CS224U Project: Text Summarization for News

Priscilla Zhao
Stanford University
puzhao@stanford.edu

Yuyu Lin
Stanford University
linyuyu@stanford.edu

Shanduojiao Jiang
Stanford University
sj99@stanford.edu

Abstract

Text summarization tries to generate shorter text without removing the actual context of information. In this project, we investigated news text summarization by exploring a baseline model which is an extractive model that selects words, phrases or sentences from the document but does not require neural networks, then we implemented an abstractive Seq2Seq model (Recurrent Neural Network model) and found the performance of Seq2Seq model is generally better than the frequency-based extractive model(baseline model). Beyond that, we also leveraged different metrics to evaluate the quality of automated news text summarization and noticed that the two models each have their own merits quantitatively and qualitatively.

1 Introduction

The past two decades has seen a huge information explosion - the rapid increase in the amount of published information or data and the effects of this abundance. From visual data such as pictures and videos posted on social media, to textual data such as transcripts, articles, or scholarly papers - as the amount of available data grows, the problem of managing the information becomes more difficult. One of the most important tasks of managing and making sense of the information is text summarization.

Automatic text summarizing is the process of compressing a document while preserving key information content and meaning. In other words, it is the problem of creating a short, accurate, and fluent summary of a longer text document. As one of the most abundant sources of unstructured data, text data usually consists of documents which can represent worlds, sentences or even paragraphs of free flowing text. To be able to develop algorithms that can automatically shorten longer text data and deliver accurate summaries has many benefits, such as reducing reading time, accelerating the process of researching for information, and increasing the amount of information that can fit in an area.

However, summarizing is a difficult skill to master. In order to automatically summarize a text document, the machine must be able to understand the document, separate the main ideas from the details, and reduce a large quantity of information to the most important main ideas.

In this project, we took a deep dive into the field of text summarization and specifically focused on news text. We compared and analyzed the difference between a frequency-based model and a Seq2Seq model quantitatively and qualitatively using ROUGE variants as our evaluation metrics. As expected, while the neural network-based Seq2Seq model performed better than the conventional frequency-based in news summarization, the frequency-based model has its merits. Our project would enrich the context of natural language understanding(NLU) by analyzing the recent news dataset and suggesting future researchers who are interested in studying news summarization try using a combination of an abstractive model and an extractive model.

2 Related Work

2.1 Prior Methods

In general, text summarization is achieved by the following techniques:

- Extractive summarization is a method of selecting a subset of words, phrases or sentences from the input document to form a summary.
- Abstractive summarization involves creating sentences that summarize the content and capture key ideas and elements of the source text,

often with significant changes and paraphrases to the original text.

- Some work also uses a hybrid approach where an extractor first selects salient sentences from the input. Then, an abstractive summarizer rewrites extracted sentences into a final summary.

Earlier approaches tend to use extractive methods that apply statistical computations to generate the summary. However, the difference lies in the statistical methods:

- Lin and Hovy's method(Lin and Hovy, 2000) first finds the topic signature, or key phrases, using likelihood ratio λ, and then extracts summary sentences based on these topic signatures.

- Convoy et al.'s method(Conroy and O'leary, 2001) uses the Hidden Markov Model to calculate the likelihood that each sentence should be contained in the summary.

- Steinberger and Jezek's method (Ozsoy et al., 2011) uses Latent Semantic Analysis on a term by sentence matrix to generate the summary.

Among the recent papers in extractive summarization, abstractive summarization and hybrid methods, we noticed they tackled different sub-tasks, for example, Cheng and Lapata's (Cheng and Lapata, 2016) work focused on shorter document; Chopra et al.'s (Chopra et al., 2016) work focused on sentence, and Subramanian et al.'s (Subramanian et al., 2019) method is for long documents of more than thousands' of words.
There are some interesting similarities and differences in their models:

- Problem abstraction:
  - They all have the same way of problem formulation: when doing extraction, they predict the probability/ label of if each sentence will be in the summary. An interesting difference is that Cheng and Lapata's(Cheng and Lapata, 2016) work include a CNN layer to map the features while others' work only includes RNN layers.
  - When doing abstraction, they all aim at predicting the next word in the summary; while Cheng and Lapata's(Cheng and Lapata, 2016) word extractor and Chopra et al.'s (Chopra et al., 2016) RAS based on conditional probability; Subramanian et al.'s(Subramanian et al., 2019) method is to train a unconditional transformer language model.

