Prompt based contextualized phrase embedding

B Tsakam,
brice_tsakam(AT)hotmail(dot)com
May 31st 2025

Abstract

Word2vec [1] gave us word embeddings, elmo and bert [2,3] made them context dependent. Word embedding evolved into whole sentence embedding models such as use [4]. At the heart of this concept is the notion that a portion of text can be represented as a distinct vector within a high-dimensional space, accompanied by a similarity metric that indicates semantic relatedness. We introduce a straightforward prompt-based contextualized phrase embedding (PCPE) method to transform a sentence within a broader context into an embedding that preserves meaningful similarity measures. The assessment utilizing the Phrase In Context similarity dataset [11] demonstrates the advantages of this contextualized phrase embedding approach.

Introduction

Text embedding is a basic component in numerous natural language processing applications including indexing, search, clustering, question answering. In the realm of contextualized word embedding, the development of contextualized text segment embedding would facilitate the application of embedding-based NLP techniques beyond individual words while taking into account the surrounding text for more sensible semantics. A given sentence, given its context would be interpreted literally or figuratively or sarcastically and embedded accordingly. To realize this contextualized phrase embedding we build on prompt based techniques to get embeddings out of generative models on one hand and on the few-shot learning concept on the other hand.

After a review of the existing phrase embedding models and techniques, we introduce a straightforward prompt technique that facilitates the contextualized embedding of a phrase, applicable at the word, predicate, sentence levels and beyond. The prompt based contextualized phrase embedding (PCPE) is assessed through a contextualized phrase similarity task. The results indicate that our PCPE method not only improves the accuracy of phrase embeddings but also enhances the understanding of semantic nuances in various contexts.

Background

Small language models such as bert [4] and later Large language models such as ChatGPT3 [5] achieved impressive progress in machine reading comprehension and text generation and demonstrated the capability and performance benefit of larger models. However their generic nature makes fine tuning necessary for optimal performance on specific tasks and/or specialized domain knowledge. The fine tuning steps makes the model less accessible to the majority of users and hence less applicable. Few-shot learning proved effective in hinting the model on how best to perform the task by the means of a few in prompt examples [6]. We use the few-shot approach to provide the context of the phrase to be embedded. By phrase we mean a group of words that amount to a part of or the whole sentence. Then we use the method presented in [7]: Prompt with Explicit One word Limitation (PromptEOL) to instruct the model in predicting the meaning of a phrase in one word, and then to use that embedding. By building on prior art, the prompt based contextualized phrase embedding (PCPE) approach produces embeddings that capture a more nuanced understanding of the phrase in relation to its surrounding text.

Method

A requirement for our work is to possess a segmentation tool that can break down a text in smaller pieces such as: paragraphs, sentences, predicates, grammatical units, and entities. In addition to traditional methods like markup and phrase segmentation tools [8], current Large Language models may be useful for this purpose [13].

In Prompt based Contextualized Phrase Embedding (PCPE) we use a single prompt template that includes an instruction and a query. We compare prompt variations defined as follows:

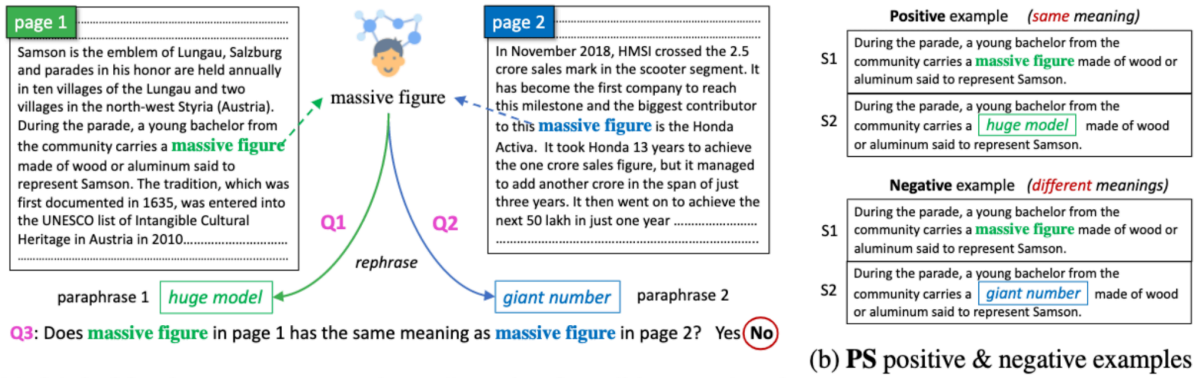
Figure 1: prompt definition

```
# Prompt definition

prompt = 'Instruct: {task_description}\nQuery: {query}'



# case 1: Define task and query: phrase only

task_description = 'Answer the query'

query = '\nDefine "{}"'.format(phrase)



# case 2: Define task and query: phrase + context

task_description = 'Answer the query given the short article:
{}'.format(context)
```

```
query = '\nDefine "{}"'.format(phrase)



# case 3: Define task and query: phrase + context + instruction for one word representation (PromptEOL [7])

task_description = 'Answer the query given the short article: {}'.format(context)

query = '\nDefine "{}" in one word: "'.format(phrase)
```

To evaluate the effectiveness, we utilize the provided prompts in turn to identify the influence of including context in the prompt and PromptEOL [7]. We use the Phrase in Context similarity training dataset [11,12]. This dataset features pairs of short texts that include the same phrase along with one positive or negative label for the pair. A positive label means the phrase has the same meaning in both texts otherwise the label is negative.

