Natural Language Generation to Generate Text Summaries

Faisal Rasheed Khan
University of Maryland Baltimore County
vb02734@umbc.edu

Abstract

Text summarization is used most often nowadays, as it plays a main role in summarizing longer documents into shorter ones for a better understanding of the human. People prefer having shorter versions of the document read compared to longer documents. Text summarization not only benefits humans but also for the digital media to generate efficient content. There has been extensive research going on for the summarization task. In this literature review, we will have a look at the various methods used for text summarization. We will be seeing how these techniques face the challenge of preserving the entire meaning of the document. We will also take a look at how the techniques have evolved over the years.

1 Introduction

In today's world, time is very important for an individual. Many people might have seen lengthy documents, and they just don't read the longer version of the documents. There are a humongous number of available lengthy documents that are useful for humans but due to their time constraints or negligence, they just ignore those. If the lengthy documents are summarized preserving the whole meaning, then it would be helpful, this is where Text summarization comes into the picture.
Before beginning the text summarization, we need to understand what exactly the text summarization is with respect to the Natural Language Processing task.
Text summarization is of two types:

i. Extractive Summarization
ii. Abstractive Summarization.

Extractive summarization is an easier task compared to abstractive summarization. In Extractive summarization, the vocabulary used will be of the source document by maintaining the meaning of the whole document. In Abstractive summarization, the vocabulary used will be of novel words and generate completely new sentences by preserving the meaning of the source document.

The challenges that text summarization faces are sentence repetition, fluency, coherence, out-of-vocabulary words, sentence precise meaning, and meaning-specific content. The recent work in the field of text summarization has overcome the challenges up to an extent, but the work for accuracy is still going on.

2 Literature Review

2.1 Traditional Methods which deal only with Extractive Summarization

2.1.1 Graph-Based Summarization

In the paper Litvak et al., 2008 the authors proposed two solutions for the text summarization task i.e., supervised and unsupervised. Supervised works on the classification algorithms on a summarized collection of documents represented in a graph. The graph represents the syntactic representation of textual web documents. The nodes represent words and edges represent semantic relationships. For unsupervised, they use a rank-based HITS Algorithm. The unsupervised doesn't train on the collection of summaries but it works on the HITS algorithm, the challenge is NP-Hard, so rather than going for convergence they run for one iteration. The dataset used here is the Document Understanding Conference 2002 (DUC, 2002). For the classification task, if the nodes of a graph belong to at least one summary then it is set to 1, else 0. The Supervised task works better than the unsupervised task. The work concludes with

84% accuracy for the classification task and for the unsupervised, it depends on the convergence criteria they set and the accuracy is around 80%. When the training data is not that good then the use of the unsupervised HITS Algorithm is recommended.

The paper Litvak et al., 2008 outcomes might not be useful considering the fact of the model description as it might not present the summaries accurately because many of the sentences/words might get repeated and uniqueness will not be there, extractive also seems to be underperformed.

2.1.2 Maximizing Semantic Volume for Summarization

Next, we look further at different techniques which can be helpful for text summarization. The paper Yogatama et al., 2015 discusses the text summarization approach by maximizing the semantic volume. To maximize the volume a greedy approach is being used which is based on the Gram-Schmidt. The baseliners considered for this technique are Maximal Marginal Relevance (Carbonell and Goldstein, 1998) and Coverage-based summarization. The greedy approach used is also an NP-Hard problem, so they try to optimize by maintaining a constraint for the volume. The datasets used here are TAC-2008 and TAC-2009. The proposed model based on the budget constraint outperforms the baseline model as it considers a good amount of semantic volume for coverage.

Compared to the Litvak et al., 2008 graph model, the model proposed by Yogatama et al., 2015 performs well because of the advantage of the semantic volume. This one captures more extractive information compared to the Graph-based model(Litvak et al., 2008). This approach has a future scope of including more volume by embedding the words by compressing volume further, but it lacks an abstractive nature and can only be used for extractive tasks.

