Testing Symmetry in BERT's Embeddings

Thomas W. Bush, Michael A. Awanah

January 2024

1 Abstract

This project investigates the symmetrical features of BERT embeddings when processing sentences with opposite meanings. We use a dataset of pairs of statements with different semantic orientations and look at the embeddings to see if there are any patterns that could show whether BERT understands semantic opposition in a symmetric or asymmetric way. The objective of this project is to enhance our overall comprehension of how advanced language models analyze and depict subtle linguistic distinctions, specifically within the framework of contrasting meanings. The consequences of our findings have the potential to impact many natural language processing (NLP) applications, such as formal reasoning and spatial derivation.

2 Introduction

For the purpose at hand, we can consider positional statements as sentences that follow this structure: The spatial relationship between the two entities, A and B, is described as "A is at the left of B." These assertions can explicitly express information about the state of A with regard to B. A vector, V (S1), can represent this information. Additionally, there is an implicit type of information about the state of B in relation to A. This information represents the inverse relationship between the two things, and it can be expressed as the phrase 'B is to the right of "A".

Humans never need to explicitly state the opposite information because their acknowledgement of their position always implies it. Starting from this observation, we have decided to investigate whether Large Language Models (LLMs) represent spatial relationships by taking into account this property or if they are unable to represent it.

Recent evidence has shown the current limitation of LLMs [1]. In particular LLMs seem to achieve *formal linguistic* abilities, but not *functinal linguistic* abilities. This is because the functional linguistic is based on other properties rather than only language. We focused on the limitation that LLMs seem to have regarding formal reasoning. In particular, it seems that they struggle when they have to apply logical rules to many cases. Some have tried to propose a

solution to this issue by no presenting all the cases at once to the model, but using a technique called *chain of thought*, where the cases are presented to the model one by one [3]. However, this method does not address the problem directly and it can be consider only a partial solution.

Spatial relationships are a special case of formal reasoning. When two entities are related in space, their relationship is bidirectional. Sentences describing their spatial relationship, however, do not represent this idea because people can implicitly infer the symmetry between the two statements to represent the bidirectionality. We think that this property is not represented in the human language network but in other brain areas related to spatial navigation and spatial representation. Thus, LLMs are an interesting subject to study for testing this property, as all their knowledge must come from language and how they represent it.

From this observation, we have decided to test how a transformer encoder BERT, which has shown to be able to excel in many linguistic domains [2], represents sentences that contains spatial relationships and whether its representations account for the property of symmetry.

3 Methods

3.1 Data Preparation

To test BERT's embedding regarding opposite positional statements we have created a custom dataset, visible in the table 1, formed by: a full sentence about a spatial relationship, the first entity of the sentence, the second entity of the sentence, and a label about their spatial relationship. For convenience we have decided to utilise just *left/right* spatial relationships. Additionally, since the scope of our study was to test a property that should be intrinsic of BERT, the size of the dataset was not a significant factor, for this reason we have used a small sample size (n=100). Over the course of our project we have handled the dataset with the pandas library.

Sentence	**Entity_1**	**Entity_2**	**Spatial_rel**
The clock is left of the painting.	The clock	The painting	left of
A tree is left of the building.	A tree	The building	left of
The lamp is left of the television.	The lamp	The television	left of
A cat is sitting left of the dog.	A cat	The dog	left of
The coffee cup is left of the computer.	The coffee cup	the computer	left of

Table 1: Example of dataset

3.2 Probing BERT

To correctly identify the most informative layer for the extraction of the embeddings about the spatial relationship, we have decided to probe BERT using

a technique called *cumulative score*[2]. For doing so we have: first encoded the spatial relationship into a binary label. Then we have trained a Logistic Regression Classifier from the *sklearn* library on each layer of BERT to classify each embedding into the correct spatial relationship. At each iteration the improvement in accuracy is calculated in relation to the added layer, in order to determine which layer is the most informative for determine the spatial relationship.

3.3 Embeddings Creation

We decided to get the embeddings from text data using a BERT model that has already been trained to understand the complex semantic relationships between words in sentences. In particular, it goes through a set of sentences, tokenizing and feeding each one, along with individual entities taken from those sentences, into the BERT model to get their embeddings. For each sentence, the process involves generating an embedding for the full phrase and separate embeddings for two entities identified within the sentence ($Entity_1 and Entity_2$). These embeddings take the mean of the last hidden state output from the BERT model and turn it into a single vector representation for each sentence and entity. This effectively reduces the contextual information the model has gathered. Subsequently, the embeddings for the two entities are concatenated to form a combined feature vector, while the sentence's embedding serves as a target vector. This procedure is repeated for each sentence in the dataset, accumulating the feature vectors (representing entity pairs) and target vectors (representing sentences) into two separate lists.

