# AutoHighlighter: Extract salient parts from Text (Project Final Report)

**Version 1.0**

**Yining Cao(rimacyn)**

## Abstract

Content representation in a concise form is of special needs nowadays. Using highlights to reflect the gist of a document can lessen our burden of information digestion in various aspects of life. This study will focus on using a sequence-to-sequence model for auto-highlighting. In order to solve the self-repetition problem of the output summarization and enable the model to cover a wide range of representative content, an intra-attentional layer was implemented.

Results are evaluated on the CNN/DailyMail dataset using ROUGE scores. The baseline model is a traditional attentional sequence-to-sequence model with ROUGE-1=33.42, ROUGE-2=14.63, ROUGE-L=34.77. Model performance improves significantly after leveraging an intra-attentional mechanism, with the average ROUGE scores increased by 1.88, 0.68, 1.30 in ROUGE-1,ROUGE-2,ROUGE-L respectively.

## 1 Introduction

With an advent of Internet and media, we are overwhelmed with thousands of news and reports everyday. That's why we need an auto-highlighter for content representation to help us summarize a document and get straight to the most important part of information.

Automatic text summarization is an important branch of natural language processing, which can be broadly classified into two categories: extractive and abstractive text summarization(Gambhir and Gupta, 2017). Extractive text summarization concatenates the most relevant sentences from the document to generate summary. It usually comprises of three major steps: intermediate representation of input text, sentence scoring, and sentence selection. Conversely, abstractive text summarization generates the summary by paraphrasing the main contents of the document using natural language generation techniques.

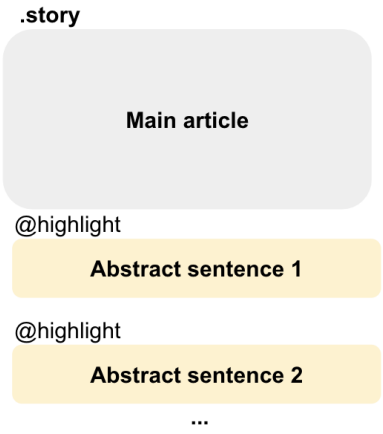
Text summarization is considered valuable in various aspects of our life, for example, lecture transcription(Miller, 2019), news (Wei and Gao, 2014),story telling(Liu et al., 2019),etc. However, it still remains a challenging problem. Even a study with state-of-art result maintains that the summaries generated from their model are still far behind gold summaries from humans.

Automatic text highlighting is similar to extractive summarization, though it considers less about the human-readability of the output. Current studies have identified self-repetition and representative part coverage as two main problems in modeling representative parts in text (See et al., 2017).

Various traditional methods for extractive text summarization proposed in the literature are based mainly on human-engineered features, as for example, combination of statistical and linguistic features such as term frequency, length and position (Mihalcea and Tarau, 2004).

This project will implement a new approach for constructing an extractive summarization model by using sequence-to-sequence model with attention mechanism to generate summary, while the generated words will be derived from the input sequence. The baseline model is adapted from SummaRuNNer (Nallapati et al., 2017). Further, to solve the problem of self-repetition in the generated sentences, an intra-attention mechanism as proposed by (Paulus et al., 2017) was leveraged. Experiment results show that an intra-attentional layer can significantly improve the ROUGE scores of the output and effectively solve the self-repetition and coverage problem.

Figure 1: Document abstract structure




## 2 Problem Definition and Data

The problem is to build a deep learning model to extract salient sentences from a single document. The input should be a piece of text, and the output should be the highlighted parts of the original text.

The model will be trained and evaluated on the CNN/DailyMail dataset(Hermann et al., 2015) using the ROUGE score(Lin, 2004) as metric. Results will be compared against both baseline approaches and results reported in my reference paper(Nallapati et al., 2017).

Previous models for text summarization are generally within two categories: (1)either implement sentence embedding based features to rank sentence from documents and select a subset of them to produce a concise and fluent summary (2) or use auto-encoding and deep generated models for abstractive summary.

The novelty and objective of this project is a combined implementation of the above two methods for extracting salient parts of a document. The high level concept is: using sequence-to-sequence model with attention mechanism to generate summary, while the generated words will be derived from the input sequence.