- Use of attention mechanism:
  - Their methods all include attentive mechanism. However, Cheng and Lapata (Cheng and Lapata, 2016) apply attention weights to extract salient sentences; Chopra et al.(Chopra et al., 2016) use attentive encoder; while Subramanian et al.(Subramanian et al., 2019) use attentive decoder.

- The function and performance of LSTM:
  - They all use LSTMs. Cheng and Lapata's (Cheng and Lapata, 2016) method use LSTM activation unit to reduce the vanishing gradient problem for long sequences.
  - Chopra et al.'s (Chopra et al., 2016) tested both Elman RNN and LSTM; in their evaluation, they found that the RAS-LSTM performs slightly worse than RAS-Elman, most likely due to overfitting.
  - Subramanian et al.'s (Subramanian et al., 2019) used lots of LSTM: they use bidirectional LSTMs when doing token-level encoding; they use another LSTM to do the next sentence-level encoding.

2.2 Evaluation Measures for Text Summaries

Evaluating the quality of a summary is crucial in text summarization research but a puzzle task till now. The question here is what are the appropriate evaluation criteria for different types of automatic summaries which may differ in the content or in the context or in the way of presenting or etc. The following section will first review the determinations for summary evaluation measures and then review the widely used metrics, ROUGE.

2.2.1 Determinations of Summary Quality

Steinberger et al., (Steinberger et al., 2009) valued the measures from the following perspective: text quality measures, co-selection measures, content-based measures, and task-based measures.

- Text Quality measures text quality in grammaticality, non-redundancy, reference clarity, coherence and structure.

- Co-Selection's main evaluation metrics are precision (P), recall (R) and F-score. Precision (P) is the number of sentences showing up in both system and ideal summaries divided by the number of sentences in the system summary. Recall (R) is the number of sentences showing up in both system and ideal summaries divided by the number of sentences in the ideal summary. And F-score could be computed as $F = \frac{2PR}{P+R}$ or a complex form $F = \frac{(\beta^2+1)PR}{\beta^2 PR}$
Beyond that, it also measures relative utility which is defined as $F = \frac{\sum_{j=1}^{n} \delta_j \sum_{i=1}^{N} u_{ij}}{\sum_{j=1}^{n} \epsilon_j \sum_{i=1}^{N} u_{ij}}$. It tries to address the problem P & R brings which is human judges hold a different opinion of the top important sentences in the document.

- Content-based addresses the problem that two different sentences could represent the same meanings and for the similarity measures it includes: Cosine Similarity; Unit Overlap; Longest Common Subsequence; N-gram Co-occurrence Statistics and Pyramids.

2.2.2 ROUGE

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is a set of metrics evaluating the quality of automatic text summarization by comparing the generated summaries to other human-made summaries. Steinberger et al's (Steinberger et al., 2009) found ROUGE measures demonstrate the best performance when measuring the similarity of summaries with human abstracts and automatic summaries. ROUGE determines the quality of automated summaries by comparing overlapping units such as n-grams, word sequences, and word pairs with human-written summaries. However, NG and Abrecht (Ng and Abrecht, 2015) announced that ROUGE does not suit for evaluating abstractive summarization and summaries with substantial paraphrasing. Rouge also does not have a good performance in opinion summaries (Tay et al., 2019) and meeting summaries (Liu and Liu, 2009). Based on lit review, ROUGE might be sensitive to the summarization style. Our study is interested in how ROUGE would perform in news text summarization's evaluation.

3 Data

We used NEWS SUMMARY[1] to train our abstractive model. We believe that abstractive models are more suitable for summarizing news articles, because news covers a wider range of topics and vocabulary than academic articles, therefore, a good news summary requires more words or sequences that are not in the original document.

NEWS SUMMARY dataset contains summarized news from Inshorts and news articles from Hindu, Indian times and Guardian. We believe that the summarized news from Inshorts are more objective with higher quality. Most importantly, summaries and articles have different authors, so they use different vocabulary for the same meaning, which is probably more applicable for training abstractive model. We trained our model with this dataset for a balance of efficiency and performance.

3.1 Dataset statistics

NEWS SUMMARY dataset consists of 4514 examples (Table 1). We split this dataset into training (80%), development (10%), and test (10%) sets.

