Improving Readability in Summarization Procedures

Varun Dashora and Marcus Manos

9 March 2022

1 Abstract

Extractive text summarization can be affected by several preprocessing steps including stripping grammar, casing, tokenization, and data masking. One of the more unexplored factors is data masking; the masking of select tokens forces the model to predict the hidden word and has the potential to enhance the readability of extractive summarization. The State of the Art approach is to randomly mask words in the corpus, which is good for imitating the distribution of words in the original text but not necessarily for creating a summary of the text. By analyzing the summaries from similar pieces of text we were able to randomly mask words by examining their parts of speech and assigning them a probability proportional to their distribution in the summary. We tested our theory out on the "sshleifer/distilbart-xsum-12-6" model developed by Sam Shleifer using the BBC News data set to fine-tune the model and validate our assumptions. [1] Overall, we notice a slight increase in readability after fine-tuning for both masked and non-nasked data.

2 Introduction

The goal of extractive text summarization is to accurately summarize the text by splicing together sentences that are the most relevant to the overall message. This can be done via several methods, the current state-of-the-art approach being the use of transformer models and their self-attention layers to find the most important segments of the text to include in the summary. Such models identify important parts of the original article by learning where and how they are mentioned in the summary itself. Two of these models are BERT and BERTSum, with BERTSum considered as the state of the art in extractive text summarization. While advances in extractive text summarization are increasingly applicable in real-world scenarios there remain problems. An issue with extractive text summaries is that since they can only use and re-arrange words from the article itself, there are issues with readability and flow. Sentences are sometimes worded poorly when salient parts of the article are selected and spliced together, since different parts of the article sound clunky to read one after another.

To address this issue, we explore how fine-tuning pre-trained models can impact readability. Such a goal necessitates some type of readability measurement; we opted to use the GRUEN metric, which measures coherence, focus, and grammar and totals each subscore to approximate overall readability. One major choice we investigated for fine-tuning was whether or not the data was masked. Certain models take advantage of this technique by hiding select words so the transformer can predict them from the context to the left and right of the masked word, which encourages a type of spatial locality and focus. The current masking-based approach focuses on implementing random masking uniformly across the dataset, meaning that all words will have an equal chance of being focused on by the self-attention mechanism. This means that the distribution of masked words will ultimately follow the distribution of parts of speech in the text; if the text is made up of 60% nouns, then around 60% of the masked word will be nouns. This process creates summaries that contain relevant information at the expense of writing quality.

Since masking relies on extracting sentences from the summaries, it can be hard to understand, rendering it useless for many business applications. It is for this reason that we wanted to improve the readability of extractive text summarization while retaining its relevant information. To this end, we examine the impact of masking different parts of speech, choosing to emulate the distribution of parts of speech in the summary.

3 Background

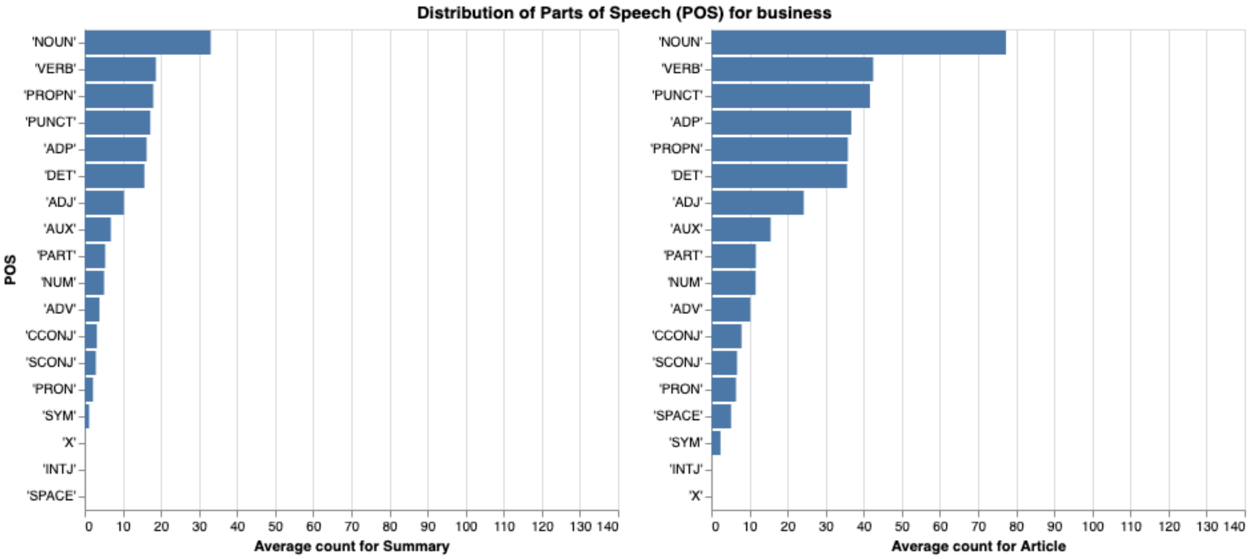
There have been several attempts to address summarization in general with varying model architectures. First, there is a modification of BERT called BERTSum that has been fine-tuned specifically for extractive text summarization [2]. In addition, an abstractive text summarization model named PEGASUS utilizes sentence masking to get a better sense of how to extrapolate information from an article, which in turn eases summarization [8].