2.1.3 Summarization using dense Embeddings

There is an approach seen in the paper Yogatama et al., 2015, whereas the paper Kobayashi et al., 2015 presents in a different way by describing summarization based on the dense embeddings and here model function is defined by the cosine similarity with an extension of the submodular function defined. Their function is calculated based on the nearest neighbors KL-divergence. The dataset used here is the Opinosis dataset (Ganesan et al., 2010) which contains user reviews compared to the DUC datasets which are formal news articles The results were evaluated on the ROUGE metric, and these results were higher compared to the Yogatama et al., 2015, although the dataset was different, the impact of Kobayashi et al., 2015 is good compared to the former. This technique also lacks the abstractive nature of the summary. We look at the techniques further that can overcome the drawbacks presented before.

2.2 Improvements in the Neural Language Models for Abstractive Summarization

2.2.1 Neural Attention Model

The paper Rush et al., 2015 summarizes the sentence using the neural attention model. It is a neural language model with an encoder and decoder. The encoder is modeled with an attention-based encoder Bahdanau et al. (2014) with a latent soft alignment over the input.

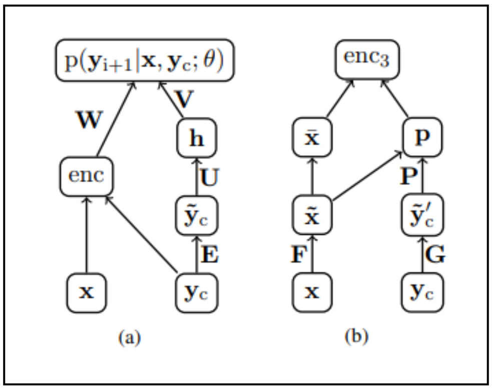

Figure 1: Rush et al., 2015 (a) NNLM decoder with additional encoder. (b) Attention-based encoder (enc3).

The model works on a conditional probability which maximizes the likelihood.

The neural language feed-forward network is given by(Rush et al., 2015):

$$p(\text{nextword}|\text{context},x;\theta) \propto \exp(Vh+W_{enc}(x,y_c)), \quad (1)$$

where h uses the tanh activation function

The model uses minimization of the negative log-likelihood. For training data, we use both the DUC-2004 and annotated Gigaword dataset (Graff et al., 2003; Napoles et al., 2012). The model outperforms the baseline models for both the datasets and the perplexity metric used for the Gigaword dataset (Graff et al., 2003; Napoles et al., 2012) and the proposed model Attention-Based (ABS) is low, which is very good compared to the same model which does not have the encoder.

Overall, the presented model in the paper Rush et al., 2015 is good from the perspective of sentence summarization which is abstractive. This model also lacks the proper level of abstraction when applied to the documents of many sentences, as this model just gives a summarization of sentences and if the sentence is repeated in a different way, then it repeats the sentence with the abstractive summary of the sentence.

2.2.2 Hierarchical Document Encoding Using Attention-Based Extractor

The task of achieving the abstract summarization seems to be a tough task. One of the challenging aspects of extractive summarization is the training data. Let's have a look at other papers on how they are going to resolve this issue, where the previous papers' proposed models are repeating sentence summarizations and are less abstractive.

