Extractive Summarization of Financial News Articles Using an LSTM Model and FinRouge Scoring

Arda Mark Sozer BA , Jacob Sycoff

University of California, Berkeley

asozer@berkeley.edu, jsycoff@berkeley.edu

April 10, 2021

Abstract

Extractive summarization methodology is further extended with a baseline BERT model and an encoder-decoder LSTM model using a 306,242 length corpus of finance-related articles. We use Recall-Oriented Understudy for Gisting Evaluation (ROUGE) to evaluate co-occurrences, and develop a further extended scoring methodology for this specific task (which we are naming FinRouge). In lieu of using a pre-trained, general-purpose model, we develop an encoder-decoder LSTM model from scratch to make use of long short-term memory characteristics while training solely on the task-specific corpus.

1. INTRODUCTION

Our goal for this work is three fold;

- Utilize a state-of-the-art extractive summarization but general-purpose model as a baseline to see how well it performs the task for finance-specific articles
- Improve the ROUGE summarization scoring metric to be more suitable for a specific-purpose task
- Develop a fine-tuned LSTM model that is trained on a task-specific corpus to better summarize news articles and build on the model at baseline

1.1 MOTIVATION

It is often far too time consuming and unreasonable to read the entirety of all news articles, especially for retail investors who need to make well-informed decisions about their investments. We are motivated by the ability for someone in the world of finance to not need to read the entirety of news articles, but rather read a much shorter summary that captures the key events and entities of an article. It is important, though, for all parts of a summary to come from the original article so as to not alter the meaning of a document (i.e. using extractive over abstractive summarization).

2. BACKGROUND

Document summarizers do exist, but few speak the language of economics or finance. In an effort to continue the general NLP task of document summarization, we attempt to fine tune a LSTM architecture to train its weights using financial articles. Existing work on this topic was introduced by FinBERT[1], which uses domain training on a

[1] "Medium." *Medium*, https://medium.com/prosus-ai-tech-blog/finbert-financial-sentiment-analysis-with-bert-b277a3607101. Accessed 4 Oct. 2021.

financial article corpus and later fine-tunes the model for sentiment analysis. We utilize this paper as our motivation for our baseline, using a pre-trained extractive summarizer using BERT. Later, we attempt to provide an alternative approach to a similar problem using an LSTM model instead.

The LSTM model is an ideal choice for several reasons, as will be later discussed. One of these is the relatively short amount of time and few resources that LSTMs take to train from scratch.[2] Because our goal is to build a bespoke model for use in the realm of finance, the LSTM is unique in its high level customizability as compared to its cost. This does not, however, come at the expense of performance.

Our novel approach is one of the first of its kind to train an LSTM for extractive summarization of finance/business related articles, and furthermore, for the production of accurate headlines within this realm. Several other researchers employ LSTMs for summarization, but generally with very different overall model architectures. One example is "From Neural Sentence Summarization to Headline Generation:A Coarse-to-Fine Approach", where Tan et al use a LSTM encoder - decoder for the purpose of abstractive headline generation[3].

3. METHODS

In this section we review the baseline model and the LSTM model we use to produce extractive summarizations for financial articles. We further discuss the tradeoffs of using either model, and explain how we will measure the accuracy of our results.

3.1 DATA

To develop towards our goal, we utilize the "US Financial News Articles[4]" project hosted on Kaggle. It consists of 306,242 news articles, importantly containing the text and title of an article, in addition to other metadata such as the language and entities. The articles are a collection that was scraped between January and May 2018 from the websites of companies that are synonymous with finance, including CNBC and Bloomberg. Through EDA, we find that on average each article text is approximately 440 words long, which would prove to be useful information later when deciding on the models we ultimately decide to use.

During preprocessing, we manually make sure to filter only English articles, eliminate false data (i.e. articles consisting of only numbers), lowercase all words, remove extra whitespaces, and transform contractions to standard words. When training a given model, we use the title of each article as the label and its full text as the input. With limited availability of financial article corpora containing man-made summarizations, we decide that the title of an article is a suitable alternative to train the data on, especially because it is the de facto summarization that most people rely on for their information. Our concern about the growing presence of "clickbait" informs our efforts rather than hindering them.