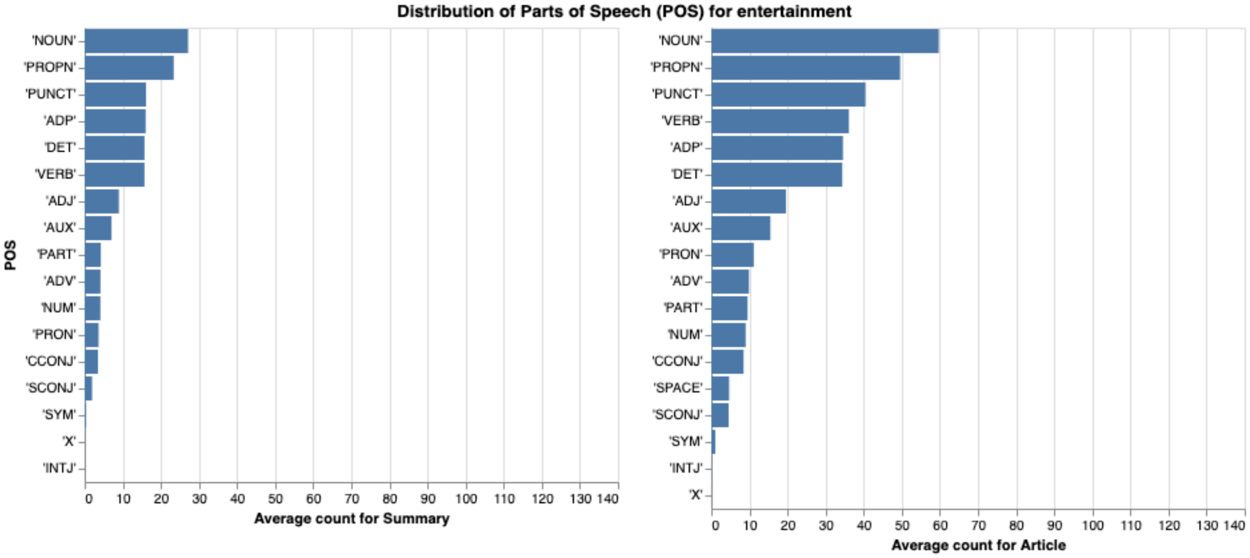
Progress regarding non-transformer models is also present. Researchers have been exploring how to use the article's natural organization and hierarchy with an encoder-only model [4]. A third attempt at improving extractive summarization involves not a neural network, but a generic multi-iteration computation that identifies relationships between sentences to avoid redundancy in the final summarization [6]. A fourth attempt evaluated the effect of masking length on masked language models [7]. This study analyzed how different masking lengths affected the ROGUE scores of masked language models using a variety of masking techniques ranging from span to entity-level masking. In addition to model architectures, there is an evaluation metric very useful for gauging readability. This is the GRUEN metric, which assesses a text for grammatical correctness, coherency, and focus [9].