The paper Cheng et al., 2016 seems to be a proper extension of the work done previously by Rush et al., 2015. Compared to the former here many sentences from a document are considered rather than just performing abstractive summarization of a sentence at a time. Here the models used are Hierarchical Document Encoder and Attention-Based Extractor. The encoder's job is to maintain the meaningful representation of documents based on words and sentences. Neural networks are used for summarization tasks. There are two extractions: sentence and word. The sentence extraction objective is to maximize the log-likelihood and the importance of the word extractor is that it ensures the long extractive summaries are not taken.
The training data used here is DailyMail and to validate both DailyMail and DUC-2002 are used, and the metrics used for evaluation are ROUGE and Human Evaluation. The two datasets are created from the DailyMail for the sentence extractor and the word extractor.
Some rules are taken into account for sentence extractor i.e. 1 if the document matches highlights, else 0. The lexical overlap is taken for word extractor between highlights and the news article. The out-of-vocabulary words are handled by replacing similar kinds of semantic words. The sentences are passed to a convolutional neural network(CNN) with max-pooling which are document-level representations and then these are passed to the Recurrent Neural Networks (RNN) which use the Long Short-Term Memory (LSTM) activation function to overcome the vanishing gradient problem. The word extraction acts as a generation task, and it generates the summary based on all the conditions applied to the layers. The model outperformed the Rush et al., 2015 model in both DailyMail and DUC-2002 datasets.

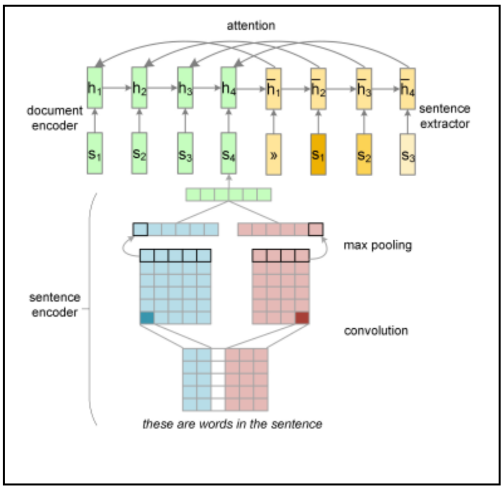

Figure 2: Cheng et al., 2016 A recurrent convolutional document reader with a neural sentence extractor

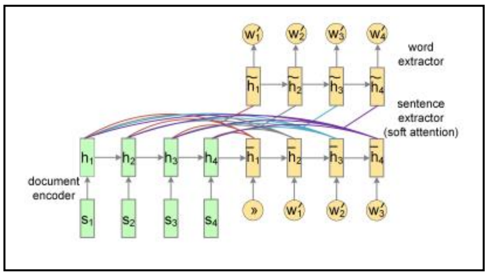

Figure 3: Cheng et al., 2016 Neural attention mechanism for word extraction.

Although the model Cheng et al., 2016 overcomes the Rush et al., 2015 model, still, the model is still not very abstractive as the redundancy is there because it is just abstractive for a few sentences and the repeated sentences occur which are not known to the model. So, this technique in the paper Cheng et al., 2016 should be studied further, and necessary updates such that the abstractive nature of the repetitive sentences is handled.

2.3 Copy Mechanisms

2.3.1 CopyNet using Sequence-to-Sequence Model

Some of the techniques fail to represent the actual summarization and coherency and they present with the unnecessary not related words which form a sentence. To handle this, the paper Gu et al., 2016 introduces the copy mechanism in Sequence-to-Sequence Learning. This technique presents the appropriate sentence by copying longer sequence words such that at least accurate information is summarized. The Sequence-to Sequence is an encoder-decoder model and it outperforms the RNN-based model.

This model also handles out-of-vocabulary words as it searches the semantic

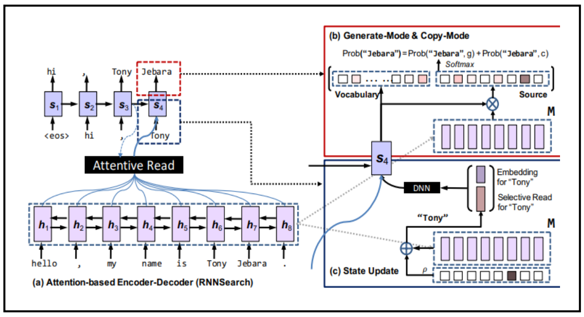

Figure 3: Gu et al., 2016 CopyNet Architecture.

unknown word sequence. The dataset used is the LCSTS dataset (Hu et al., 2015) which is gathered from news media on Sina Weibo.

