stacked embedding with GPT2 and BERT does perform around the same score as GPT2 and worse than BERT. As both models are trained on a English data set one would think they would perform better together, or at least better than in this case. As explained earlier stacked embeddings combine the characteristics of two embeddings. One

reason for this lower score can possibly be explained that the stacked embedding does not inherit the right characteristics of two models in this case, and thus not generating a high score. The other stacked embedding, with Flair and multilingual BERT, performs the best on all languages except English. It seems that combining the characteristics does work in this specific case. The scores are not exceptionally higher than multilingual BERT, but they exceed all the scores on low resource languages. As Flair and multilingual BERT both score high on these low resource languages, their combined assets make the best model yet. BERT's bidirectional Transformer architecture might be improved by combining it with Flair's contextual string embedding characteristics. The only downside is that this stacked embedding performs worse on English then multilingual BERT. This could be seen as a trade-off for improving the overall score of low resource languages.

5 Discussion

During this research I mainly focused on comparing different contextualised word embedding models with each other and tried to explain differences and similarities between models. The research does not dive deeper into subject than earlier research. While I know the comparisons are limited and there are much more models available, I think this research is a starting point for more research about contextualised word embedding models and their effect on predicting graded change on context.

The methods used are inherited from earlier research. There is little to doubt about the scores that were provided by the baseline model, as these were generated by the original creators of the baseline model. Because of this most of the results can be generalized.

The research has been limited in size due to lack of time. There could have been more research, for example by trying more different models and more variations of stacked embedding combination, as this research only supports two different combinations. This is possible in further research.

In this research only the effect of context has been investigated, while the definitive scores are also supported in the CoSimLex data set. In further research models could be compared to each other on a slightly different task, in which this score has to be predicted.

In the future, there will probably en even bigger rise in contextualised word embeddings available. This research can be used as a foundation for comparing these models to each other, and earlier models.

6 Conclusion

The goal of this research was to find and explain the differences of contextualised word embedding models. In this research we found that the BERT architecture offers a great framework for the task of predicting the effect of context in word

similarity. Along these findings, we can conclude that there should be more research to stacked embeddings, as we found examples of stacked embeddings improving the scores of multilingual models.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649, 2018.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

Jay Alammar. The illustrated gpt-2 (visualizing transformer language models), Apr 2019. URL `https://jalammar.github.io/illustrated-gpt2/`.

Carlos Santos Armendariz, Matthew Purver, Matej Ulcar, Senja Pollak, Nikola Ljubesic, Marko Robnik-Sikonja, Mark Granroth-Wilding, and Kristiina Vaik. Cosimlex: A resource for evaluating graded word similarity in context. *CoRR*, abs/1912.05320, 2019. URL `http://arxiv.org/abs/1912.05320`.

Carlos Santos Armendariz, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Ivan Vulić, and Mohammad Taher Pilehvar. SemEval-2020 task 3: Graded word similarity in context. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 36–49, Barcelona (online), December 2020. International Committee for Computational Linguistics. URL `https://aclanthology.org/2020.semeval-1.3`.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069, 2019.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

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

Hansi Hettiarachchi and Tharindu Ranasinghe. BRUMS at SemEval-2020 task 3: Contextualised embeddings for predicting the (graded) effect of context in word similarity. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 142–149, Barcelona (online), December 2020. International Committee for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.semeval-1.16`.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3509–3514, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1355. URL `https://www.aclweb.org/anthology/D19-1355`.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL `http://arxiv.org/abs/1706.03762`.