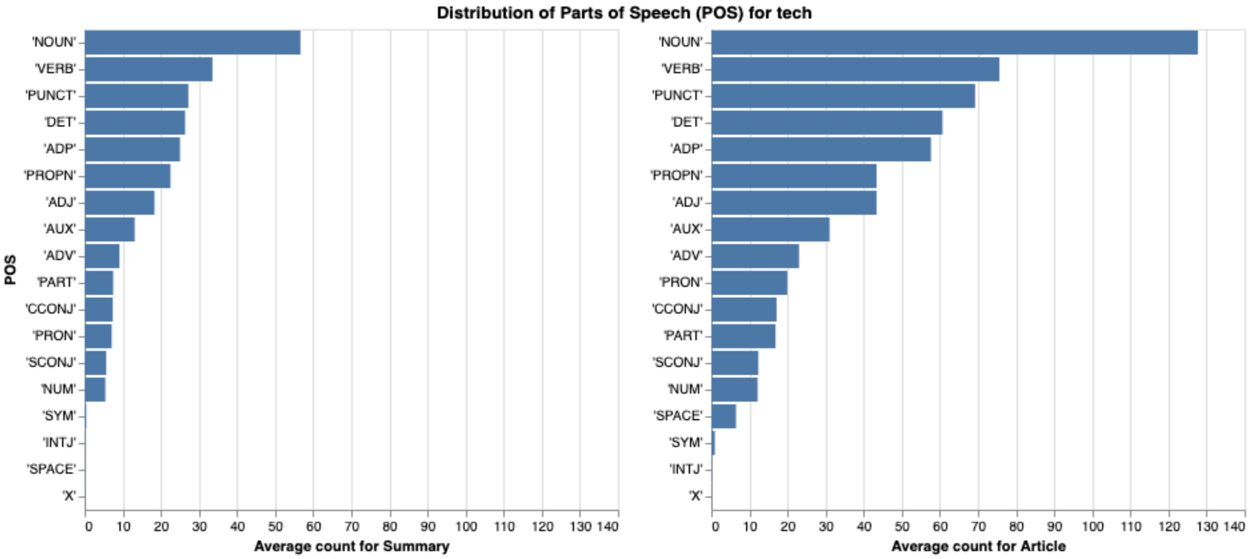
4 Methods

4.1 Part-of-Speech Exploration

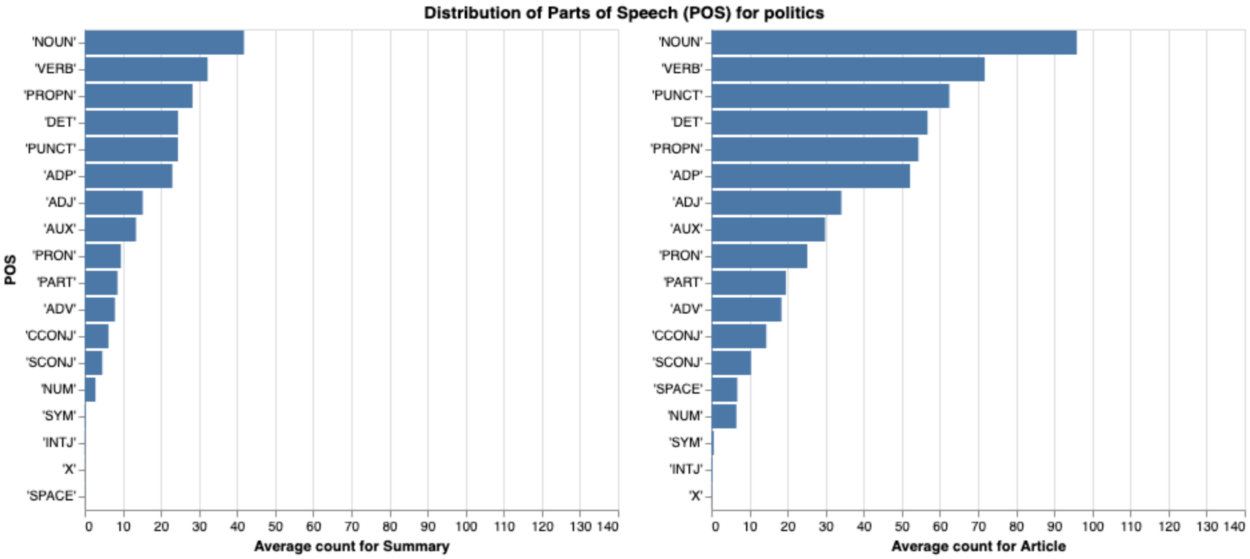
Typically, words are masked randomly across the dataset, which results in roughly the same proportion of words being masked that make up the text. This method may not necessarily be the best as the makeup of a summary is often different from the makeup of the text. We began our process by examining the distribution of the parts of speech for 100 summaries across each of the five categories in the BBC news dataset. We validated our assumptions that the distribution of parts of speech was indeed different in summaries versus articles, and discovered the distributions varied depending on the article's category. Next, we compared the distribution of parts of speech within each category versus each other, again this led us to some interesting results (as seen in the below visualizations). We initially thought that each category would have a relatively similar distribution to summaries since they all require condensing information down into a format that's easy to digest but, we discovered that for certain parts of speech and categories there was a large difference. One of the most prominent examples being the average occurrence of Nouns in the technology genre was nearly double that in the entertainment genre. Given this stark difference, we decided to keep the summaries separated into their genre.