The model is trained and evaluated on CNN/DailyMail dataset which contains news articles and associated highlights. The structure of '.story' file is show in Figure 1. Figure 2 gives a summarization of the data samples used for training, validation and testing dataset for this project (number stands for number of samples used)

Figure 2: Summary of dataset

| CNN | | DailyMail | | Total |
|---|---|---|---|---|
| 92,579 | | 219,506 | | |
| Training | Validation | Testing | Invalid | 312,085 |
| 287 x 1000 | 13 x 1000 | 11971 | 114 | |

## 3 Related Work

For extractive summarization of text, both supervised learning and unsupervised learning methods have been used in previous studies. While supervised learning methods require gold summarization as ground truth, the unsupervised methods mostly rank sentences using heuristic scores, which makes it more adaptable to an unseen domain.

Earlier studies on text summarizing have proposed some linguistic and statistical features that can be used for sentence scoring and ranking. Common features include sentence position,length,frequency of keywords, etc(Mihalcea and Tarau, 2004). In KLSum(Haghighi and Vanderwende, 2009), the researchers proposed a Kullback-Liever(KL) divergence based summarizing method.

For the extractive summarising model, Metha, et al.(Mehta et al., 2018) apply an attention based LSTM network to select salient sentences based on context embedding. Miller(Miller, 2019) leveraged BERT model for text embeddings and K-Means clustering to identify sentences closest to the centroid summary selection for lectures.

Recent years, Neural encoder-decoder models, such as sequence-to-sequence models, have gain much attention in this domain. These models use recurrent neural networks such as long-short term memory network(LSTM) to encode an input sentence into a fixed vector. In the decoding process, this vector will be used for generating a new output sequence through another RNN(See et al., 2017). Nallapati et al. has built SummaRuNNer (Nallapati et al., 2017), sequence-to-sequence model, for text summarization. Their experiments reveals repeated phrases as a key problem with the generative models. To solve this problem, Paulus et al(Paulus et al., 2017) introduces a key attention mechanism called 'intra-temporal attention' for both encoding and decoding process. It solves the repeated problem by recording previous attention weights and penalizes tokens with high attention scores in the previous stage.

However, the sequence-to-sequence model will generate words freely, which is a desirable property for abstractive summarization, but not as for the extractive model. The idea of copy attentional parts from source document proposed in CopyNet(Gu et al., 2016) offers a solution for incorporating copying mechanism in sequence-to-sequence learning.

In this project, I will implement an sequence-to-sequence model for extractive text summarization with intra-attentional and adapt the idea from the CopyNet, by adding a cutoff after the attention layer, to enforce the model to do an extracting work.

## 4 Methodology

**Data preprocessing**

The data is from CNN/Daily Mail dataset(Hermann et al., 2015). In order to be consistent with our baseline model, adapted from Nallapati et al(Nallapati et al., 2017), I implement the similar data preprocessing method as is described in their study.

Main articles and abstract sentences are read separately from the original '.story' file. 114 files are excluded where the article texts are missing. The sentences are tokenized using Stanford CoreNLP. With the tokenized files, a size of 200,000 vocabulary is generated.

To support batch training, these files are processed into .bin format and be split into chunks of 1000 examples per chunk.

The whole process is shown in Figure 3. In the end, totally 287 chunks for training, 13 chunks for validation and 10 chunks for testing are created.

**Baseline Model**

The baseline model is a traditional sequence-to-sequence model with attention. In the encoding process, it uses a single layer bidirectional LSTM to produce a sequence of encoder hidden states $h_i$. For decoding, a single-layer unidirectional LSTM receives the input word embedding of the previous word $x_t$ and produces a decoder state $s_t$ on each time step $t$. The attention distribution is calculated as:

$$e^t = W_{atten}^T tanh(W_h h_i + W_s s_t + b)$$
$$a_i^t = softmax(e^t)$$

The final hidden vector $c_t$ for generating distribution over the vocabulary is obtained by a weighted sum over all previous hidden states:

$$c_t = \Sigma_i a_i^t h_i$$
$$P_{vocab} = W_{out}(W_{hid}[s_t; c_t] + b_{hid}) + b_{out}$$
$$P_{vocab}^* = softmax(P_{vocab})$$

In order to force the model to choose words from original article, I manually set $P(w) = 0$ if $w$ is not a word from the original article before doing the softmax.

