LLM2Vec: Large Language Models Are Secretly Powerful Text Encoder

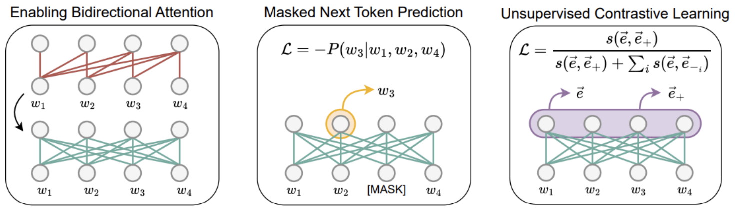

Enabling Bidirectional Attention
w_1 w_2 w_3 w_4
w_1 w_2 w_3 w_4
Masked Next Token Prediction
$\mathcal{L} = -P(w_3|w_1, w_2, w_4)$
w_3
w_1 w_2 [MASK] w_4
Unsupervised Contrastive Learning
$\mathcal{L} = \frac{s(\vec{e}, \vec{e}_+)}{s(\vec{e}, \vec{e}_+) + \sum_i s(\vec{e}, \vec{e}_{-_i})}$
$\vec{e}$ $\vec{e}_+$
w_1 w_2 w_3 w_4

Figure 1: The 3 steps of LLM2Vec. First, we enable bidirectional attention to overcome the restrictions of causal attention **(Bi)**. Second, we adapt the model to use bidirectional attention by masked next token prediction training **(MNTP)**. Third, we apply unsupervised contrastive learning with mean pooling to learn better sequence representations **(SimCSE)**.

This paper caught my attention because it discusses a topic that has been intriguing me lately (along with new cool methods for the generative diffusion models and Mamba/SSM). The focus of this paper is on adapting pretrained language models to perform discriminative tasks, such as topic modeling or sentiment analysis, part-of-speech tagging, etc. Most language models today are trained to generate text, i.e., perform generative tasks (based on a decoder-only architecture).

You might wonder why we need models for discriminative tasks if most can be converted into generative tasks. For instance, sentiment analysis can be turned into generating the sentiment of a given text (e.g., "The sentiment of this text is positive"). However, the question is whether this “conversion” is optimal in terms of size, performance, and training effort required for this task, especially in scenarios with strict memory or latency requirements.

Can we do better? As mentioned, most powerful LLMs released in the past 3 years are generative models with decoder architectures (GPT, Gemini, Claude, etc.). Models trained for discriminative tasks with encoder architectures have become quite rare.

Hence, the reviewed paper aims to adapt (calibrate) a generative language model (decoder) for discriminative tasks.

Now, why not simply fine-tune a pretrained decoder-based LLM for a discriminative task? To understand why this might not be optimal, we need to acquire deep understanding on how encoder and decoder models are trained. During an encoder-based model training, we mask certain tokens and train the model to predict them, using all tokens in the text to predict the masked ones. If the dataset is large and diverse enough, the model learns to "understand" the language statistically. In contrast, the decoder model is generative, designated to create new data and trained to predict the next token. During decoder-based model training we hide future tokens from the model during mask token prediction.

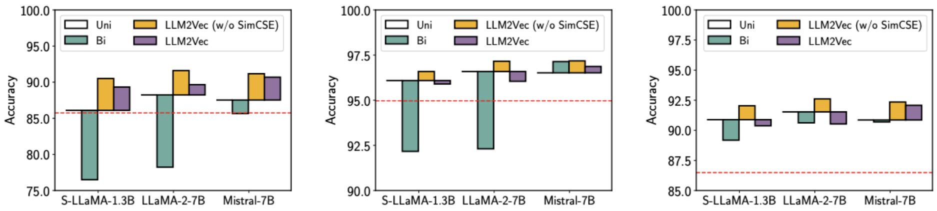
Due to the different training schemes, it is naive to expect decoder-trained models to excel in discriminative tasks after fine-tuning (though it is not impossible and may work well for some tasks depending on labeled data size and quality). For example, for part-of-speech tagging, the representation of a word in a pretrained decoder model considers only the previous words, which is not optimal for this task.

After this long intro, let's focus on the reviewed paper. It proposes a way to adapt a trained decoder model for discriminative tasks in 3 steps:

1. Disable the masking of future tokens in the attention mechanism, allowing the model to use all tokens to build representations of each token. However, the paper notes that model performance drops afterward (hence, there are two more steps in the process).
2. During training, instead of predicting the masked token from its contextual representation, predict it from the previous token's representation. The rationale behind this is not 100% clear.

3. Use contrastive learning. Contrastive learning methods train data representations (usually unlabeled) by bringing similar examples' representations closer and pushing apart dissimilar/unrelated examples'' representations (e.g. in terms of cosine similarity). The paper suggests training the model to bring representations of the same sentence with different dropouts closer. Recall that dropout essentially zeros out connections/weights between different neurons in the model. At the same time Representations of different sentences are trained to be far apart in the embedding space.

The paper claims that combining these steps transforms the model into an encoder capable of producing strong data representations, showing decent performance in several discriminative tasks.



(a) Chunking (b) NER (c) POS

Figure 2: Evaluation of LLM2Vec-transformed models on word-level tasks. Solid and dashed horizontal lines show the performance of `Uni` and `DeBERTa-v3-large`, respectively.