CS230

Evaluating text summarization methods with transfer learning

Princess Vongchanh
Department of Computer Science
Stanford University
vongchan@stanford.edu

Abstract

Powerful abstractive text summarization methods have become widely accessible thanks to recent work combining Transformer models with self-supervised pre-training methods. However, such models are typically pre-trained on large public corpuses whose data shares the same syntactic structure (such as news articles and legislative bill summaries), and ought to be evaluated on large texts whose syntactic structure differ. In this project, I explore a combination of abstractive and extractive text summarization methods in hopes of producing a summary which is fluent (semantically and syntactically correct) and comprehensive. These metrics are evaluated using Cosine Similarity, ROUGE-1, ROUGE-L, and BERTScore. I found that PEGASUS can correct some syntactic errors, and transferring its learning to BART improves fluency and semantic preservation overall.

1 Introduction

While text generation methods grow increasingly better at embedding meaning into words and sentences, the limited context of texts they are trained on or prompted by puts them at risk of covert bias. Object writing is a complex challenge for even the most professional writers. By increasing the diversity of domains where it is applied, significant parts of human life may be improved by machine text generation. One area which may benefit from abstract text summarization in particular is voter participation, which is in part limited by how inaccessible the process of becoming well-informed is. This issue is seen locally, as the Santa Clara County Civil Grand Jury published a report this year (2022) to address how voters who lack the time and resources to properly participate in government "are being deceived" by complex and misleading language. Even PEGASUS, a state-of-the-art Transformer model published only two years ago, was trained on legislative bill summaries, rather than legislative bills themselves.

Nonetheless, PEGASUS excels as a text generation task and produces abstract text summaries whose semantic preservation is comparable to others of its kind. To explore the role of abstract text summarization methods in processing objective texts, this project explores two methods of transfer learning: 1) an abstractive summarization model, followed by an extractive summarization model (abstract-to-extract method) and 2) the same abstractive summarization model, followed by another abstractive summarization model (abstract-to-abstract method). The first model used in both methods is PEGASUS, which inputs sections of raw text from Colorado Proposition 122 as strings, or sequences, inside of a list, and outputs a concatenated abstract summary of each section. This concatenated summary then became the single-sequence input for TextRank, which outputs an extractive summary based on sentence relevance. It also became the single-sequence input for BART, which will output another abstracted summary of CO Prop 122. The goal of applying additional

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

algorithms is to understand how models may build off of one another to achieve a specified outcome (an accurate summary of the Prop that even unskilled readers can understand).

2 Related work

Khatri et al. (2018) used an extractive summarization technique to identify significant information in a text by framing context vectors as frequent n-grams that appear in sentences and encoded these contexts with significant attention. This is a clever approach for attention modeling; PEGASUS performs similarly by masking significant words/sentences from a sequence such that they are "extracted." They differ by PEGASUS's ability to generate new tokens, which is particularly useful for complex texts.

Unlike Khatri et. al's model which applies extraction before abstraction to identify significant words and phrases, a study led by Miaschi and Dell'Orletta (2020) shows that Transformers produce more meaningful word embeddings because they can encode sentence-level properties within single-word embeddings; this makes them a powerful pre-training method. They specifically analyze BERT, which is the basis for state-of-the-art Transformers today, including PEGASUS.

Cohan et al. (2018) developed a promising abstractive summarization model for single, long-form documents after recognizing that their sections are poorly identified by models pre-trained on corpuses of relatively short (<1000 tokens) texts. To solve this, they built a hierarchical encoder that models the discourse structure of a document. This technique proved to be effective for essay-like documents, but it may not perform as well when sections are well-defined with little to no transition between them, such as in legislature.

Mihalcea et al. (2014) identified that relevance between two sentences can be determined by the number of common tokens they share, regardless of their sequence position and weight embeddings. This research has been widely adopted as the Cosine Similarity function, which I will use to ensure that the final summary's flow of information allows it to be easily read, regardless of the Proposition's original structure.

Krishna et al. (2020) find that disparate pseudo data doesn't hinder the encoder-decoder itself; instead, the algorithm may additionally benefit from this data in training because it pays attention to a wider range of embeddings. Although I am not training a tokenizer for custom word embeddings or fine-tuning an algorithm, this affirms the benefit of recycling a model's output as another's input. The abstract-to-abstract method of my project specifically benefits from learning a different, arguably more natural, sentence syntax.