This model outperforms the RNN-based models, and it is obvious because it doesn't deviate from the actual content which summarizes i.e., the exact important words are preserved. Although it performs well by handling this, it lacks abstractive text and the sentences can be repeated because of the copying nature.

2.3.2 Attention-Based Encoder Decoder Sequence-to-Sequence RNN

Let's take a look at the techniques improved from the above-discussed ones, here in the paper Nallapati et al., 2016 where the authors use Sequence-to-Sequence RNN with extra novel models. They consider Attentional Encoder-Decoder RNN on two different corpora. They have also proposed a new dataset consisting of multisentence summaries. The inclusion of different novel models for the Seq2Seq RNN are Large Vocabulary Trick (LVT) (Jean et al., 2014), Feature-rich Encoder, and Switch Generator-Pointer. Feature-rich Encoder has Linguistic features such as POS, NER, TF, and IDF. With the help of the Switch Generator, out-of-vocabulary words are handled as the pointer keeps track of the source document, the G – softmax layer produces words, and the P – Pointer network is activated to copy from the source. The dataset used here is

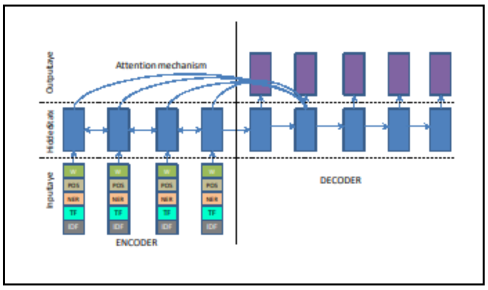

Figure 4: Nallapati et al., 2016 Feature-rich-encode

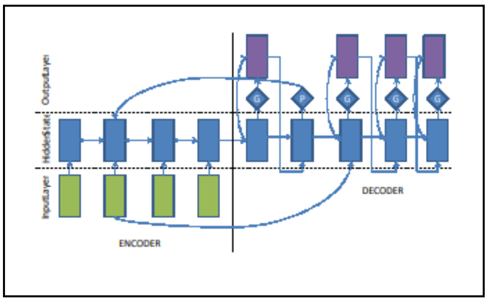

Figure 5: Nallapati et al., 2016 Switching generator/pointer model

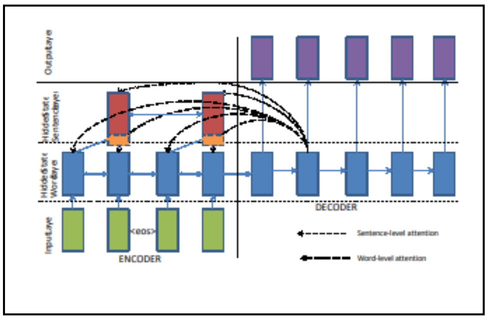

Figure 6: Nallapati et al., 2016 Hierarchical encoder with hierarchical attention

annotated Gigaword corpus(Rush et al., 2015), and training is done on 200-dimensional (Mikolov et al., 2013) word2vec vectors trained on Gigaword corpus(Rush et al., 2015). The results are compared with the different types of novel models and Rush et al., 2015 and validated on Gigaword and DUC-2003 corpus. The metrics used here are ROUGE-1, ROUGE-2, ROUGE-L. The copy percent is also shown in the table. The current model has the lowest copy percent(78.7%) which means the remaining words are abstract compared to the Rush et al., 2015 models which have a 91.5% copy rate i.e. less abstractive compared to the model Nallapati et al., 2016. For the DUC-2003, Nallapati et al., 2016 performed the best compared to the former.

