# A hierachical, explicable model for similarity of long texts

Bennett Bullock

October 3, 2022

## 1 Introduction

This exercise will focus on a similarity metric for transcripts that can be used to identify peer companies for Causeway. Peer companies are companies that are in the same sector and, as much as possible, are concerned about the same issues. The higher this metric is, the more likely these two companies are to be peer companies. I will be using a variation of the techniques I described in my interviews, but the content presented here will be original. By doing this, I hope to demonstrate that I can solve difficult NLP problems quickly and creatively.

Because transformers such as BERT or RoBERTa have limits on the input length, there has been significant effort to find embeddings for longer texts, the most prominent of which is LongT5. In my experience, LongT5 is effective in comparing long texts, although it does suffer from vagueness in certain text comparisons. The major issue with comparing two long texts is that many words/phrases from one text compare to many words/phrases from the other, adding an noisy, unreliable, and often unexplainable comparison. Did two documents match because they have similar numbers of vacuous phrases, or did they match because they say the same things about the same topics? In this exercise, I will introduce an alternative way to use transformers to compare long texts.

I will attempt to meet the following criteria for the metric:

1. **Reliability**. Companies in the same sector should match (show maximum similarity). If the topics discussed in the transcripts are similar, they should match as well.

2. **Explainability**. The basis for comparison should be explainable (we should be able to see which parts of the text dominate the match).

3. **Noise-Robustness**. The metric should be resilient to noise, such as vacuous language ("my next question", "thank you", "this is Rob from JP Morgan").

# 2 Improving on BERT embeddings

To understand our approach, it is important to compare it to the conventional approach for extracting BERT embeddings. From Viswani et al., an embedding for a sentence can be derived from the encoding stack. For a string with $k$ tokens and a transformer with $h$ attention heads in each block in the stack, the output of each block can be described - eliminating the activator and normalization functions for simplicity - as:

$$M_n = M_{n-1} + \mathbf{F}(M_{n-1} + \mathbf{concat}(\mathbf{Att}_1(M_{n-1}), \mathbf{Att}_2(M_{n-1})...)W_O) \quad (1)$$

$M$ is $k \times d$, or a matrix whose rows are embeddings for the token. **Att** creates a version of $M$ whose rows are contextually modified by the other tokens, and different attention heads account for semantic, syntactic, and contextual dependencies, as discussed here. $W_O$ is $hd \times k$, which reduces the concatenation of attention heads to a $k \times d$ matrix that can be summed with $M_{n-1}$. $\mathbf{F}$ is a feedforward layer.

It is helpful to think of the final output of the encoding stack as a matrix whose rows are embeddings of the tokens which now contain relevant information supplied by other tokens - is the token a subject or object, is it a verb or a noun, is it modified by an adjective, and if it has multiple meanings, which meaning it may be.

In a typical BERT embedding, the output of the final N layers of the encoding stack are summed as $\hat{M} = \sum_i M_i$, and then the rows are summed as $\vec{m} = \sum_j \hat{M}_{j,*}$. In my experience, this presents a challenge with longer

texts. The cosine similarity of two vectors $\vec{m}_1$ and $\vec{m}_2$ depends on a dot product:

$$\vec{m}_1^T \vec{m}_2 = \sum_{i,j} \hat{M}_{i,*} \hat{M}_{j,*}^T \tag{2}$$

We can see that if the number of terms in the sum grows, it becomes more likely that meaningless comparisons will be added to the similarity. In our case, meaningless comparisons are those that have cosine similarities around 0.5-0.7.

However, I have observed that a row of $\hat{M}$ is contextualized by other tokens nearby, so that in "the projections of revenues are" and "revenue projection may turn out to be", the embeddings for "are" and "turn" may be much more similar, because they are contextualized by previous words. Therefore, selecting the tokens with the maximum similarity would be less noise-prone than the conventional metric. Generalizing this for $\hat{M}_1$ and $\hat{M}_2$ gives an asymmetric similarity metric. In this metric, rows of $\hat{M}_1$ are L2-normalized, such that the dot product of each vector is a cosine similarity. For the $i$-th row of $\hat{M}_1$, the similarity is $\mathbf{max}_j \hat{M}_{1,i,*}^T \hat{M}_{1,j,*}$. We can then define the similarity between the two texts as the sum of these across rows. To normalize for text length, we apply a denominator analogous to the denominator in cosine similarity using the number of rows and columns:

$$s_{1,2} = \frac{\sum_i \mathbf{max}_j \hat{M}_{1,i,*} \hat{M}_{1,j,*}^T}{\sqrt{\mathbf{numRows}(\hat{M}_1)\mathbf{numColumns}(\hat{M}_2)}} \tag{3}$$

This metric satisfies the criteria of noise-robustness by matching the most similar tokens and eliminating noise generated by other matches. However, because token embeddings are modified by other token embeddings, a match for one token matches other parts of the text as well. A further modification made to the algorithm was to only include distances above a threshold, 0.9, which augmented noise-robustness by eliminating vacuous maximum similarities.