Dataset Size	4514 articles
Training Set Size	/
Mean Article Length	344.0 words
Mean Summary Length	59.0 words
Article Vocabulary Size	71516 words
Summary Vocabulary Size	24972 words
Total Vocabulary Size	75209 words
Occurring 10+ Times	12252 words

Table 1: NEWS SUMMARY Dataset statistics

3.2 Preprocessing

Our preprocessing method includes:

- Filtering out the non-alphanumeric characters (except space, full stop, question mark, exclamation mark)

[1] https://github.com/sunnysai12345/News$_{summary}$

- Stripping https and slashes and replacing any url as such https://abc.xyz.net/browse/sdf-5327 to abc.xyz.net
- Removing multiple spaces
- Removing any single charecters hanging between 2 spaces
- Transforming the characters into lower case
- Splitting the text into sentences
- Tokenization: splitting the sentences into words

4 Model

In this paper, we are interested in comparing a frequency-based extractive model (baseline model) with an abstractive Seq2Seq model on the task of text summarization for news data. Based on our understanding, baseline model should be lightweight, fast, and has a relatively good result. The frequency-based extractive model is easy to use and could generate reasonable results for extractive summarization (Sakhadeo and Srivastava, 2018), which perfectly aligns with our expectation of a baseline model. Recurrent neural network (RNN)-based sequence to sequence (seq2seq) model is a model that takes a sequence of items (words, letters, time series, etc) and outputs another sequence of items, which has been successfully applied to several natural language processing (NLP) tasks.

4.1 Baseline Model

The baseline model generates the summary based on the frequency of words in a sentence, which was used to calculate the scores for each sentence within the input text. This model have the following steps:

- Step 1: Remove stop words and tokenize each input text.
- Step 2: Create a frequency table of words for each processed input text. In this step, a dictionary was used to keep track of how many times each word appeared in the feedback after removing the stop words. The intuition behind this is that we can use the dictionary over every sentence to know which sentences have the most relevant content in the overall text.
- Step 3: Assign score to each sentence depending on the words it contains using the frequency table mentioned in step 2.
- Step 4: Assign a certain score to compare the sentences and generate the summary. In this step, we chose to use the average score of a sentence to represent the "value" of a sentence. If the "value" of a sentence is above a threshold, which was set as 1.2 * average score of all the sentences in an input text, we will then add this sentence to the summary.

4.2 Seq2Seq Model

Figure 1 shows the architecture of the Seq2Seq model that we use for text summarization. A Seq2Seq model is a model that takes a sequence of items (words, letters, time series, etc) and outputs another sequence of items. In our task, the input is a series of words, and the output is the summarized series of words. Such a model usually consists of an encoder module and a decoder module. In our implementation, we use 4 LSTM layers for the encoder, and 3 LSTM layers for the decoder.

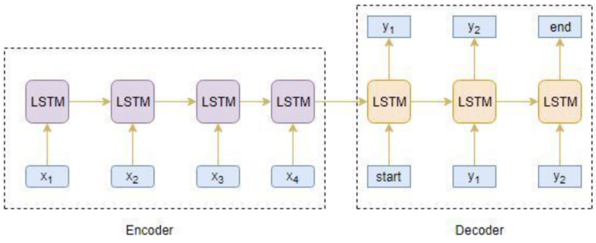

Figure 1: Seq2Seq Model Architecture

LSTM is a special class of RNN (recurrent neural networks) for its ability to learn long-term dependencies. Standard RNNs have the form of a chain of repeating modules of neural network, which could be a relatively simple structure such as a single tanh layer. LSTM is different from standard RNNs for its intricate design of the repeating modules. Figure 2 shows the module details. Specifically, LSTM uses a series of "gates" which control how the information in a sequence of data comes into, is stored in and leaves the network. There are three gates in a typical LSTM: forget gate, input gate, and output gate. Forget gate decides which bits of the cell state are useful given both the previous hidden state and new input data. Input gate determines what new information should be added to the network's long-term memory given the previous hidden state and new input data. Output gate decides the new hidden state given the previous hidden state and the new input data. These gates can be thought of as filters and are each their own neural network.