So far we have seen the Neural Sequence-to-Sequence RNN which has outpowered the other models seen in both abstractive and extractive text summarization, to handle out-of-vocabulary words we have seen the concept of pointer network which copies the words from source documents if any oov occurs. But the base model Neural Seq2Seq RNN can't handle oov words, Neural Seq2Seq RNN + pointer network handles oov words but not the repetitive sentences as the authors are not keeping track of the generated sentences in the document.

2.3.3 Pointer-Generator Model

To handle the above drawbacks, the paper See et al., 2017 suggests the technique to do so.

The model here with the pointer-generator covers the oov words and incorporates the coverage criteria to keep track of the completed texts in order not to repeat the sentence. The

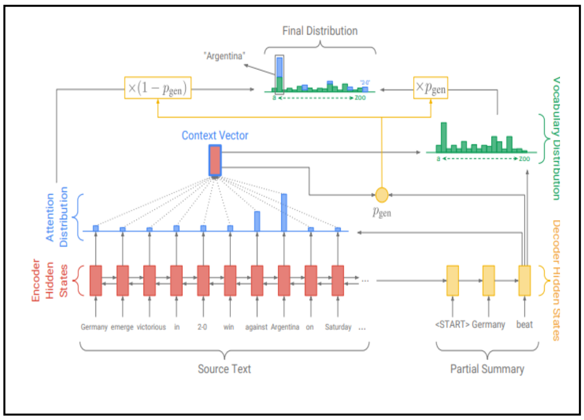

Figure 6: See et al., 2017 Pointer-generator model

model is trained on the CNN/Daily Mail (Hermann et al., 2015) dataset (Nallapati et al., 2016) and some scripts supplied by Nallapati et al., 2016. The outcomes of this model are compared with Nallapati et al., 2016 and they are very good for the metric ROUGE. The model See et al., 2017 is more abstractive compared to Nallapati et al., 2016. The best model is abstractive but it does not produce novel n-grams, whereas the baseline model of See et al., 2017 produces more novel n-grams and the reason will be because of the oov words which are treated as <UNK>. So far this is a very good model compared to all models that aim for abstraction and this model See et al., 2017 also doesn't seem to be perfect, and more research should be done regarding this.

2.4 Transformer-Based Models

Now the only ones left to see the area of study are pre-trained models and the transformer-based models. The transformer models are trained on a vast amount of data.

2.4.1 Transfer Learning with Finetuned Language Models

The paper Gehrmann et al., 2019 uses transfer learning instead of copy-attention mechanisms. The datasets used are TL;DR corpus which has user-written summaries from Reddit, which is an abstractive dataset. Many abstractive summarization models have an inductive bias because they always generate extractive summaries. When a model is trained on the abstractive dataset, it can gain abstraction from datasets. The authors here, train on TL;DR corpus and evaluate both TL;DR and CNN/DM corpus. The models used here are LSTM as a baseline, LSTM+Copy from See et al., 2017, Transformer+Copy and Transformer+Pretrain. The evaluation metrics used here are ROUGE and %novel n-grams and n-gram abstractiveness. The results show that the LSTM is biased over the short documents i.e. CNN/DM dataset and they perform worse than the transformers. Transformer+Copy performs better than Transformer+Pretrain because of the copy mechanism on the ROUGE metric. To see how the models fare over the abstractiveness we see the metric %novel n-gram and n-gram abstractiveness and we see that Transformer+Pretrain has a higher level of abstract summary. They concluded that the models that perform better are less abstract compared to the models that fared lower. The task of finding the abstractive summary is always challenging.

