An Machine Learning Exegesis of Ancient Scripture

Drake Brown

1. Introduction

The scriptures are a wonderful source of information for our religion and contain countless secrets and insights. New research comes out every day on the critical examination of scripture and how to best interpret these divinely inspired documents. However, up to this point in time, most if not all study guides (e.g. topical guide, bible dictionary, guide to the scriptures, ect.) have all been created by individual people with their own biases and foresight. While many of these people were assuredly inspired, they all were limited by the mortal capacity to understand scripture. For this reason I use machine learning techniques, in particular natural language processing with sentence transformers, to better understand the scriptures.

There has been an explosion in natural language processing in recent years. As computing power has increased, we have been able to build larger and larger models that gain a more complex understanding of language. In particular the Transformer (Vaswani et al., 2017) has led to incredible gains in complexity and semantic understanding.

In this work I:

- Fine tune Sentence Transformers on existing King James Version of the Bible and current Modern translations to support better comparisons of old English text to modern conference talks or topic search.
- Modify the pooling operation to get better contextual representation of short passages.
- Fine tune the model on current interpretations of biblical text to obtain better semantic comparisons across scriptures.

2. Background

Transformers are a broad class of architecture that can perform many tasks. A transformer is traditionally characterized by a encoder-decoder architecture. The user feeds in a sequence of information into the encoder, which summarizes the information that is then passed to the decoder that in turn outputs a sequence of information. These can be used for next word prediction, sentence translation, and even question and response. In this work we are mainly concerned with the encoder portion of the Transformer.

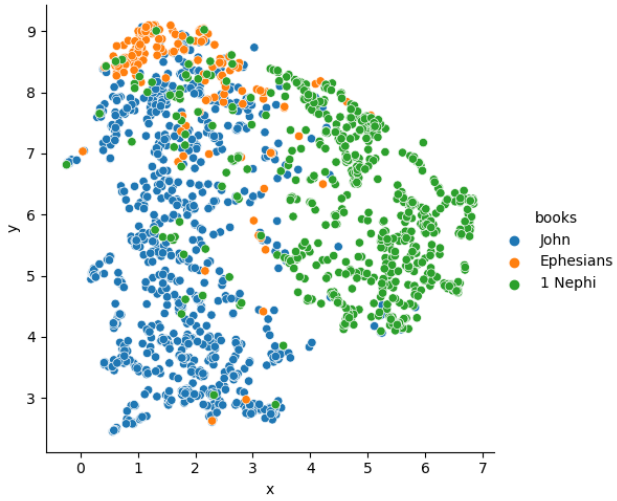


Figure 1. The UMAP projection of 3 books of scripture onto 2 dimensional space. We can see that the different books of scripture gather into clusters. This reflects writing style and contextual similarities found in the same books of scripture

Transformers are able to learn such as broad class of representations due to their attention mechanism. The attention mechanism learns layer wise relationships among elements in a sequence, typically performed using key, query, and value matrices. Typically this method scaled quadratically with respect to the number of input elements, however it is easily parallelizable so it can be computed with many GPU's .

Transformers typically outputs a corresponding vector for every input vector, with the output vector being a semantic representation of the input. Sentence Transformers differ from the norm in that they output a single vector in response to many input vectors. This is normally done by running the sequence through a regular transformer and then averaging across all of the embeddings. Further fine tuning is then performed by maximizing dot product similarity across pairs of training samples (sentences) that are deemed similar (Liu et al., 2019).

3. Results

There are several methods I tried to incorporate to improve the sentence embedding for any given verse of scripture. Typically I evaluated the results by viewing the UMAP (McInnes et al., 2018) graph to identify if such adjustments corresponded to an increased separation among books. Additional evaluations were made by comparing similarity between individual verses and by searching queries(e.g. transform the word "charity" and see which verse is the most similar). For this project I am using the Roberta sentence transformer because it seemed to perform the best.

3.1. Contextual separation

Upon starting this project, the model made pretty good initial estimations on which verses were similar vs dissimilar. However, at times the verses pulled up were out of context and didn't really reflect the actual events in the passage. For example, scriptures that had an "exhort", were often paired with other scriptures that also had that same word even if the exhortations were very different.

To try and alleviate this out of context issue, I altered the pooling operation of the model. Instead of just processing each verse individually, the model would process all of the verses in the chapter at once and then only average over the specific verses. In order to do this, I had to calculate the position of each verse within the tokenization of the entire chapter(this is non-trivial) and then apply a mask to the whole output of the transformer. Additionally, the transformers could only process 512 tokens at once. So I needed to partition each chapter into sub-groups of length 512, run them through the transformer, then concatenate them at the end.