Loss is calculated for each time step $t$ using the negative log likelihood of the target word $w_t^{tar}$ The overall loss for a sequence is the summation over all time steps:

$$loss_t = -logP(w_t^{tar})$$
$$loss = \frac{1}{T}\Sigma_{t=0}^T loss_t$$

**Model with intra-attention**

For the intra-attentional model I used to solve the repeated Follow the equation mentioned in (Paulus et al., 2017), each $e_i^t$ is calculated with a bilinear function:

$$e_i^t = s_t W_{attn} h_i$$

In order to penalize input tokens that have obtained high attention scores in past decoding steps, the attention weights are penalized using the following temporal attention function (with (t>1), $e_i^1$ is manually defined as $exp(e_i^1)$):

$$e_i^{t*} = \frac{exp(e_i^t)}{\Sigma_{j=1}^{t-1} exp(e_i^j)}$$
$$a_i^t = \frac{e_i^{t*}}{\Sigma_{j=1}^n e_j^{t*}}$$

Other neural network structures are the same as the baseline model.

**Training process**

The model is implemented using PyTorch with codes available on github.[1] For both the baseline and intra-attentional model, I use the following configuration:

embedding dimension: 256;

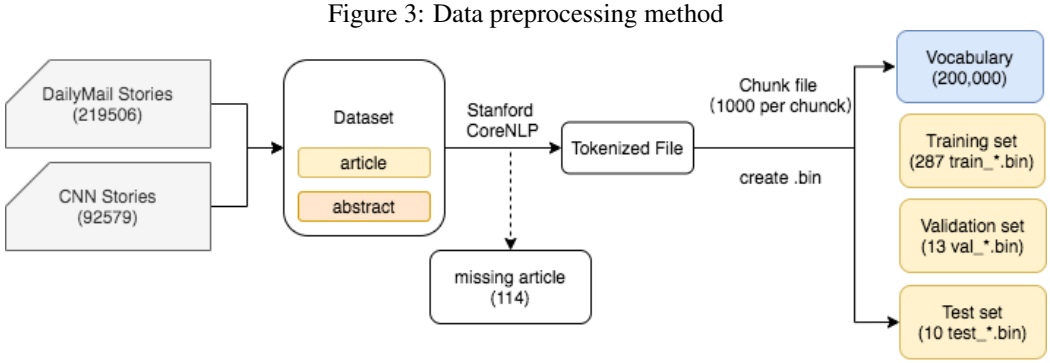
hidden dimension: 512;

max encoding steps: 200;

max decoding steps: 65;

[1] https://github.com/Rrrima/auto-highlighter

Figure 3: Data preprocessing method




max iterations: 50000;

Both the baseline model and the intra-attentional model are trained on the GreatLakes GPU.

The total loss for the baseline model descends from 6.42 to 0.93 on the training set, and reaches 1.25 and 1.83 on the validation set and testing set respectively.

For the intra-attention model, the total loss descends from 7.13 to 0.88 on the training set, and reaches 1.37 and 1.62 on the validation set and testing set respectively.

## 5 Evaluation and Results

**Evaluation metric**

Recall-Oriented Understudy for Gisting Evaluation (GOUGE)(Lin, 2004) score will be used for evaluation. It is a set of metrics for evaluating automatic summarization and machine translation in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation. Specificlly, the following 3 metric of ROUGE will be used:

- **ROUGE-1:**Refers to the overlap of unigram (each word) between the system and reference summaries.
  $GOUGE - 1 =$
  $$\frac{\sum_{S \in (GT)} \sum_{uni \in (S)} Count_{match}(uni)}{\sum_{S \in (GT)} \sum_{uni \in (S)} Count(uni)}$$
  where 'uni' stands for unigram, 'GT' stands for reference summary.

- **ROUGE-2:**Refers to the overlap of bigrams between the system and reference summaries.
  $GOUGE - 2 =$
  $$\frac{\sum_{S \in (GT)} \sum_{bi \in (S)} Count_{match}(bi)}{\sum_{S \in (GT)} \sum_{bi \in (S)} Count(bi)}$$
  where 'bi' stands for bigram.

- **ROUGE-L:**Longest Common Subsequence (LCS)(Lin and Och, 2004)based statistics. Longest common subsequence problem takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.
  $$GOUGE - L = \frac{LCS(X, Y)}{m}$$
  where, LCS(X, Y) is the length of a longest common subsequence of X and Y.

