Proposal: Explaining embedding vector results for scoring alignments

Riley Gavigan[1], Lucian Ilie[1*]

[1]Department of Computer Science, University of Western Ontario, London, N6A 5B7, Ontario, Canada

October 1, 2023

Abstract

The proposed E-score alignment scoring method [1] generates results that further research will explore, leading to performance benefits. Cosine similarity results generated by the models were mostly positive instead of being randomly distributed. Exploration of this distribution will lead to a better understanding of the different models and the embedding vectors being produced. Different embedding models also differ greatly in performance with varying mean ranges. From this research, insight will be obtained to improve embedding performance and transformer model architecture.

1 Introduction

2 Relevant Background

2.1 Protein transformers

2.2 Embedding vectors

2.3 *E*-score calculations

3 Methods

3.1 Visualizing *E*-score dispersion

To gain a deeper understanding of the *E*-score results generated for the different embeddings as shown in Figure 1, dispersion will also be visualized. By visualizing the standard deviation and range/interquartile range of the results for each embedding type, there will be more insight to draw conclusions from. This will help understand why cosine similarity mean results are mostly positive (0..1), and why different models generate broader or narrower average ranges.

The *E*-score codebase will serve as the foundation for performing further analysis and visualization of dispersion.

3.2 Comparing embedding models

Average and dispersion *E*-score results will be compared between the available embedding models to understand the factors contributing to the better performance of particular models over other models. This data will be used to draw conclusions about embedding performance differences.

3.3 Modifying parameters

By modifying different parameters and generating resulting *E*-scores for different combinations from insight gained from differences between models, a greater understanding of the primary factors contributing to performance will provide insight to improve embeddings. Examples of parameters that can be modified are alignment type, gap penalty, and gap extension penalty.

4 Objectives

4.1 Embedding vector distributions

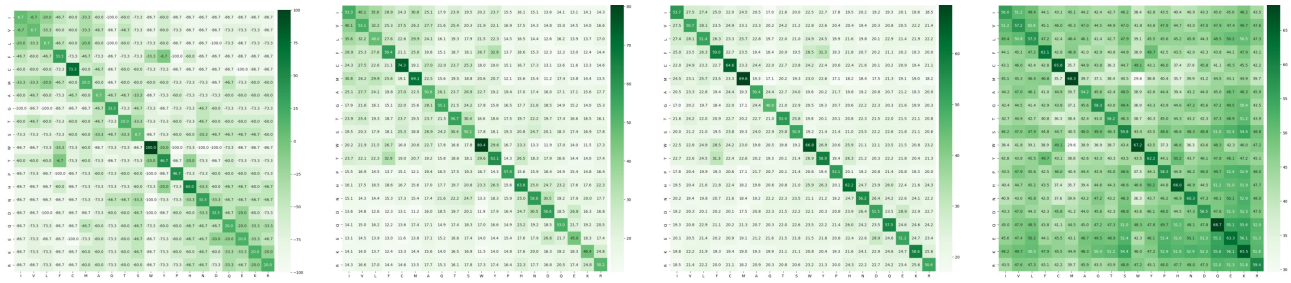
By understanding the distributions of the embedding vector values, there will be a further understanding as to why the cosine similarity results are mostly positive (0..1) for all of the alignments when calculating *E*-score, instead of being randomly distributed from (-1..1).

Because cosine similarity averages are mostly positive, there are likely factors within the embedding vectors that are contributing to the angles between the embeddings generally not being opposite in direction to the same extent that they are similar in direction.

4.2 Cosine similarity results

After having a stronger understanding of the embedding vector distributions, the cosine similarity results will be explored by visualizing the *E*-score dispersion. Testing different combinations of alignments and modifying parameters will aid in exploring cosine similarity.

*Thesis supervisor. Bioinformatics Lab - csd.uwo.ca/ ilie/lab.html



(a) (b) (c) (d)

Figure 1: *Heatmaps of (a) BLOSUM62 matrix (scaled to -1..1) and three aligned matrices of average E-scores for the NBD_sugar-kinase_HSP70_actin MSA: (b) ProtT5-score, (c) ProtAlbert-score, and (d) ProtXLNet-score.*

By exploring and understanding cosine similarity results, conclusions will be drawn pertaining to the different factors contributing to the average results that were initially observed.

4.3 Embedding performance differences

The results between different embedding models vary greatly in their average range of values and in comparison to BLOSUM [2] matrix results. Understanding why different embedding models generate different ranges in *E*-score values will serve as a basis for drawing implications from the research. With knowledge about the factors that contribute to embedding performance, improvements can be made to the different embedding models for scoring alignment.

5 Research Implications

There are three primary advancements that can be explored from the implications drawn from the embedding vector and cosine similarity results. These would serve as beneficial follow-up research to improve embedding performance, transformer models, and to apply the same advancements to Natural Language Processing (NLP).

5.1 Improving embedding performance

From the insight gained about the properties that contribute to the performance of different protein embeddings, the performance of the embeddings can be improved for sequence alignment. This would be an application of the data-supported explanation for why different embeddings have different cosine similarity averages and ranges.

5.2 Transformer model improvement

The above insights can be used as a starting point to modify the architecture of the different transformer [3] models such as the ProtTrans models ProtT5, ProtBert, ProtXLNet, and ProtAlbert [4]. This research can be extended to improve these models through different processes such as hyperparameter tuning and optimization.

5.3 Natural Language Processing

All of the above work can be repeated for Natural Language Processing contextual embeddings such as ELMo [5], BERT [6], RoBERTa [7], XLNet [8], and T5 [9].

References

[1] S. Ashrafzadeh, G. Brian Golding, and L. Ilie. “Scoring alignments by embedding vector similarity”. In: (2023).

[2] S. Henikoff and J. G. Henikoff. “Amino acid substitution matrices from protein blocks”. In: *Proceedings of the National Academy of Sciences* (1992). DOI: 10.1073/pnas.89.22.10915.

[3] A. Vaswani et al. “Attention is all you need”. In: (2017). DOI: arXiv.1706.03762.

[4] A. Elnaggar et al. “Prottrans: Toward understanding the language of life through self-supervised learning”. In: *IEEE transactions on pattern analysis and machine intelligence* (2021). DOI: 10.1109/TPAMI.2021.3095381.

[5] M. E. Peters et al. “Deep contextualized word representations”. In: (2018). DOI: arXiv:1802.05365.

[6] J. Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: (2018). DOI: arXiv:1810.04805.

[7] Y. Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: (2019). DOI: arXiv:1907.11692.

[8] Z. Yang et al. “XLNet: Generalized autoregressive pretraining for language understanding”. In: *Advances in neural information processing systems* (2019). DOI: 10.48550/arXiv.1906.08237.

[9] C. Raffel et al. “Exploring the limits of transfer learning with a unified text-to-text transformer”. In: (2020). DOI: arXiv:1910.10683.