When done on a chapter wide basis, I found that mostly the resulting vectors became non-sensical. If two verses were in the same chapter, even if their pooling was over different indices, they were typically mapped to the same spot as reflected in 3.1. Additionally, the separation between books we saw earlier disappeared. Interestingly, it appears that perhaps these transformers are not suited for processing such long sequences and lose information if used on such a wide scale. Additionally, it seems that the tokens for each vector in the same chapter got mapped to nearly the same spot. When I ran additional tests to verify how similar adjacent output vectors were before pooling I found that with many sentences, vectors in the same sentence had at least a 80% cosine similarity (a lot larger than most verses). This is interesting in the context of these transformer because it implies that when we run it over a lot of sentences, the differences in the outputs start to lessen

To fix this issue I changed the pooling operation from running the transformer over the whole chapter then pooling

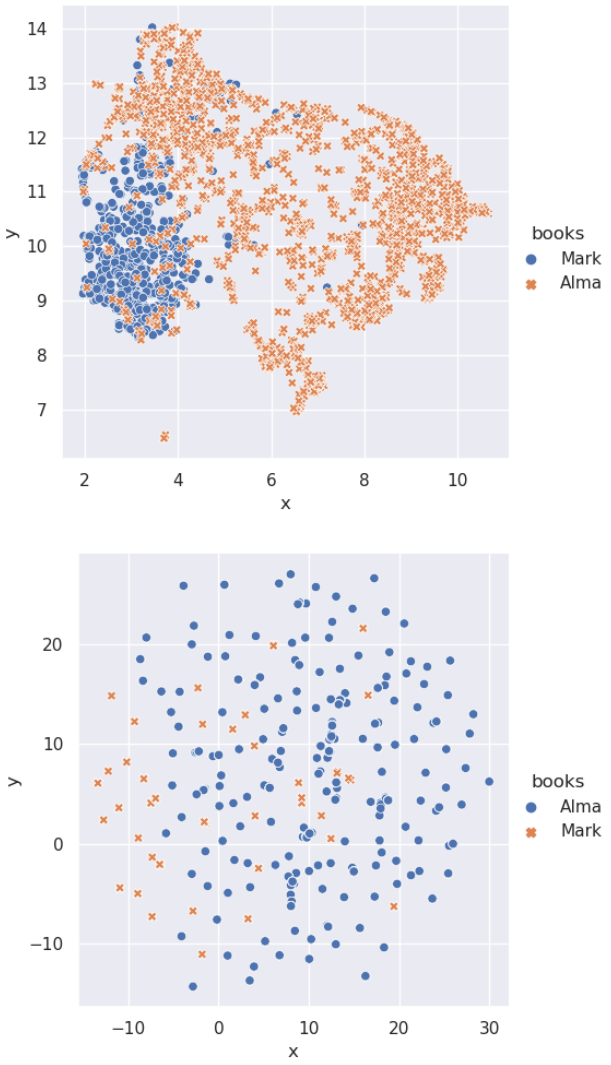


Figure 2. Processing the entire chapter and then sub-pooling resulted in much worse performance for the transformer architecture.

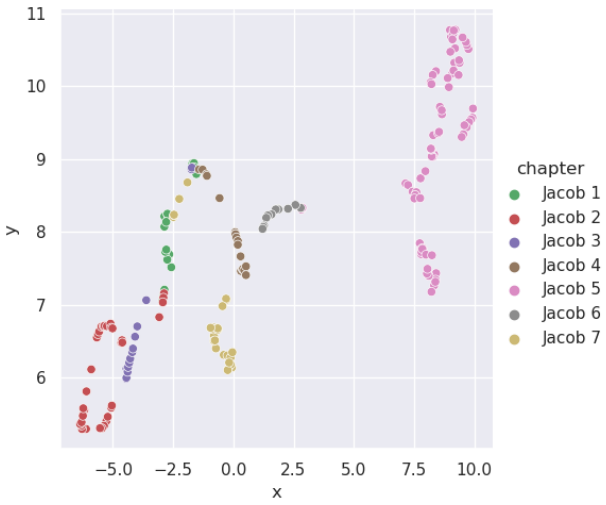


Figure 3. Notice that the chapters seem to form strands on the projection, showing how one idea leads logically into the next in each story.

the verse, to just running the transformer over a sliding window of size 5. This dramatically improved the coherence of the model. Interestingly you can see that the chapters themselves also separate out, in contrast to earlier versions, with the verses themselves seeming to form strands through the projection. Also of note is the fact that some chapters seem to lead into others, perhaps reflecting the fact that the current Book of Mormon chapters are not the same as the original chapters.

3.2. Translation to Modern English

When comparing scriptural verses to conference talks given in previous years I found that the model in general did not infer that sections were very similar to each other even if they did have a similar topic. This is likely due to the difference in wording between the old English and modern English. To improve upon this I fine tuned a sentence transformer using MegaBatchMarginLoss on comparisons between a King James Bible and the World English Version to perhaps teach the model to be invariant to such language differences.

Fine tuning the model on these verses did not result in increased separability by UMAP. However, it did start ranking quotes from general conference to be higher correlated with various scriptural passages. This could be useful in the future for creating a machine learning powered study tool for the whole of gospel library as current resources do not have a similarity ranking between all verses of scripture and all portions of general conference talks.