2.4.2 Self-Attention Guided Copy Mechanism

Advancing on the copy mechanism of the Transformer can be seen in the paper Xu et al., 2020. Here the copy mechanism is the graph-based self-attainment which captures the words from the source document nicely. Compared to

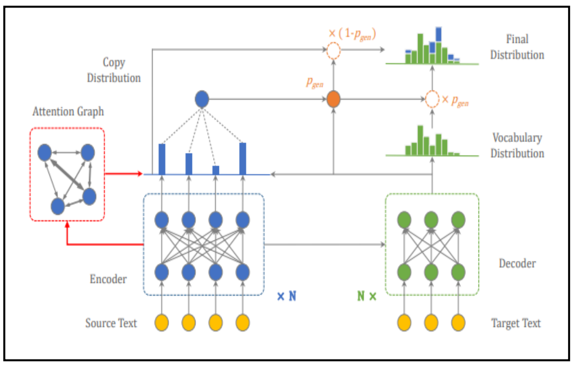

Figure7:Xu et al., 2020 Transformer+encoder self-attention graph

the working of the model Gehrmann et al., 2019 Transformer+Copy, this graph-based method copies the words accurately. Here the training data is CNN/DailyMail and Gigaword. This model Xu et al., 2020 works better compared to the Transformer+Copy Gehrmann et al., 2019 in terms of the accuracy in presenting the Out-of-Vocabulary words from the source document. In this paper, the authors incorporate graph-based self-attention guided copy mechanism into the Transformer model.

2.4.3 Pretrained Encoder like BERT for Abstractive Text Summarization

We will see the pre-trained encoders like Bidirectional Encoder Representations from

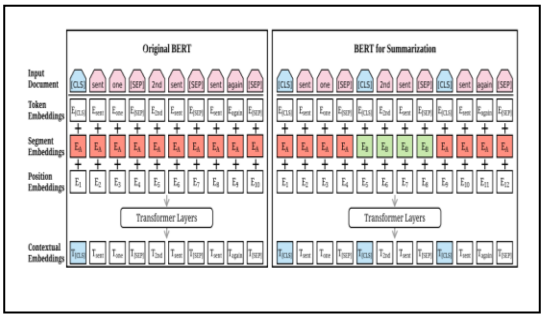

Figure 8: Liu et al., 2019 Architecture of the original BERT model (left) and BERTSUM (right)

Transformers (BERT; Devlin et al. 2019) presented in the paper Liu et al., 2019. Above Figure 8 is the architecture of BERT. This model follows a two-step approach of fine-tuning the encoder twice with extractive and abstractive summarization objectives.

The model is trained for 3 different datasets. They are CNN/Daily (Hermann et al., 2015) Mail, New York Times Annotated Corpus (NYT; Sandhaus 2008), and XSum (Narayan et al., 2018a). The model is evaluated on those 3 datasets The Bert Model outperforms the other models for any dataset. This model achieves state-of-the-art performances for both extractive and abstractive summarization. The previous models aimed for the better summarization task, here the authors aim for good document encoding.

The authors conclude that the model outperforms the other models of any text summarization task such as Extractive or Abstractive.

The general issues with the transformer models are large training data requirements, require high performing computers which are very expensive.

3 Background Evaluation

The evaluation metrics used to evaluate are ROUGE (Lin, 2004), BLEU, Human Evaluation, and perplexity.

ROUGE considers the overlap between n-grams in the output summary to the source document. It focuses on the recall. It measures how good is the generated summary to the source document. In ROUGE there are different types such as ROUGE-1, ROUGE-2, and ROUGE-n where they measure the overlap of unigrams, bigrams, and n-grams. ROUGE-L (Lin, 2004) is the overlap of the longest common subsequence. The higher the ROUGE scores are the better.

While ROUGE focuses on Recall, BLEU focuses on the precision of the summary with the original document. This metric sees the quality of the summary and its fluency.

Human Evaluation involves the individual evaluation criteria of everyone which helps in identifying fluency, readability, and coherence from a human perspective.

Perplexity is also a metric, which indicates the understanding of a summary. A lower perplexity score is a good one.

Perplexity = exp(average cross-entropy loss)

%novel n-gram is a proxy for abstractiveness and n-gram abstractiveness overcomes the drawback of normalization by %novel n-grams.