[2] Datasci-w266. "Datasci-W266/2021-Spring-Main." *GitHub*, 6 Mar. 2021, https://github.com/datasci-w266/2021-spring-main.

[3] Tan, et al. Peking University, "From Neural Sentence Summarization to Headline Generation:A Coarse-to-Fine Approach." https://www.ijcai.org/Proceedings/2017/0574.pdf

[4] "US Financial News Articles." *Kaggle*, https://kaggle.com/jeet2016/us-financial-news-articles. Accessed 4 Oct. 2021.

3.1 BASELINE MODEL

The baseline model we use is the pre-trained "BERT Extractive Summarizer[5]," a model that uses HuggingFace Transformers to run extractive summarizations. This model works by "first embedding the sentences, then running a clustering algorithm, finding the sentences that are closest to the cluster's centroids."

We chose this model because we wanted to test our fine-tuned alternative model against the general-purpose current state-of-the-art model. While BERT has the added advantage of not employing any Recurrent Networks, it is not trained on a specific purpose task.

3.1.1 BASELINE MODEL RESULTS & DISCUSSION

Upon observation, the approach we use at baseline does not produce totally favorable results. We find that the summarizations produced do not extract only the most important parts of the document. In fact, we find that approximately 12% of the produced summarizations are identical to their input counterparts. One sample input text to the model is, "March 27(Reuters) - AU Optronics Corp Says it plans to pay cash dividend of $1.2/share for 2017." The resulting summarization of this text is identical to the input, and it does not filter out unnecessary information contained within the text.

To more formally evaluate the summarizations produced at baseline, we use Recall-Oriented Understudy for Gisting Evaluation (ROUGE). More specifically, we use the N-gram Co-Occurrence Statistics (ROUGE_n) [Appendix Figure 1] to compare the similarities of unigram and bigram across summarizations. We find that the unigram recall of our baseline model is equal to 0.464, the bigram recall is 0.459. While we also calculate the precision scores, we decide to focus on recall as it provides a more precise measure of the Co-Occurrences in summaries using the vocabulary of the reference text instead of that of the candidate text.

3.1.1 MODIFICATIONS TO ROUGE

ROUGE scores are the go-to scoring metric for summarization and text classification tasks. Yet, for purposes that are more specific, there are likely a collection of words and phrases that should be awarded higher for being included in the output when compared to less field-specific words that are also in the source text.

We alter PyPi's Rouge package[6] source code to "fine-tune" the metric, altering it to be better suited for finance-specific vocabulary. Our approach to this task was the following:

- Given a word in a pre-set collection of finance-related terms (such as 'corporation,' 'stock,' or 'economy')
    - If that word exists in the reference text, then calculate the rate that it also appears in the generated hypothesis (call this value X).
    - Add this to the original ROUGE_n score (call this Y) of the given hypothesis to evaluate the new ROUGE_n score of X + Y.

The added benefit of a scoring system like this is that it does not change the ROUGE

[5] "Bert-Extractive-Summarizer." *PyPI*, 7 Mar. 2021, https://pypi.org/project/bert-extractive-summarizer/.

[6] "Rouge." *PyPI*, 11 Mar. 2020, https://pypi.org/project/rouge/.

score for summarizations, which should not contain words that are task-specific. As a tradeoff, however, and a possibility looking ahead, the scoring metric is now in the range [0, ∞). While that is altered from the original range [0, 1], it does not take away the ability for one to compare two scores and decipher which one was summarized more effectively.