I used the following quotes and tested which verses they corresponded with changed with fine tuning. Interestingly, most quotes correlated less with with their top hit verse after the fine tuning. This could perhaps either be due to a loss of performance by the fine tuning or the language invariance leading to less similarity. However, the new verses are more correlated than the previous ones. This could lead to a better study guide for scriptures in that one could search for general conference quotes correlating with scriptures to better explain them.

"The reward for keeping covenants with God, is heavenly power. The power that strengthens us to withstand trials, temptations, and heartaches." – President Russell M. Nelson
"I came to realize in a profound way that the same Son of God who atoned for my sins, is the same Redeemer who will also save those who have deeply hurt me." – Kristin M. Yee

Quote	Previous Verse-Score	New Verse-score
Nelson Quote	Moroni 8:26-0.59	Hebrews 6:17-0.46
Yee Quote	DC 138:4-0.45	Romans 5:10-0.29

3.3. Fine tune on interpretations of Verses

I wanted to increase the model's understanding of scripture and fine tune it to all the current prophetic commentary that we have about the standard works. For this reason I obtained a JSON file of all of the institute manual on the new testament. The new testament institute manual contains an explanation for almost every verse in the new testament. I parsed this manual and fine tuned each of the verses on their corresponding explanations in the manual to attempt to encode not only textual similarity but exegetical similarity as well.

Initially, when I fine tuned it there was not much change in the output. I believe this was likely either due to the fact I had trained it for very few epochs (only 5) and there was relatively little data compared to training earlier on the entire bible. Or due to my loss function. To rectify this, I increased my number of epochs to thirty and started using MultipleNegativesRankingLoss which is recommended if you only have positive pairs. Additionally, in the specs I read that an increased batch size improved performance. So I increased the batch size from 16-64.

I iterated through several more ideas to try and increase the effectiveness of the model at separating the books. Each of these ultimately resulted in less defined clusters than we had previously 3.3. It is possible that the model is learning effective interpretation based representations for each verse, it is just not reflected as a common theme across the book.

In future work, I would be interested in fine tuning on the entire institute manual(not just the New Testament) and then cluster the results. It is likely that although the clusters may

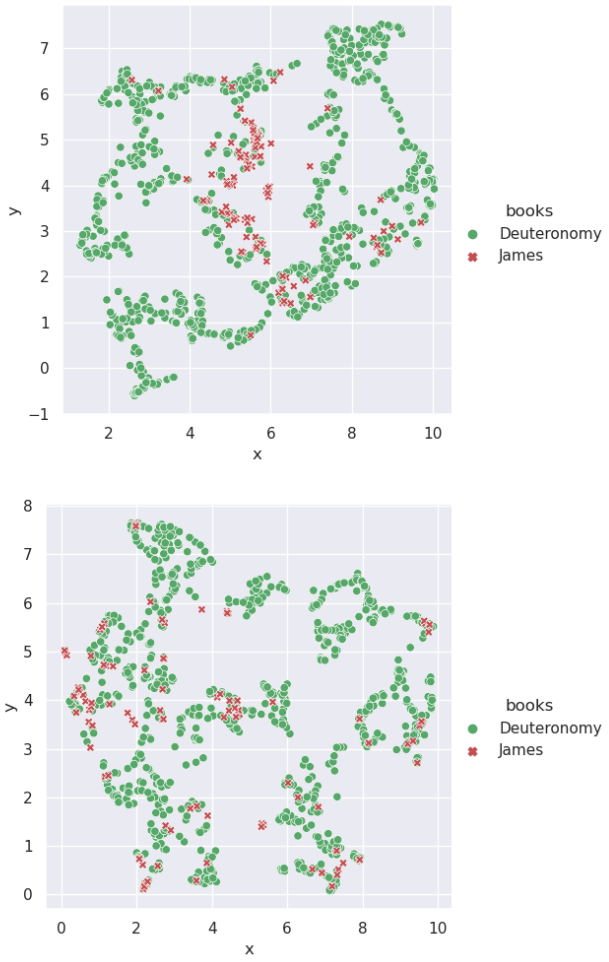


Figure 4. The UMAP clusters before (top) and after (bottom) fine tuning on the institute manual. Initially the book of James organized into a cluster down the middle with Deuteronomy kind of surrounding it. Upon fine tuning the clusters disappeared. This could be due to the interpretations found in the manual not necessarily lining up with stylistic similarities that are found within the books.

not reflect books, they very well might, that other clusters may form based on interpretive information such as charity or faith.

4. Conclusion

Overall, there were many different strategies I attempted to improve the applicability and knowledge of these sentence transformers specifically in the gospel context. Current translations of the bible increased the effectiveness of the model when comparing to modern gospel insights. Additionally, while we found that a sliding window of size 5 greatly enhanced the models context awareness, it is possible that using a model better suited for long passages could produce better results. Lastly, although fine-tuning on interpretations did not result in better defined clusters for classes, there is likely more information gained that resulted in the model capturing better exegetical similarity. Future work could include trying to understand which clusters are formed naturally by these scriptural interpretations to see what topics are most emphasized in our understanding of the scriptures.

References

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. URL https://arxiv.org/abs/1802.03426.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.