3 Dataset and Features

The long, single-document used in this project is a PDF of Article 170: the Natural Medicine Health Act of 2022, or Colorado Proposition 122 [6], which was approved earlier this year. It was chosen above other legal texts precisely because it is a ballot measure which was both highly viewed and highly agreed upon. Preprocessing of the document includes turning it into a TXT file of all lowercase characters, removing numbers and bullet points with RegEx, combining bullet points in a single sentence separated by semicolons, and finally a manual review to separate it into sections based on topics. In the latter step, I discovered some classical errors with converting special fonts into text, such as random spaces between words and unnecessary whitespace surrounding paragraphs; some of these I returned to the code to correct, and others were miniscule enough for manual correction when I saw them.

Pre-existing libraries and tokenizers were used for all tokenization and normalization processes. I configured PEGASUSTokenizer's built-in normalization process to pad sequences so that each input was the same length. PEGAUSTokenizer [7] also utilizes its own pre-trained sub-word embeddings, which is unique and difficult to extract embeddings from without fine-tuning the tokenizer. The tokenizer was allowed to truncate sequences containing more than 1024 tokens. The sentence_transformers library allowed for sentence tokenization based on BERT word embeddings [8] with little to no manual tokenization or normalization of the inputs. When applying TextRank [9], sentences less than 10 words in length were removed to account for section titles having been

maintained in the aforementioned data cleaning steps. Lastly, BARTTokenizer performs similarly to PEGASUSTokenizer and does not require additional tokenization or normalization steps.

Two hand-written Prop 122 summaries, one from Ballotpedia and the other its official ballot text [10], were used to measure fluency and comprehension and required no pre-processing.

4 Methods

Method 1: abstract-to-extract

Sections of the raw text are input to a PEGASUSForConditionalGeneration model [11] , and then concatenated to form a full abstract summary of the text which contains a similar amount of significant information from each section. Then, this concatenated summary is transferred to the TextRank algorithm in combination with BERT word embeddings.

PEGASUSForConditionalGeneration is a strong candidate for text summarization because it utilizes a language modeling head (LMH). The LMH can be framed as a global attention technique, "reminding" unigrams of their grammatical and semantic potential beyond local context; this is comparable to performance without a LMH, which may result in grammatically correct, but semantically incorrect or inaccurate, sentences. TextRank applies the Cosine Similarity function to determine how relevant sentence pairs are to one another and generate a similarity matrix. This matrix is then fed into the PageRank [12] algorithm for ranking based on relevance. BERT word embeddings were chosen to generate sentence embeddings because BERT and PEGASUS similarly embed meaning via BPE. I was unable to apply BERT sentence embeddings to the PEGASUS model without undergoing the computational expense of fine-tuning PEGASUS itself. I similarly do not utilize models for long-form documents (such as Longformer and PEGASUS-X) because of their computational cost.

Method 2: abstract-to-abstract

The first model of this method is the same one used to output a full abstract summary by concatenating section summaries. The second method is BARTForConditionalGeneration [13], which uses a standard sequence to sequence architecture with BERT as its encoder and GPT-2 as its decoder. This suggests that word and sentence embeddings by BARTTokenizer are similar to those created by BERT, with weight changes made only to masked tokens in the decoding stage. BERT, BART, and PEGASUS each encode with rigorous masking functions, though they are consecutively more challenging and produce comparable outcomes. BARTForConditionalGeneration also uses a LMH.

5 Experiments/Results/Discussion

BERTScore F-1 and ROUGE-L F-1 metrics were used to determine how generated summaries compared to professionally-written ones; these summaries will be referred to as targets and references, respectively. BERTScore measures the similarity score of individual target words to each reference word; the method effectively analyzes semantic meaning, and is popular for quantifying semantic preservation. ROUGE-L identifies the longest matching sequence of words between targets and references; it is an effective comparison metric because both the targets and references are significantly shorter than CO Prop 122 itself in their attempt to highlight only the most significant considerations made in the document. Additionally, the two references differ in their fluency and semantic preservation as well: reference 1 fluently conveys Prop 122's purpose, while reference 2 comprehensively conveys its range of content. One key metric for comparing the methods was defined overall: I added the average ROUGE-1 F-1 score across both references to the average BERT F-1 score across both references. The experimental method which produced the highest overall score would be considered the most adept at abstractly summarizing legislation.

My hypothesis that PEGASUSForConditionalGeneration would struggle to handle syntactically dissimilar texts from those it was pre-trained on was correct. More specifically, the section summaries it generated varied in fluency, some of them a sequence of full sentences and others a sequence of partial sentences separated by a period. The concatenated section summary scored 0.846 (an average BERT score F-1 score of 0.431 and an average ROUGE-L F-1 score of 0.415); although this is high accuracy, I was motivated to do better.