3.4 Matrix Extraction

Since we were interested in the spatial relationships between the entities we have though about these relationship as matrices that take the two entities vectors and multiply them to create the final vector that represent their spatial relationship. To obtain the matrix of the spatial relationship **left of** we have decided to train a Neural Network on the two entities embeddings produce by BERT to produce the final embedding of their spatial relationship. We have though that the weight matrix of the model can represent the spatial relationship **left of**. The model 1 has been built using the *PyTorch* library, and it is formed by a linear layer that reduce the dimension of the concatenation of the entities and a final layer that produce the output.

The model has used *Adam* as optimiser and the *cosine similarity* as loss function. The model has been trained on 50 epoches with a batch size of 16. The data has been given to the model using the *Dataset* and *DataLoader* function from the *PyTorch* library.

After the training phase we have tested whether our model had correctly represented the spatial relationship *left of* by its gradients. On the test set the model has produced an average cosine similarity value of: 0.91 which implies

```
class TransformationNet(nn.Module):
    def __init__(self, input_size, intermediate_size, output_size):
        super(TransformationNet, self).__init__()
        self.reducer = nn.Linear(input_size, intermediate_size)
        self.transformer = nn.Linear(intermediate_size, output_size)

    def forward(self, x):
        x = self.reducer(x)
        x = self.transformer(x)
        return x
```

Figure 1: Implementation of the Transformation Network in PyTorch

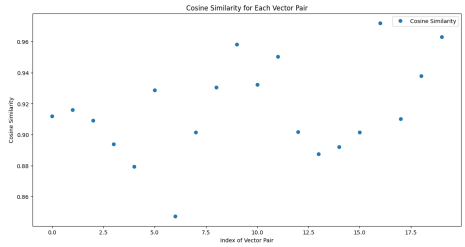

Figure 2: Cosine Similarity

that the output produce by our model is similar to the final embeddings from BERT. We have then plotted these results by using the *Mat Plot Lib* 2.

3.5 Matrix inversion

To test our hypothesis about symmetry, we have inverted the weight matrix representing *left of* to obtain the symmetrical relationship *right of*. To do so, we have used the *numpy* library for linear algebra operation 3.

```
transformation_matrix = model.transformer.weight.detach().numpy()
inverse_matrix = np.linalg.inv(transformation_matrix)
```

Figure 3: Implementation of the Transformation Network in PyTorch

3.6 Testing the Inverted Matrix

Finally we have replicated the methods above to test whether a new model using the inverted matrix, without prior training, would be able to produce embeddings similar to the ones produced by BERT about the spatial relationship *right*

of. We have used a custom dataset with sentences expressing the right spatial relationship, with the same structure of the one used above. We have obtained a list of embeddings for each sentence from BERT and a list of embedding for each couple of entities from our model with the inverted weight matrix, then we have measured the cosine similarity between the two list of embeddings, obtaining a result of 0.005. From this result it appears that the embeddings produced by BERT do not represent the property of symmetry that we naturally perceive from opposite positional statements.

4 Conclusion:

Our project tries to investigate how LLMs represent sentences about physical relationships. We were interested in understanding whether these models were able to capture properties that humans likely process in brain areas outside the language network.

We implement our code on Google Colab and make our code available on GitHub.[1]

4.1 Contributions:

Distribution of the work:

- **Thomas**: probing, embeddings, model building, matrix extraction, testing new model, report (concluson, formatting in latek)
- **Michael**: writing of the report, data preparation (creation of dataset, encoding of labels), probing, model testing

References

[1] K. Mahowald, A. A. Ivanova, I. A. Blank, N. Kanwisher, J. B. Tenenbaum, and E. Fedorenko. Dissociating language and thought in large language models. 1 2023.

[2] I. Tenney, D. Das, and E. Pavlick. Bert rediscovers the classical nlp pipeline. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 4593–4601, 5 2019.

[3] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. C. Quoc, V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 12 2022.

[1] `https://github.com/Thomasbush9/python_project_clc`