We use this scoring metric on our baseline approach. Our new unigram recall score is 0.860 (up from 0.464), while that of bigram recall is 0.854 (up from 0.459). [Appendix Figure 4]

3.2 TRAINING AN LSTM MODEL

Inspired by the motivation to train an appropriate model with the text and headlines of financial articles, we develop a fine-tuned encoder-decoder LSTM model to produce extractive summarizations [Appendix Figure 2]. The choice of this approach was two-fold:

- Encapsulate context into the summarizations outputted by the model using context vectors from the encoder (i.e. taking advantage of the Long Short-Term Memory characteristics).
- Supply the entire training corpus based solely on finance-related text, instead of using a pre-trained general purpose model with adjusted weights. This way, we hope the model will be very task-specific and reliable for decisions relating to finance.

We train the model on 80% of the corpus alongside the articles' titles. We use a batch size of 10 and 5 epochs during training. Similar to the approach taken for the baseline model, we apply the same preprocessing on the data.

3.2.1 MODEL RESULTS & DISCUSSION

Using this approach, we produce an accuracy of 0.745 when comparing co-occurrence between hypothesis and reference texts. Furthermore, we use Categorical Cross-Entropy Loss to evaluate the loss for the multi-class classification task. Using this metric, we derive a loss value of 0.950 [Appendix Figure 3].

However, there may exist a drawback to our approach of training using titles as labels. In recent years, news headlines have become less of a summary and more of a 'clickbait', meant to catch the attention of readers. This trend often has a tradeoff of foregoing the basic content within a document. For example, a recent news article in the Wall Street Journal has the title, "Boeing 737 MAX Faces Fresh Inspection[7]." While interesting, these kinds of headlines do not provide the basic information that is to be taken away if one were to read the entire text. More formally, these types of headlines are concise but do not meet the other two basic requirements for an effective summarization, including accuracy and objectivity.

4 CONCLUSION

We were motivated by the prospect to shorten the time it takes for investors to stay updated about current affairs, and to help them more efficiently select articles to read based on their contents. We utilized an extractive summarizer using BERT to see

[7] Cameron, Alison. "Boeing 737 MAX Faces Fresh Inspections." *WSJ*, 10 Apr. 2021, https://www.wsj.com/articles/boeing-flags-potential-737-max-electrical-issue-on-specific-jets-11617972277.

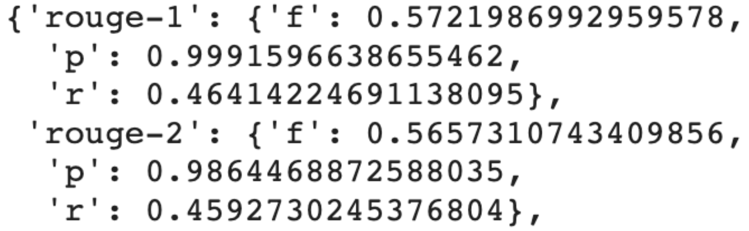
how a general-purpose state-of-the-art model behaves in a more task-specific role. We then modify the ROUGE to improve its scores for the objective in this paper, as well as other more niche NLP tasks (we name this scoring metric as FinRouge). Finally, we attempt to develop an LSTM model that is entirely trained on financial articles to produce extractive summarizations for financial news pieces.

4.1 EXTENSIONS

Accurate, concise, and objective summaries of articles are not readily available. As a result, we propose an effort to gather effective human written summaries, which are the gold standard for labels in tasks such as extractive summarization. These could be generated manually using services like Amazon Mechanical Turk, where participants are compensated for labelling articles with appropriate one-sentence headlines. This data could then be used to train a model to predict summaries that are less of a "clickbait," but instead an accurate summary of the content.