# 3 Description of the algorithm

The algorithm consists of (a) extracting parts of the transcript ("lines"), using only lines that were 500 characters long, and filtering their tokens using

a stoplist, (b) extracting the output of the encoding stack of a BERT model, (c) computing the similarities of each line, and (c) based on the similarities of each line, computing the similarities of each transcript.

Filtering stopwords from the original line may seem a bit controversial, as BERT models are designed to process natural language text with stopwords. However, when I computed similarities without stopwords, I found that vacuous parts of a line ("I think that", "I was wondering", "Could you elaborate on") dominated the similarity, and more topically relevant parts of the line were ignored. In terms of how a transformer works, this could be explained by partial activation of attention heads. Attention heads that account for stopwords are no longer activated, while attention heads that account for semantically relevant words contribute to contextualizing tokens. This is purely speculative, however.

We compute $\hat{M}$ for a line. For each line in $t_1$ and $t_2$, we compute the similarities, producing a set $S = \{s_{t_1,l_1,t_2,l_1}, s_{t_1,l_1,t_2,l_2}, ...\}$. $S_K$ is the maximum $K$ elements of $S$. The sum of these $K$ similarities is the similarity between $t_1$ and $t_2$.

# 4 Results and Conclusion

Results for 15 the most similar transcripts are in Table 1. Similarities in bold indicate that the two transcripts are for companies in the same sector. Table 2 indicates the shared topics of the first or second line for company pairs in different sectors. "capital markets" refers to discussions about portfolio management, "capital structure" refers to discussions about internal allocation of capital, "labor" refers to dealing with increasing cost of labor and maintaining employees, and "customer base" deals with managing customers.

All airlines match with other airlines, with more spotty results for the automotive sector. There are too few companies in other sectors to make any meaningful comparison. General Motors seems to be similar to companies in other sectors, probably due to its high level of financialization - "capital markets" are prominent in across-sector comparisons.

Table 1: Maximum similarities for transcripts

| **Company** | **Compared Company** | **Similarity** |
|---|---|---|
| United Airlines Holdings | American Airlines Group | **0.366** |
| JetBlue Airways | United Airlines Holdings | **0.359** |
| Southwest Airlines | American Airlines Group | **0.325** |
| Citigroup | General Motors | 0.303 |
| American Airlines | United Airlines | **0.295** |
| Cross Country Healthcare | Citigroup | 0.263 |
| General Motors | Citigroup | 0.263 |
| Exxon Mobil | Citigroup | 0.231 |
| Ford Motor | General Motors | **0.210** |
| Sysco | Delta Airlines | 0.187 |
| Delta Airlines | United Airlines | **0.170** |
| Tesla | General Motors | **0.147** |
| Rivian Automotive | General Motors | **0.130** |
| Stellantis NV | Sysco | 0.120 |
| Sirius XM Holdings | Citigroup | 0.060 |

Table 2: Shared topics

| **Company** | **Compared Company** | **Topics** |
|---|---|---|
| Citigroup | General Motors | capital markets |
| Cross Country Healthcare | Citigroup | capital structuring |
| General Motors | Citigroup | capital markets |
| Exxon Mobil | Citigroup | capital markets |
| Sysco | Delta Airlines | labor |
| Stellantis NV | Sysco | capital markets |
| Sirius XM Holdings | Citigroup | customer base |

A larger sample size is necessary to go any further, but I believe this metric satisfies the criteria specified above. Its mathematical structure, supported by stopword filtering and applying the distance threshold, makes the metric highly noise-robust. The ability to see which lines dominate the similarity ensures explainability. Finally, the fact that airlines and most automotive companies compared to one another according to topically similar lines sug-

gests that it is reliable.

Assuming this metric holds up against a larger sample size, there are several ways to apply this to Causeway NLP projects. A symmetric similarity metric (taking the maximum similarity between two transcripts) could be used to k-means cluster companies within their sector. Given that the similarity is based on genuine concerns and not vacuous text, this clustering would identify which companies were facing different issues. If we applied this to news, changes in cluster sizes over time could then be used to build a propagation model, which would in turn serve as a basis for an alerts dashboard - the kind several Causeway PM's were interested in for emerging markets. Finally, we could also use the metric in the in-sample decision-tree-based scheme I described in previous interviews. This could be used to to create low-dimensional vectors that could then be used in regressions for earnings estimates.