4 Challenges of the summarization task

The main challenges of the text summarization are:

i. Preserving the meaning of the document which is similar to the source document.
ii. Handling Out-of-Vocabulary Words
iii. Fluency and coherence
iv. Sentence repetition
v. Abstractive summaries

Out-of-vocabulary (OOV) words are handled by copy mechanisms. Many attention-based techniques are used for the fluency and coherence of the summarization. The sentence repetition is very challenging and is handled by the pointer-based techniques.

5 Summary

In the literature review of the text summarization task, we have seen the improvement of the models for extractive and abstractive over the years. The

Extractive task is easier compared to the abstractive task. We have seen the traditional methods of graph-based supervised and unsupervised ranking-based algorithms for extractive summaries. With the following improvements, we have seen the Neural Language Model for the text summarization which is also abstractive. To improve the accurate meaning of the sentence we have seen the copy mechanisms. To improve the out-of-vocabulary words we use pointer-based networks. The abstractive task came into the picture with the Sequence-to-Sequence models with an encoder and decoder. Here we get at least some abstractiveness compared to the extractive models. To not repeat the sentence in a document we use coverage with the help of pointer network seq2seq. And then we have seen the transformers and the BERT model, BERT outpowers all the models in both extractive and abstractive tasks. For the transformers, the research work needs to be done because the lower ROUGE value model is more abstractive and vice versa.

6 Future Scope for Improvements

There is always a need for advancements in text summarization, considering the challenges seen. Attention-based techniques are used for fluency and coherence, these long-range dependencies could further be improved to capture the source text effectively for the summarization. Handling Out-of-Vocabulary words is done by copy mechanisms, but the advancements need to ensure abstractiveness by just replacing it with novel words. Task-specific improvements should also be made as we just can't use computationally expensive methods for a simple task. Trained models on one type of dataset should also work very well on similar kinds of other data.

References

[1] Marina Litvak and Mark Last. 2008. Graph-Based Keyword Extraction for Single-Document Summarization. In *Coling 2008: Proceedings of the workshop Multi-source Multilingual Information Extraction and Summarization*, pages 17–24, Manchester, UK. Coling 2008 Organizing Committee.

[2] Dani Yogatama, Fei Liu, and Noah A. Smith. 2015. Extractive Summarization by Maximizing Semantic Volume. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1961–1966, Lisbon, Portugal. Association for Computational Linguistics.

[3] Hayato Kobayashi, Masaki Noguchi, and Taichi Yatsuka. 2015. Summarization Based on Embedding Distributions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1984–1989, Lisbon, Portugal. Association for Computational Linguistics.

[4] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

[5] Jianpeng Cheng and Mirella Lapata. 2016. Neural Summarization by Extracting Sentences and Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

[6] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

[7] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gu̇lçehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

[8] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

[9] Sebastian Gehrmann, Zachary Ziegler, and Alexander Rush. 2019. Generating Abstractive Summaries with Finetuned Language Models. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 516–522, Tokyo, Japan. Association for Computational Linguistics.

[10] Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

[11] Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-Attention Guided Copy Mechanism for Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1355–1362, Online. Association for Computational Linguistics.

[12] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Advances in Neural Information Processing Systems, pages 1693– 1701.

[13] Evan Sandhaus. 2008. The New York Times Annotated Corpus. Linguistic Data Consortium, Philadelphia, 6(12)

[14] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don’t Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[16] David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. Linguistic Data Consortium, Philadelphia, 4(1):34

[17] Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada. Association for Computational Linguistics.

[18] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

[19] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On Using Very Large Target Vocabulary for Neural Machine Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China. Association for Computational Linguistics.

[20] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[21] 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.

[22] Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In Proc. of SIGIR.

[23] Language Independent Sentence-Level Subjectivity Analysis with Feature Selection (aclanthology.org)

[24] LCSTS: A Large Scale Chinese Short Text Summarization Dataset (aclanthology.org)