Besides using ROUGE score as subjective metrics, I also output the highlighted sentences to give an objective evaluation.

**Evaluate it with metrics - ROUGE Results**

The network architecture, hyper-parameters and training process are discussed in the Methodology part. There are totally 50 checkpoints saved during training and I used the last checkpoint file for testing.

The average ROUGE scores for the baseline and intra-attentional model are shown in Table 1. As the baseline method is largely depend on the model

| Model | ROUGE-1 | ROUGE-2 | ROUGE-L |
|---|---|---|---|
| MyBaseline | 33.42 | 14.63 | 34.77 |
| Intra-attentional | 35.30 | 15.31 | 36.07 |
| SummaRuNNer | 39.53 | 16.20 | 35.30 |

Table 1: ROUGE Results

SummaRuNNer(Nallapati et al., 2017), I also do a comparison between their reported socre and my baseline.

In terms of the ROUGE-1 and ROUGE-L scores, the performance of both the baseline model and the intra-attentional model are slightly worse than the one reported in SummaRuNNer(Nallapati et al., 2017). This is likely due to both an insufficient training and the data preprocessing method, which will be discussed in detail in the Discussion section.

However, if only compare the intra-attentional model with the baseline model, where same data preprocessing methods and iteration times are applied, we can see the intra-attention mechanism can significantly improve the summarization performance.

**Read the Summarization - Objective Results**

In order to have a more intuitive evaluation on the result, I randomly choose one sample from the testing dataset, and produce the output with ROUGE scores calculated for this specific sample (see Figure 4).

Figure 4: Sample output for baseline

Source Article (truncated)

-LRB- CNN -RRB- -- No. 1 **Florida State re-emerged in the second half** and rallied to defeat No. 2 Auburn 34-31 in the BCS National Championship college football game on Monday in Pasadena , California . **The Seminoles** -LRB- 13-0 -RRB- **outscored the Tigers** -LRB- 12-2 -RRB- 24-10 after halftime to **win their third national title** . It was FSU 's second BCS crown and third national title . **Heisman Trophy winner and** FSU quarterback **Jameis Winston** was 20-for-35 for 237 yards and **two touchdowns** , including the game-winner to Kelvin Benjamin with 13 seconds remaining . The game was the final championship contest under the current system , which featured the top two teams in the Bowl Championship Series poll . **Next season , the top four teams will meet in a playoff** where it will take two victories to win the title .

Baseline Output

Florida State defeat college football game. Second BCS crown win. Tigers win. Heisman Trophy winner was touchdowns. Heisman Trophy and the touchdowns. Winner yards and two touchdowns.

Highlight Output in Article

-LRB- CNN -RRB- -- No. 1 Florida State re-emerged in the second half and rallied to defeat No. 2 Auburn 34-31 in the BCS National Championship college football game on Monday in Pasadena , California . The Seminoles -LRB- 13-0 -RRB- outscored the Tigers -LRB- 12-2 -RRB- 24-10 after halftime to win their third national title . It was FSU 's second BCS crown and third national title . Heisman Trophy winner and FSU quarterback Jameis Winston was 20-for-35 for 237 yards and two touchdowns , including the game-winner to Kelvin Benjamin with 13 seconds remaining . The game was the final championship contest under the current system , which featured the top two teams in the Bowl Championship Series poll . Next season , the top four teams will meet in a playoff where it will take two victories to win the title .

ROUGE-1 : 31.43;
ROUGE-2 : 14.91;
ROUGE-L : 34.38

The red parts highlighted in the 'Source Article' part are derived from gold summary. In the 'Highlight Output in Article' part, light blue highlighted parts appear once in the output and dark blue highlighted parts appear multiple times in the output summary. We can have a intuition that self-repetition can be a problem for this baseline model.

As comparison, I choose the same article to generate summary using the intra-attentional model. The result is shown in Figure 5

Figure 5: Sample output for intra-attentional model

Intra-attentional Model Output

Florida State defeat college football game. Second BCS crown and Heisman Trophy win two touchdowns. Next season, top four teams meet.

