

PROBLEM

Text summarization is a meaningful way to handle large amounts of text data. For this project, we worked on improving the existing text summarizer, Google's Textsum. Additionally, we explored transfer learning, whether a text summarizer trained on CNN/DailyMail can be used to summarize Amazon book reviews.

We introduced a novel approach, using a universal sentence encoder and word embedding to improve the Textsum model. With the newly improved model, we trained on the CNN/DailyMail dataset and applied it to Amazon book reviews to qualitative test the accuracy. To our knowledge, there haven't been any results published with this approach.

DATASET

Google's Textsum was developed using the Gigaword dataset, not open sourced and inaccessible by Stanford students. However, recent research on summarizer has moved toward using the CNN/DailyMail dataset. This dataset has the original text and a short set of summarized bullet points that represent meaningful "highlights" of each article.

The CNN/DailyMail contains 300k+ articles, and each article includes 3 to 4 bullet points. We use the first two sentences of each article as model input, and the first bullet point as the gold label sentence. We split the dataset to train/dev/test as 80:15:5.

EXPERIMENTS AND RESULTS

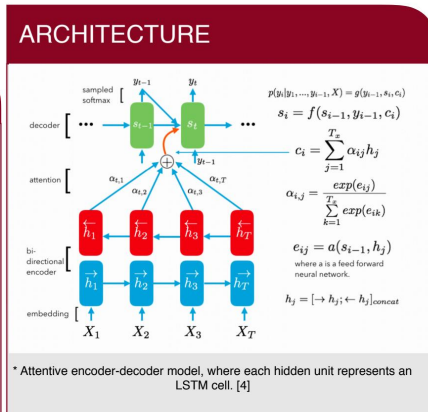
- TextSum:** We used an open source Tensorflow model, TextSum, as our baseline.
- TextSum + ConceptNet Numberbatch(CN):** Because of the limited amount of training data in CNN/DailyMail dataset, we initialized our word embeddings with ConceptNet Numberbatch to give our model a more powerful semantic representations of the source input tokens. (CN performed the best in our model compare to word2vec, GloVe, etc.)
- TextSum + ConceptNet Numberbatch(CN) + Universal Sentence Encoder(USE)** Instead of using the first two sentences, we applied Universal Sentence Encoder to find the two most similar sentences to the gold label sentence then feed them into TextSum.

Pre-processing	Learning Model	Rouge-1	Rouge-2	Rouge-L
None	TextSum	0.078	0.004	0.0725
Tokenized	TextSum + CN	0.108	0.019	0.102
Tokenized	TextSum + CN + USE	0.122	0.036	0.110

TextSum

Textsum uses an encoder-decoder model with a bidirectional LSTM-RNN encoder and an attentive unidirectional LSTM-RNN decoder with beam search. The encoder-decoder model is trained end-to-end.

For the first decode timestep, we feed in the last output of the encoder as well as the embedding for the start (<S>) token. For subsequent decode timesteps, the decoder uses the last decoder output in addition to the work embedding for the previous word to generate the next word. During the training step, the previous word is the actual previous word from the gold label, but during the decoding step, the previous word is the previously-generated word. The decoding process will continue until the generated summary reaches the max decode length set at 30, or until it generates an EOS token.[1]



Application - Transfer Learning on Amazon Reviews

"This is a great book. It has twists and turns and many surprises. One of those stories I would read again. There is so much I am sure I missed something."

→ "I made for great payment, simply for our <UNK>"

DISCUSSION

Similar to other published results, the decoded results we produced from our baseline model is inadequate with the CNN/DailyMail dataset. After training on 80% of the data set, we noticed a large number of <UNK> tokens in the summaries which made it unreadable. However, once we initialized the model with word embeddings, ConceptNet Numberbatch, the decoded results had significantly less <UNK> tokens. Additionally, after we introduce Universal Sentence Encoder into the model, we obtained the best results.

Once we applied the model on Amazon reviews, we got nonideal decoded results. Since the diction between news articles and Amazon reviews are vastly different, the decoded results followed the typical news vernaculars on Amazon reviews. However, we were surprised to find that the model showed good sentimental understanding: on a positive Amazon review the decoded message showed a positive news piece while on a negative Amazon review the decoded message showed a negative news piece.

FUTURE WORKS

There are certain areas for improvements. If we had six more months on this project we would like to: first, add Amazon reviews and other nonnews data sources to the training dataset. Second, tune the model by improving the current complex TextSum architecture. Third, instead of using only two sentences for training, we will use more sentences from the article to train the model. Finally, the model takes about a week to train on 1 GPU, we would like to purchase more GPUs and computing time to fine-tune the parameters of the model.

REFERENCES

- [1] Vincent C., Eduardo M., Liezl P., & Danqi C. (2017). An Examination of the CNN/DailyMail Neural Summarization Task.
- [2] Cer, D., Yang Y., Kong S., Hua N., Limtiaco N., St. John R., Constant N., Guajardo-Cespedes M., Yuan S., Tar C., Sung Y., Strope B., & Kurzweil R. (2018). Universal Sentence Encoder.
- [3] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation By Jointly Learning To Align and Translate. ICLR 2015.
- [4] Nallapati, R., Zhou, B., Santos, C. N. Dos, Gulcehre, C., & Xiang, B. (2016). Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. *Proceedings of CoNLL*, 280-290.