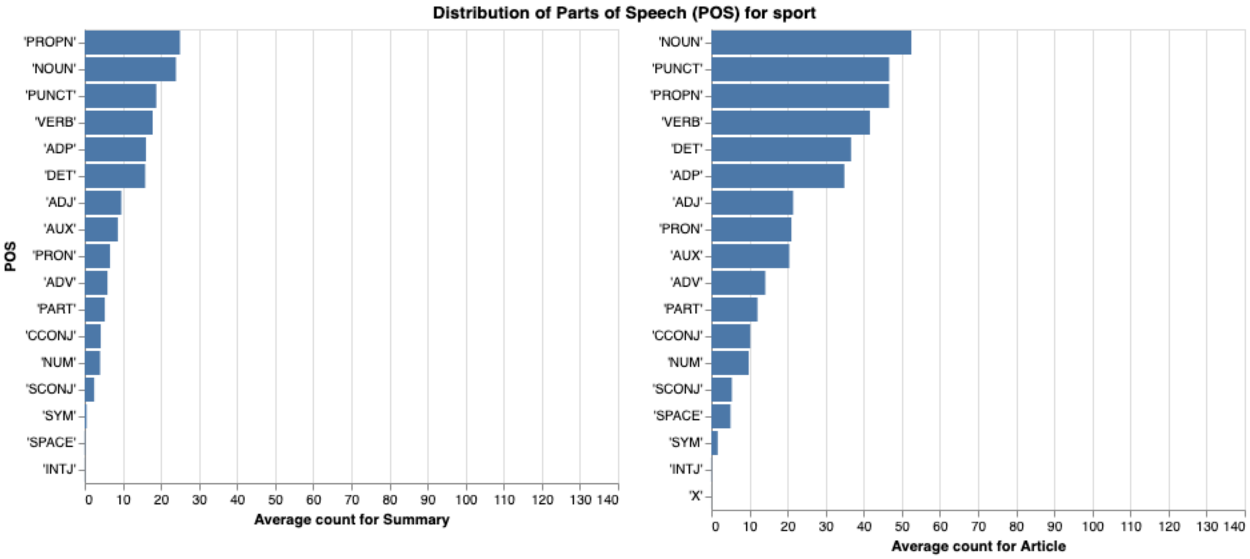




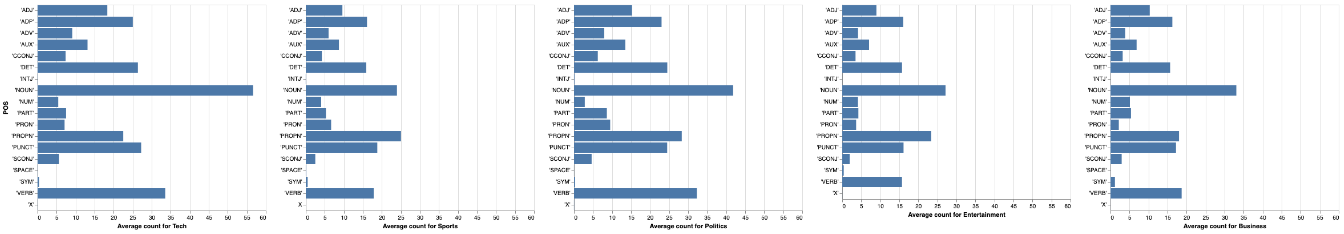
Distribution of Parts of Speech (POS) for tech



Distribution of Parts of Speech (POS) for politics



Distribution of Parts of Speech (POS) for sport



Since we believed the main regarding surrounding readability was the BERT model focusing on the masked words in proportion to the article rather than the summary we implemented masking according to the average word distribution for each article category. To do this we used a spacy model trained on the ”en_core_web_md” corpus and predicted the probability of masking according to the distribution of parts of speech that made up the summary (if adjectives accounted for 8% of the words in the summary, then the probability of masking a word if it was an adjective would be 8%). To examine the difference we fine-tuned an existing BERT model (Distil Bert, pre-trained on the xsum dataset) with a randomly masked dataset with a probability of .20, and one that was masked according to the summary. Since our dataset was limited in size we randomly selected 30 groups of 100 articles for testing and 10 groups of 30 articles for fine-tuning. We examined the variance between the groups and found that it was minimal, validating our results further.

4.2 Models

We investigated several models for testing, which included Sam Shleifer’s DistillBART [5], Pegasus [8], T5 [3]. Due to memory limitations, we were not able to fine-tune Pegasus or Facebook’s BART model, leaving T5 and DistillBART.

4.3 Training

For training, we used the huggingface transformer library to set up our models and feed them data. Each training example consisted of a news article and its summary. Training DistillBART involved 4 epochs of training on 12 examples. These examples were randomly selected from our original dataset of choice. In order to train T5, we used a larger training set of 1720 examples for 10 epochs. We did not use dropout. The weight decay for our training procedure was 0.01, and 0.00005 was our training rate. In addition, we used 500 warm-up steps as recommended by HuggingFace documentation.