APPENDIX

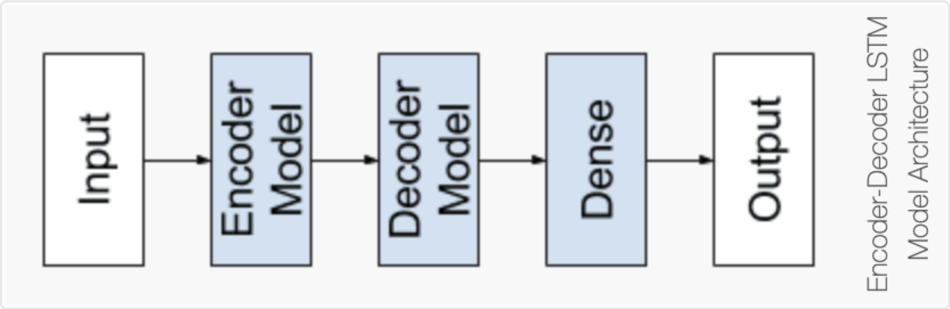
Figure 1:



```
{'rouge-1': {'f': 0.5721986992959578,
   'p': 0.9991596638655462,
   'r': 0.46414224691138095},
  'rouge-2': {'f': 0.5657310743409856,
   'p': 0.9864468872588035,
   'r': 0.4592730245376804},
```

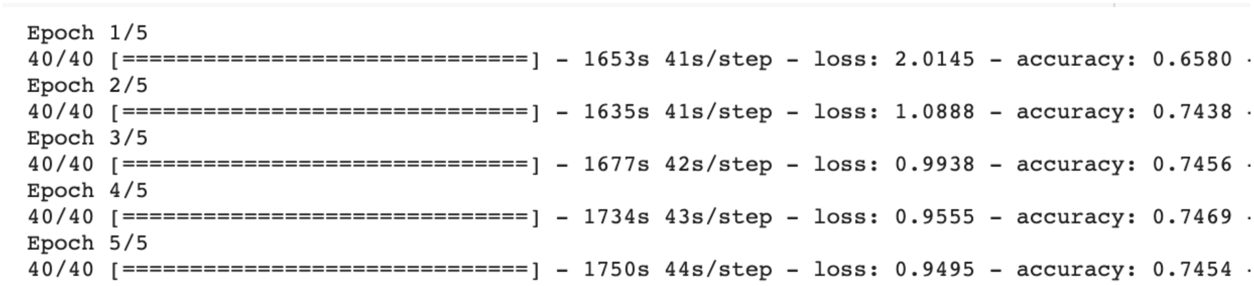
Precision, Recall, and F1 scores for unigram and bigram Co-Occurrences at Baseline

Figure 2:



The sequence-to-sequence LSTM model emulated

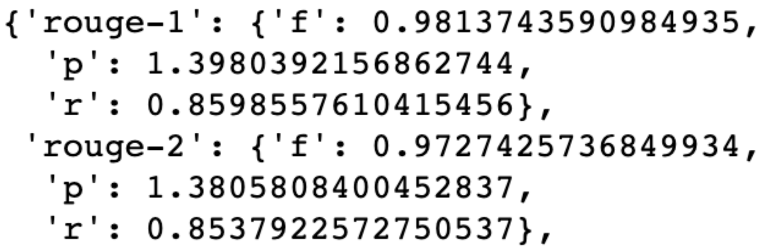
<u>*Figure 3:*</u>



```
Epoch 1/5
40/40 [==============================] - 1653s 41s/step - loss: 2.0145 - accuracy: 0.6580 ·
Epoch 2/5
40/40 [==============================] - 1635s 41s/step - loss: 1.0888 - accuracy: 0.7438 ·
Epoch 3/5
40/40 [==============================] - 1677s 42s/step - loss: 0.9938 - accuracy: 0.7456 ·
Epoch 4/5
40/40 [==============================] - 1734s 43s/step - loss: 0.9555 - accuracy: 0.7469 ·
Epoch 5/5
40/40 [==============================] - 1750s 44s/step - loss: 0.9495 - accuracy: 0.7454 ·
```

Model output accuracy and loss

<u>*Figure 4:*</u>



```
{'rouge-1': {'f': 0.9813743590984935,
  'p': 1.3980392156862744,
  'r': 0.8598557610415456},
 'rouge-2': {'f': 0.9727425736849934,
  'p': 1.3805808400452837,
  'r': 0.8537922572750537},
```

Precision, Recall, and F1 scores for unigram and bigram Co-Occurrences of Baseline Using Modified ROUGE scoring