Figure 2: Phrase in Context Dataset [12]



(a) Q1 & Q2 ask annotators to rephrase "massive figure" in page 1 and page 2. Q3 asks whether this phrase's meaning is the same in both pages.

(b) **PS** positive & negative examples constructed using page 1 context (similarly, we repeat for page 2).

Evaluation

We leverage the SFR2 generative model [9] to obtain sentence embeddings following the methodology in [7]: the LLM is prompted to generate the next token, then the hidden vector of the final token is extracted as sentence embedding. Then a simple linear discriminative model finds the optimal similarity threshold to predict the positive or negative label. The performance on the Phrase in Context Similarity Dataset is presented (see Table 1).

SFR2 provides a competitive baseline compared to the anterior models implemented in [12] without fine tuning, while embedding the phrase only. We see performance improvement when the context is included. The best results are obtained when using also the PromptEOL technique.

Table 1: cosine similarity results

Model	Phrase	Phrase + context	PCPE	Delta
SFR2	51.3*	63.3*	64.8*	13.50
PhraseBERT	51.75	63.40		11.65
BERT	51.05	64.10		13.05
SimCSE	52.15	62.50		10.35
SpanBERTLarge	50.40	66.30		15.90

Phrase only is the base line that does not include any context. Phrase+context includes the context and Phrase. PCPE includes the phrase, the context and the instruction to represent the phrase in one single word. (*: SFR2 experiments including PCPE are evaluated on only 50% of the dataset, 3500 sample because of limited compute resources)

Prompt based contextualized phrase embedding achieves competitive performance with previously evaluated models on this dataset both for the phrase only and the phrase + context cases. In addition, PCPE features the instruction to embed a specific phrase from the context and effectively produces different prompts for the different sentences of the given context. Therefore, in contrast to the alternative techniques, PCPE has the unique ability to guide the model to generate an embedding for the target phrase as opposed to the complete context.

Discussion

The accuracy of the Phrase in Context similarity task is improved from 51.3% to 64.8% using the SFR2 model. Using the Prompt based Contextualized Phrase Embedding (PCPE) technique. Given a single context PCPE provides different embeddings for each of the phrases. An associated tool may extract noun phrases, action phrases, prepositional phrases, entities, etc while PCPE enables getting a contextualized embedding. This technique improves on existing contextualized embedding and offers better control on the granularity of the text segment being embedded while retaining semantics.

The non isomorphic nature of embeddings [10] implies that the linear method based thresholding will have limited success, as one cannot anticipate a universal threshold. Consequently, the accuracy could see significant enhancement by learning how a particular embedding position affects the similarity threshold. This emphasizes the necessity to validate this method against a wider array of datasets.

We find the presented prompting technique effectively improves the embeddings without resorting to fine-tuning. Further research would encompass validating this technique against diverse datasets, in conjunction with other generative models. And also with more nlp tasks such as indexing, searching, clustering and extractive question answering.

References

[1] Mikolov, Chen, Corrado, Dean (2013). "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781

[2] https://en.wikipedia.org/wiki/ELMo

[3] https://en.wikipedia.org/wiki/BERT_(language_model)

[4] Cer, Yang, Kong et al. (2018). "Universal Sentence Encoder". arXiv:1803.11175

[5] https://en.wikipedia.org/wiki/GPT-3

[6] Brown, Mann, Ryder et al. (2020). "Language Models are Few-Shot Learners". arXiv:2005.14165v4

[7] Jiang, Huang, Luan et al. (2023). “Scaling Sentence Embeddings with Large Language Models”. arXiv:2307.16645

[8] Sadvilkar and Neumann. (2020). PySBD: Pragmatic Sentence Boundary Disambiguation. *Proceedings of Second Workshop for NLP-OSS, ACL*

[9] Meng, Liu, Rayhan Joty et al. (2024) “SFR-Embedding-2: Advanced Text Embedding with Multi-stage Training”, https://huggingface.co/Salesforce/SFR-Embedding-2_R

[10] Ethayarajh (2019). “How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings”, arXiv:1909.00512

[11] https://phrase-in-context.github.io/

[12] Pham, Yoon, Bui, Nguyen (2023).“PiC: A Phrase-in-Context Dataset for Phrase Understanding and Semantic Search”, https://arxiv.org/pdf/2207.09068

[13] Zaratiana, Tomeh, Holat, Charnois (2023). “GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer“, https://arxiv.org/abs/2311.08526