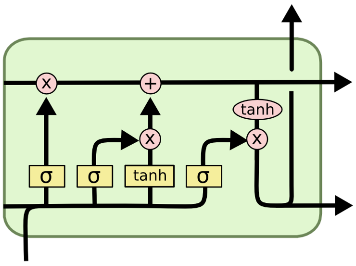

Figure 2: LSTM Module Detail

Having explained the details of LSTM layers and our Seq2Seq model structure, the following steps explains how we use the model for text summarization:

- Step 1: We first preprocess the data by tokenizing, analyzing rare words, and adding _START_ and _END_ tokens.
- Step 2: Then we fit the model with the preprocessed data.
- Step 3: We build the dictionary to convert the index to word for target and source vocabulary
- Step 4: We define the functions to convert an integer sequence to a word sequence for summaries.

5 Methods

5.1 Metrics

For this project, we use ROUGE variants as our evaluation metrics. Specifically, we use:

- ROUGE-1: refers to the overlap of unigram (each word) between the system and reference summaries.
- ROUGE-2: refers to the overlap of bigrams between the system and reference summaries.
- ROUGE-L: Longest Common Subsequence (LCS) based statistics that take into account sentence level structure similarity naturally and identify longest co-occurring in sequence n-grams automatically.

In general, ROUGE-N is computed using the following equation (Lin, 2004):

$$\frac{\sum_{S\in ref}\sum_{gram_n\in S} Count_{match}(gram_n)}{\sum_{S\in ref}\sum_{gram_n\in S} Count(gram_n)}$$

5.2 Seq2Seq Model Details

For our Seq2Seq model, we use a latent dimension of 300 and embedding dimension of 200. During training, we apply drop out with a probability of 0.4, and use RMSprop as our optimizer. We use sparse categorical crossentropy, which computes the crossentropy loss between the labels and predictions when there are two or more label classes, as our loss function. We trained the model twice: we first trained it with 10 epochs, and then trained it with 50 epochs with early stopping.

6 Results & Analysis

6.1 Baseline Model Results

For the baseline model, due to the limitation of using statistical methods only, not every input text could have a corresponding summary output. Out of the 102915 texts, 37325 of them are empty. This means that there is a 36.27% chance that the model could not provide a summary for an input text, and thus it is not always reliable to use the baseline model for text summarization. Table 2 shows an example output from our baseline model.

Just as shown in this example, though our baseline model can generate relatively decent summaries, it tend to leave out the important information, such as the "90% salary hike".

6.2 Seq2Seq Model Results

Our Seq2Seq model performs significantly better in terms of the capability of producing a summary given an input text - it is able to output a summary given any input text. An example summary is shown in Table 3. Just like in this example, Seq2seq model is able to grasp the important information from the given text succinctly, but it sometimes fails to produce a sentence that is grammatically and syntactically correct.

Figure 3 and 4 show the loss curves for training the Seq2Seq model for 10 epochs, and 50 epochs with early stopping, respectively. For the model trained for 10 epochs, we can see that it still hasn't fully converged yet as we haven't

Original Text	Saurav Kant, an alumnus of upGrad and IIIT-B's PG Program in Machine learning and Artificial Intelligence, was a Sr Systems Engineer at Infosys with almost 5 years of work experience. The program and upGrad's 360-degree career support helped him transition to a Data Scientist at Tech Mahindra with 90% salary hike. upGrad's Online Power Learning has powered 3 lakh+ careers.
Reference Summary	upGrad learner switches to career in ML & Al with 90% salary hike
Predicted Summary	Saurav Kant, an alumnus of upGrad and IIIT-B's PG Program in Machine learning and Artificial Intelligence, was a Sr Systems Engineer at Infosys with almost 5 years of work experience.

Table 2: Baseline Model Output Sample

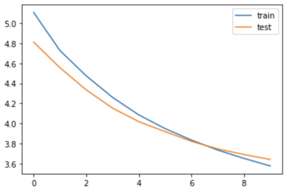

Figure 3: Model Loss(10)

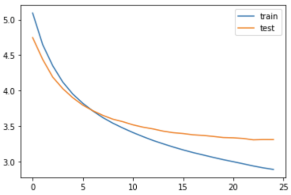

Figure 4: Model Loss(25)

seen a plateau at epoch 10. For the model trained for 50 epochs, which early stopped at epoch 25, we can see that the model started to overfit the training data as the training loss keeps going down while the validation loss starts to go back up. In this project, we produce summaries and report the results using the model trained for 10 epochs, which might explain why our model could not generate sentences that are grammatically and syntactically correct sometimes.