Highlight Output in Article

-LRB- CNN -RRB- -- No. 1 Florida State re-emerged in the second half and rallied to defeat No. 2 Auburn 34-31 in the BCS National Championship college football game on Monday in Pasadena , California . The Seminoles -LRB- 13-0 -RRB- outscored the Tigers -LRB- 12-2 -RRB- 24-10 after halftime to win their third national title . It was FSU 's second BCS crown and third national title . Heisman Trophy winner and FSU quarterback Jameis Winston was 20-for-35 for 237 yards and two touchdowns , including the game-winner to Kelvin Benjamin with 13 seconds remaining . The game was the final championship contest under the current system , which featured the top two teams in the Bowl Championship Series poll . Next season , the top four teams will meet in a playoff where it will take two victories to win the title .

ROUGE-1 : 35.14;
ROUGE-2 : 16.67;
ROUGE-L : 35.29

We can see from the results that the result that with an optimization of the attention function, the model can give a more comprehensive summarization with more information captured. Moreover, the effect for solving the self-repeating problem is very significant.

## 6 Discussion

**ROUGE result discussion**

Compare to the baseline model, our intra-attentional model has better performance in all of the three ROUGE scores. The optimization of the attention layer contributes to this significant improvement. With the self-repeating problem solved by the penalization of previous important tokens (see methodology part for detail), the output summarization is able to cover more parts of the text, rather than constantly focus on several import words in previous steps. Thus, more important information can be captured within the fixed decoding steps(65).

However, in terms of the ROUGE-1 and ROUGE-2 score, the performances of both the baseline model and intra-attentional model are slightly worse than the one reported in SummaRuNNer(Nallapati et al., 2017). Despite a possibility of insufficient training, the data preprocessing method may account for the poorer per-

formance.

For the CNN/DailyMail dataset, most summarization corpora only contain human written abstractive summaries as ground truth. To solve this problem, the original paper used an unsupervised approach to convert the abstractive summaries to extractive labels. They selected sentences from the document that maximize the Rouge score with respect to gold summaries. However, I used the summaries from the dataset directly and did a hard cutoff($P(w) = 0, if w \notin article$) in the generation process to ensure every word is extracted from the original article. This can explain for the important words missing in the output (i.e. a worse ROUGE-1, ROUGE-2 score).

We can still see an improvement in the ROUGE-L score, which refers to the longest common subsequence of using the intra-attentional model. This can also be explained by its ability of covering more important parts of the text.

**Sample output discussion**

The sample output of the baseline is shown in Figure 4. From the Baseline output we can see that though the subjective metrics (i.e. ROUGE scores) are fairly well, it is not human-readable. Another problem is that it repeats itself when generating words (see dark blue highlighted parts). The repetition can cause two problem :(1)produce redundant output; (2) miss some important part as the attention mechanism will keep forcing the model to focus on some important words in the previous stage.

The repetition problem will be solved with the intra-attention mechanism as described in the previous section. We can see a more comprehensive and concise summarization in Figure 5.

As the objective of this project is to generate highlights to reflect the gist of a document, rather than give a coherent summary, the human-readability problem will not be solved in this project. But adding POS features to the input data could be a possible solution, since POS tags will teach the model to learn about grammar. Further, (Paulus et al., 2017) also mentioned using Reinforcement Learning as a solution to improve human readability.

## 7 Conclusion

In this study, I implement a sequence-to-sequence model for extractive text summarization. In order to solve the self-repetition and coverage problem of the output summarization, an intra-attentional layer was implemented.

With a penalization of tokens that obtained high attention score in the previous stages, the model is able to repeat words less and cover more important parts of the original text. The average ROUGE scores are improved significantly after applying the intra-attention mechanism (with 1.88, 0.68, 1.30 improvement in ROUGE-1,ROUGE-2,ROUGE-L respectively).

However, the performance is likely to be further improved with a better data preprocessing method (i.e. implement sentence comparison and extraction rather than directly using original summaries as labels).

## 8 Other things we tried

Before I tried the intra-attentional mechanism to solve the coverage problem, I first tried using different ratio of encoding and decoding steps. I want to share some interesting results here: For summary of the following paragraph:

| Source Article (truncated) |
| --- |
| The French prosecutor leading an investigation into the crash of Germanwings flight 9525 insisted Wednesday that he was not aware of any video footage from on board the plane . Marseille prosecutor Brice robin told CNN that `` so far no videos were used in the crash investigation . `` he added , `` a person who has such a video needs to immediately give it to the investigators . robin 's comments follow claims by two magazines , German daily Bild and French Paris match , of a cell phone video showing the harrowing final seconds from on board Germanwings flight 9525 as it crashed into the French alps. |

The intra-attentional model will generate different summaries when setting $R = \frac{max\_decoding\_steps}{max\_encoding\_steps}$ to different values:

- $R = 0.1$
  robin french prosecutor was not aware of videos .
- $R = 0.2$
  the french prosecutor insisted no videos of any videos in the crash .
- $R = 0.5$
  robin French prosecutor insisted he was not aware of any video footage . madsssrseille flight 9525 says he was not aware of the crash investigation.

