Analysis of The Paragraph Extraction Models

- ➔ We have seen that Word Substitution gives us similar words of better quality than Word Stacking, but the no. of similar words is restricted to 4, considering the given constraints. On the other hand, there is apparently no restriction on the no. of similar words in Word Stacking.

- ➔ The main difference comes when we are using a deterministic and a non-deterministic model.

- ➔ Word Substitution works better in a deterministic model, like in Weighted Frequency count. This is because a deterministic model relies heavily on the "quality" of the data that is to be "determined".

- ➔ Word Stacking works better in a non-deterministic model, like word2vec, doc2vec, BERT. However, there is an anomaly related to the performance of doc2vec with Word Stacking. This is because, these models are heavily depending on "how much data is available".

- ➔ Weighted Frequency Count is a deterministic model. This makes it heavily dependent on the quality of similar words it is getting from the pre-trained model. The only thing it does is counts the weighted frequency. And this weight is coming from a pre-trained model from where we are extracting the words. This leads to high coupling and low cohesion among the two processing models.

- ➔ Word2Vec is a non-deterministic context insensitive model. We are using a greedy strategy to get the paragraph vectors. And this depends directly on how many similar words you can get. More the data, better is the approximation. Since, the pretrained data has a huge vocabulary, getting decent results was pretty much expected. It solves the coupling and cohesion issues. But, the problem of dependency on the greedy strategy still persists.

- ➔ Doc2Vec is again a non-deterministic context insensitive model. Here, our model is incorporating paragraph identifiers in the training portion. This works decently on large documents if the quality of query paragraph (similar words) is good, that is, if Word Substitution is used. However, it miserably fails on small datasets because the model is solely standing on the Document, and since the document is small, the paragraph vectors are very random.

- ➔ One thing to note is that, it fails on all kinds of data if Word Stacking is used. Intuitively, this should not have been the case. However, there is a trade-off between the quantity of similar words and the quality of similar words. And in this case, the quality of similar words wins. This is because the Doc2Vec model results are heavily dependent on the unidirectional context of the training-data, which is again dependent on the quality of the query paragraph, that is, the quality of the similar words. So, the problem that persists is the context that is not taken into account, and the anomaly of dependency on the quality of similar words which is unusual on non-deterministic model.

- ➔ BERT is a non-deterministic context sensitive model. Here, our model is taking the bidirectional context into account. And since we are using a pre-trained Sentence Transformer model, the problem of depending entirely on the document (like in case of small documents) is also removed. Also, the anomaly of dependency on the quality of similar words is removed since the context is bidirectional. Hence, we are getting good results with Word Substitution, and better results with Word Stacking.

Inference

We can conclude that Word Stacking to expand the query and a BERT Sentence Transformer pretrained model to process the document performs pretty well considering an unknown dataset. The above discussed methods can potentially be useful in a variety of domains, beginning from basic searching to complex information extraction, to recommendation engines.