Method 2 produces the most fluent and comprehensive abstract summaries of ballot measures. It scores significantly higher than Method 1 at 1.356; Method 1 scored 1.296. In order to understand why this might be, I looked to its ROUGE-1 F-1 and BERT F-1 scores.

The BERT scores of both methods were higher when calculated against reference 1; this demonstrates that they were more fluent, or semantically meaningful and syntactically correct, than comprehensive. However, Method 1's average ROUGE-L score was lower than that of the concatenated section summary. Manual review of the summary shows that the shorter, partial sentences were ranked as the most relevant. This is reasonable because each word in a partial sentence is likely to appear in other sentences, while longer sentences may contain unique words with little relevance elsewhere. Additionally, the ROUGE-L score compared against reference 2, the more comprehensive hand-written summary, was higher than it was against reference 1. While reference 2 adequately conveys different sections of Prop 122, both its semantic meaning and syntactic correctness are limited. Even without manual review of Method 1's summary, this implies that its structure is more similar to reference 2's. Overall, this means that an extractive summarization techniques which learns from an abstractive technique results in better summaries than the standalone abstractive technique, though it is unable to meaningfully increase semantic preservation and syntactic correctness of legislation. I infer that TextRank alone would have performed worse had PEGASUS not pre-processed Prop 122's especially complex sentences. This is significant when considering how this project may be applied to the real world: voters who approach nonsensical ballot measures may be discouraged from developing further understanding.

Meanwhile, BART, with learning transferred from PEGASUS, performed 60% significantly better than the standalone PEGASUS model. Both the concatenated section summary and final target resulted in higher ROUGE-L and BERT scores compared against reference 1; this is unsurprising, since abstractive techniques are designed to highlight significant information. This did not pose a limitation to comprehensiveness though: the target resulted in a higher average ROUGE-L, meaning that method was able to represent different sections of Prop 122 well. Manual review of the target affirms that this method also effectively handles sentences of varying syntax: the partial sentences had been removed or corrected, and complex, yet syntactically correct, sentences had been generated. This may be due to the structure of BART's input, whose range of ideas were individually expressed in two or three sentences such that the model did not weigh specific topics or words higher than others.

6 Conclusion/Future Work

Experiments conducted by this project make evident that abstract-to-abstract transfer learning methods are better suited for long single-document text summarization than abstract-to-extract methods. The former method allows for previous semantic or syntactic errors to be corrected, and more comprehensive details to be extracted when key points have already been identified. The project, however, is limited as an exploration of applications; by fine-tuning Transformer models and customly pre-training their tokenizers, a study of similar structure may result in greater and disparate findings. I leave this task up to future researchers, as well as look forward to other model combinations on the road to automatic text summarization.

7 Contributions

This project was performed individually with no outside contribution.

References

[1] Khatri, C., Singh, G., Parikh, N. (2018). Abstractive and extractive text summarization using document context vector and recurrent neural networks. arXiv preprint arXiv:1807.08000.

[2][9] Miaschi, A., Dell'Orletta, F. (2020, July). Contextual and non-contextual word embeddings: an in-depth linguistic investigation. In Proceedings of the 5th Workshop on Representation Learning for NLP (pp. 110-119).

[3] Mihalcea, R., Tarau, P. (2004, July). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing (pp. 404-411).

[4] Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. arXiv preprint arXiv:1804.05685.

[5] Krishna, K., Wieting, J., Iyyer, M. (2020). Reformulating unsupervised style transfer as paraphrase generation. arXiv preprint arXiv:2010.05700.

[6][10] Ballotpedia. (2022). Colorado Proposition 122, Decriminalization and Regulated Access Program for Certain Psychedelic Plants and Fungi Initiative (2022). https://ballotpedia.org/Colorado_Proposition_122,_Decriminalization_and_Regulated_Access_Program_for_Certain_Psychedelic_Plants_and_Fungi_Initiative_(2022).

[7][11] Huggingface. (2022). Pegasus. https://huggingface.co/docs/transformers/model_doc/transformers.PegasusForConditionalGeneration.

[10] Huggingface. (2022). https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens.

[12] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[13] Huggingface. (2022). https://huggingface.co/docs/transformers/model_doc/bart.

[14] [15] Paszke, A. et al., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035. https://torchmetrics.readthedocs.io/en/latest/.

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.