Though setting different $R$ can not finally solve the repetition problem, it can be used to control the output verbosity. This observation can be implemented into a text summarization application.

For example, let the user select how detailed they want the summarization be and control it with the ratio of encoding and decoding steps.

## 9 Future work

**Better data preprocessing**

As is mentioned in the discussion part, I currently use the summaries from the dataset directly. Though by manually scanning the dataset, the human generated gold summaries are mostly extractions from the original text, there are still several abstractive words. To solve this problem, Nallapati et al(Nallapati et al., 2017) proposed an unsupervised approach of converting the abstractive summaries to extractive labels by extracting the sentences with maximizing the ROUGE scores. This is a promising practice to further improve the summarization performance.

**Alternative words selection in decoder**

Currently I did a hard cutoff($P(w) = 0, if w \notin article$) in the decoding process to ensure every word is extracted from the original article. Inspired by the unsupervised learning algorithm mentioned in the data preprocessing process, It is possible to apply the same unsupervised learning method for words selection.

**Integrating Reinforcement Learning**

As is suggested by Paulus et al(Paulus et al., 2017), implementing Reinforcement Learning (RL) algorithm to train RNN-based models for sequence generation is applicable. Their RL model on text summarization task outperforms the one trained with supervised learning algorithm. They first trained their model with supervised learning, then they continuously train it using RL and define the ROUGE-L score as the reward function. It will be interesting to implement this algorithm and see whether there is any performance improvement.

## 10 Work Plan

Generally, I kept pace with the work plan I had. I used a different baseline model from the one I suggested in the proposal. I did this switch because after reading more related papers, I found that a sequence-to-sequence model is much easier to implement and train than Generative Adversarial Networks (GAN), while it is as well likely to have a good performance on text summarization.

Besides the model selection and implementation part, which is of course, the most challenging part, I spent unexpected amount of time on setting up training environment.

**[Feb.5 - Feb.19]:**
Learning sequence-to-sequence models and paper reading;

**[Feb.19 - Mar.4]:**
Baseline model selection and implementing; Dataset preparation and preprocessing

**[Mar.4 - Mar.15]:**
Baseline model training and evaluation; implement possible solutions and iterate the system; In the meanwhile, I keep change the hyperparameters of the baseline model and re-train it for several times.

**[Mar.15 - Mar.25]:**
Finalize the baseline model; implement the intra-attentional mechanism

**[Mar.24 - Apr.3]:**
Intra-attentional model training and evaluation

**[Apr.3 - Apr.13]:**
Writing project update report

**[Apr.13 - Apr.18]:**
Resume training on the current checkpoints for both the baseline model and the intra-attentional model, to see whether the performance can improve.

**[Apr.18 - Apr.23]:**
Generate more outputs and write the blog, report about this project.

## References


Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review* 47(1):1–66.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393* .

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. pages 362–370.

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.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Association for

Computational Linguistics, Barcelona, Spain, pages 74–81. https://www.aclweb.org/anthology/W04-1013.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, USA, ACL '04, page 605–es. https://doi.org/10.3115/1218955.1219032.

Peter J Liu, Yu-An Chung, and Jie Ren. 2019. Summae: Zero-shot abstractive text summarization using length-agnostic auto-encoders. *arXiv preprint arXiv:1910.00998* .

Parth Mehta, Gaurav Arora, and Prasenjit Majumder. 2018. Attention based sentence extraction from scientific articles using pseudo-labeled data. *arXiv preprint arXiv:1802.04675* .

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. pages 404–411.

Derek Miller. 2019. Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165* .

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304* .

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368* .

Zhongyu Wei and Wei Gao. 2014. Utilizing microblogs for automatic news highlights extraction. In *COLING*. World Scientific, pages 872–883.