6.3 ROUGE Comparisons

We implemented ROUGE-1, ROUGE-2 and ROUGE-L to compare the reference summary and the summary generated using our models. P and R means precision and recall, and F-score is computed as $F = \frac{2PR}{P+R}$. Table 4 shows the ROUGE results for our baseline model outputs, and Table 5 shows the ROUGE results for our Seq2Seq model outputs. We used 9983 validation samples to calculate the ROUGE scores for the Seq2Seq model due to the time the model takes to do real-time inferencing.

Original Text	students of government school in uttar pradesh sambhal were seen washing dishes at in school premises on being approached basic shiksha adhikari virendra pratap singh said yes have also received this complaint from elsewhere we are inquiring and action will be taken against those found guilty
Reference Summary	students seen washing dishes at govt school in up
Predicted Summary	school students fall ill after being raped by up school

Table 3: Seq2Seq Output Sample

We can see that the Seq2Seq model performs better using ROUGE-1 and ROUGE-l as evaluation metrics, while the frequency-based model performs better using ROUGE-2 as the evaluation metric. This is probably because the frequency-based model extracts words and phrases from the original input text, and is thus more likely to contain target bigrams. Another finding is that the frequency-based model tend to have higher recall than the Seq2Seq model across all ROUGE variants.

	ROUGE-1	ROUGE-2	ROUGE-L
P	0.1760	0.0605	0.1524
R	0.5159	0.2008	0.4489
F	0.2566	0.0905	0.2221

Table 4: ROUGE Results for Baseline Outputs

	ROUGE-1	ROUGE-2	ROUGE-L
P	0.3547	0.0504	0.3448
R	0.3109	0.0454	0.3021
F	0.3279	0.0473	0.3186

Table 5: ROUGE Results for Seq2Seq Outputs

7 Conclusion

In this project, we gained hands-on experience in researching, developing, comparing, and analyzing different models for the task of text summarization on news data. Specifically, we implemented and compared a frequency-based extractive model with a neural network-based abstractive Seq2Seq model. In general, while the Seq2Seq model performs better than the frequency-based model in terms of capturing key information accurately and succinctly, it sometimes lacks the ability to produce grammatically and syntactically correct sentences, and takes much longer to train and inference. In the future, it would be beneficial to train the model longer, or explore other neural network architectures such as Transformer.

Known Project Limitations

For this project, the main limitations are:

1. **Model Training Time:** Due to the unstableness of Google Colab, though we were able to obtain the model loss graph trained for 50 epochs with early stopping, we were not able to save the model in time for analysis, and thus we believe this is one of the reasons why our current Seq2Seq model could not produce high quality sentences.

2. **Dataset:** Due to the limit of our computational resource, we only used NEWS SUMMARY[2] for training and testing. However, it would be beneficial to include more news data such as NEWSROOM (Grusky et al., 2018).

3. **Model Architecture:** Just as mentioned in Related Work section, Transformers could generate relatively good text summarization results. If commputational resource allows, it would be beneficial to explore implementing and training Transformers.

[2] https://github.com/sunnysai12345/News_Summary

References

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *CoRR*, abs/1603.07252.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

John M Conroy and Dianne P O'leary. 2001. Text summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 406–407.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*.

Feifan Liu and Yang Liu. 2009. Exploring correlation between rouge and human evaluation on meeting summaries. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(1):187–196.

Jun-Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for rouge. *arXiv preprint arXiv:1508.06034*.

Makbule Gulcin Ozsoy, Ferda Nur Alpaslan, and Ilyas Cicekli. 2011. Text summarization using latent semantic analysis. *J. Inf. Sci.*, 37(4):405–417.

Archit Sakhadeo and Nisheeth Srivastava. 2018. Effective extractive summarization using frequency-filtered entity relationship graphs. *arXiv preprint arXiv:1810.10419*.

Josef Steinberger et al. 2009. Evaluation measures for text summarization. *Computing and Informatics*, 28(2):251–275.

Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher J. Pal. 2019. On extractive and abstractive neural document summarization with transformer language models. *CoRR*, abs/1909.03186.

Wenyi Tay, Aditya Joshi, Xiuzhen Jenny Zhang, Sarvnaz Karimi, and Stephen Wan. 2019. Red-faced rouge: Examining the suitability of rouge for opinion summary evaluation. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 52–60.