4.4 Testing

Testing involved taking an average GRUEN score per summary among each group of a list of 20-100 articles for each article type we considered. The GRUEN metric developed by Zhu and Bhat, examines the linguistic quality of the generated text. The GRUEN metric evaluates the Grammar, non-Redundancy, focUs structure, and coherENce, and was an important measure as it aims to echo what a human would do if they looked at the generated summaries and tried to score it on readability.

Summary	'NTPC shares have risen 13 rowing on its. the government has said it will use the money to feed the.'	'NTPC shares have risen 13 percent. The government has said it will use the money to feed BERT some training examples.'
GRUEN Scores	0.332	0.819

Table 1: Sample GRUEN Scores

4.5 GRUEN as a Proxy for Readability

It is important to establish the GRUEN metric as a proxy for readability, as our results make use of the GRUEN score to establish whether or not our methods work. Refer to the summary examples in Table 1; the leftmost summary has many grammatical errors and poor coherency, whereas the rightmost summary has good coheerency and is much more readable. From the GRUEN scores shown below the summaries, we can see that the GRUEN metric can approximate a summary's readability.

5 Results and Discussion

5.1 Model Results

Model	Business	Entertainment	Politics	Sports	Tech
Baseline DistillBART	0.757	0.758	0.746	0.759	0.749
Fine-Tuned DistillBART	0.773	0.859	0.801	0.840	0.838
Baseline T5	0.756	0.742	0.702	0.717	0.729
Fine-Tuned T5	0.734	0.722	0.711	0.722	0.692

Table 2: GRUEN Scores Across Categories for Different Model Configurations

As a baseline, we evaluated the DistillBART model with a set of 500 articles, 100 for each article type. We averaged the GRUEN score for each article and displayed it in table 1. From this test, the model produced summaries with consistent levels of readability across all 5 article categories with scores ranging from 0.749 for the technology category on the low end, and 0.759 for Sports on the high end. The fine-tuned DistillBART model gauged how much of a performance increase can occur with some fine-tuning. We found that there is an increase in GRUEN metric after fine-tuning, which suggests that with fine-tuning on readable summaries, model outputs will also increase in readability. This improvement was important since it revealed the baseline for an extractive summarization model when compared to the other masked fine tuned models.

In addition to DistillBART, we evaluated T5's GRUEN scores before and after fine-tuning to see if there was any impact. On average, T5 has similar scores to Baseline DistillBART for the business and entertainment categories; the model produces lower GRUEN scores for other categories. This was surprising to us since T5 is an abstractive summarization model, a class of summarization models that prioritize revision and clarity over correctness (like extractive summarization models do). In order to improve this score, we fine-tuned the model with a dataset masked using the parts of speech to determine the probability of the word being masked. For each word the probability of masking was the probability of it occurring in the summary, for example, if nouns made up 8

6 Conclusion

Given our results, we arrive at the conclusion that readability can affect summarization, and masking words according to the probability distribution can have a positive impact depending on the genre of text. Some limitations to our findings arise in training time, and dataset choice, as we do not have access to large-scale private training computers and can only fine-tune models for at most 10 epochs. Additionally, we did not have the resources to experiment further with different masking methodologies, such as implementing random masking at different percentages, calculating the masking probability differently, or using a mask-filled model to determine the least important words to mask. Given these shortcomings, we were able to discover evidence to support the idea that masking does have an effect on the performance of summarization models, and it can be used as a hyperparameter during the fine-tuning stage.

7 Citations

References

[1] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[2] Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[4] Qian Ruan, Malte Ostendorff, and Georg Rehm. Histruct+: Improving extractive text summarization with hierarchical structure information. *arXiv preprint arXiv:2203.09629*, 2022.

[5] Sam Shleifer. https://huggingface.co/sshleifer/distilbart-cnn-12-6.

[6] Dehao Tao, Yingzhu Xiong, Jin He, Yongfeng Huang, et al. An unsupervised extractive summarization method based on multi-round computation. *arXiv preprint arXiv:2112.03203*, 2021.

[7] Changchang Zeng and Shaobo Li. Analyzing the effect of masking length distribution of mlm: An evaluation framework and case study on chinese mrc datasets. *Wireless Communications and Mobile Computing*, 2021, 2021.

[8] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[9] Wanzheng Zhu and Suma Bhat. Gruen for evaluating linguistic quality of generated text. *arXiv preprint arXiv:2010.02498*, 2020.