Unsupervised Summary Generation of TED Transcripts

Anshul Gupta (IMT2014006)
Pranav S. (IMT2014039)
Suprgya Bhushan (IMT2014056)

Abstract

Text summarization has become an important and timely tool for assisting and interpreting text information in today's fast-growing information age. It is the task of extracting salient information from the original text document. By writing a transcript summary, you digest the record down to what is essential. There are two fundamental approaches to text summarisation; extractive and abstractive. Extractive summarisation is the strategy of concatenating extracts taken from a corpus into a summmary, while abstractive summarisation involves paraphrasing the corpus using novel sentences. We propose a unique approach towards exhaustive unsupervised text summary generation by using a measure of 'distance' between sentences to capture new 'concepts' that should be included in our summary.

Dataset

The dataset was downloaded from Kaggle, and consists of the transcripts of English TED videos uploaded on the official website of TED until September 21st 2017. This amounts to a total of 2468 transcripts. There is no ground truth for the dataset with which we can compare our summaries.

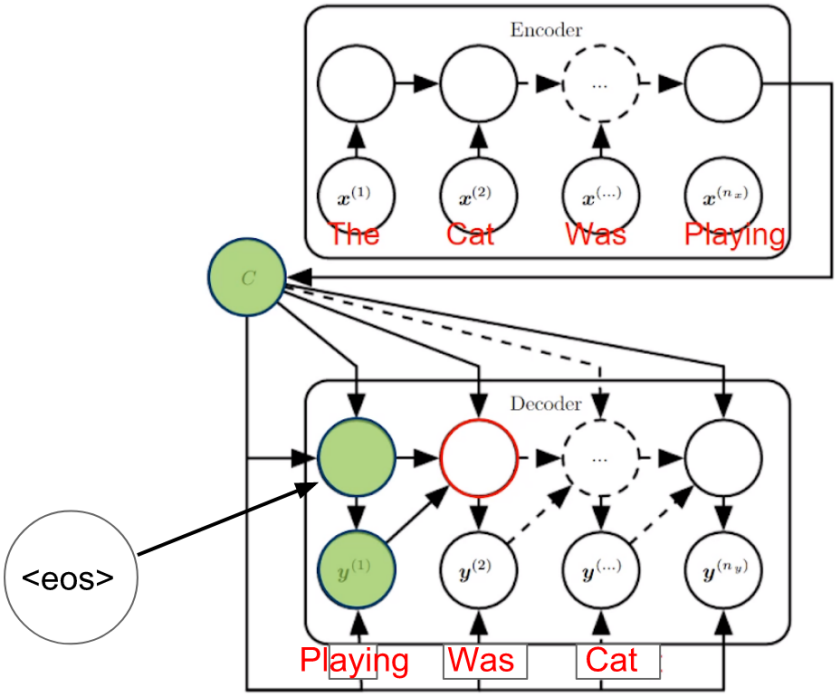
Method

Our method consists of three steps:

1. Generate sentence embeddings
2. Get distances between sentences (using sentence embedding vectors) in a transcript
3. Generate a threshold value to decide which sentences to keep

We first generate sentence embeddings for every sentence in a transcript using an AutoEncoder. This low dimension representation will help capture the notion of distance between sentences by grouping similar sentences together, and keeping dissimilar sentences further away. Our AutoEncoder consists of an encoder GRU (Cho et al, 2014) and a decoder GRU. The encoder generates a context for the source sentence and the decoder uses this context to generate the reversed source sentence. We use the reverse of the source sentence as our target as inspired from Sutskever (2014) to introduce short term dependencies and make the optimization problem easier.

The encoder is fed with a 200 dimensional embedding vector for a word at every time step until the end of the sentence. The final hidden state serves as a context for the decoder, that uses it along with the previously generated word and previous hidden state to generate the next word. Once the network has been trained, we use the context generated from the encoder as a sentence embedding for any source sentence.



After we have obtained sentence embeddings for every sentence in the transcript, we calculate the euclidean distance between consecutive sentences and normalize it to a value between 0 and 1. We hypothesize that for every new 'concept' that is introduced in the talk, the distance between the preceding sentence and that sentence will be large. Hence, by keeping every sentence whose distance from the previous sentence is above a certain threshold we should get a good

summary of the talk.

To generate this threshold value we perform a grid search for different values ({ 0.2, 0.4, 0.6, 0.8 }) and try to minimize a loss function. This loss function is a function of the distance for sentences not included, and the number of sentences included. We try five different loss functions:

1. $arg\ min_\theta \sum_{\texttt{sentences not included}} -log_{10}(\texttt{d}) \ + \ \texttt{num\_included}$
2. $arg\ min_\theta \sum_{\texttt{sentences not included}} -log_{2}(\texttt{d}) \ + \ \texttt{num\_included}$
3. $arg\ min_\theta \sum_{\texttt{sentences not included}} (\texttt{d}) \ + \ \texttt{num\_included}$
4. $arg\ min_\theta \sum_{\texttt{sentences not included}} e^{(\texttt{d})} \ + \ \texttt{num\_included}$
5. $arg\ min_\theta \sum_{\texttt{sentences not included}} 2^{(\texttt{d})} \ + \ \texttt{num\_included}$

where θ is the threshold value, and d is the distance from the previous sentence.

Results

Loss Function	Best Threshold
1	0.8
2	0.8
3	0.8
4	0.2
5	0.2

In the first three loss functions, the first part increases very slowly as a result of which the largest threshold value is always chosen to minimize the second part. Further, while the negative logarithm represents an intuitive idea of information lost, being a monotonically decreasing function we get lower values for higher values of distance which should not be the case.

We tried the fourth and fifth functions as they are monotonically increasing and have a steeper slope. However, we encounter the opposite dilemma of rapid increase in the first part of the loss function for higher values of threshold, as a result of which the smallest threshold is always chosen in order to minimize the first part.

Discussion

We successfully generated summaries for transcripts in a completely unsupervised manner. However, since the computation time was large, the summaries of only a few transcripts could be generated. As part of future work, different loss functions can be experimented with to come up with more varied threshold values. Attaching weights to the two parts of the loss function is an idea that can be explored too. Lastly, one can also try a different, possibly faster sentence embedding model.

References

1. Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

2. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

3. Zha, Z. J., Yu, J., Tang, J., Wang, M., & Chua, T. S. (2014). Product aspect ranking and its applications. IEEE Transactions on knowledge and data engineering, 26(5), 1211-1224.

4. Mani, I., & Maybury, M. T. (1999). Advances in automatic text summarization. MIT press.

5. Nallapati, R., Zhou, B., Gulcehre, C., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023.

6. Gupta, V., & Lehal, G. S. (2010). A survey of text summarization extractive techniques. Journal of emerging technologies in web intelligence, 2(3), 258-268.

7. Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In Mining text data (pp. 43-76). Springer, Boston, MA.