Summary Refinement through Denoising

Nikola I. Nikolov, Alessandro Calmanovici, Richard H.R. Hahnloser
Institute of Neuroinformatics, University of Zürich and ETH Zürich, Switzerland
{niniko, dcalma, rich}@ini.ethz.ch

Abstract

We propose a simple method for post-processing the outputs of a text summarization system in order to refine its overall quality. Our approach is to train text-to-text rewriting models to correct information redundancy errors that may arise during summarization. We train on synthetically generated noisy summaries, testing three different types of noise that introduce out-of-context information within each summary. When applied on top of extractive and abstractive summarization baselines, our summary denoising models yield metric improvements while reducing redundancy.[1]

1 Introduction

Text summarization aims to produce a shorter, informative version of an input text. While extractive summarization only selects important sentences from the input, abstractive summarization generates content without explicitly re-using whole sentences (Nenkova et al., 2011). In recent years, a number of successful approaches have been proposed for both extractive (Nallapati et al., 2017; Narayan et al., 2018) and abstractive (Chen and Bansal, 2018; Gehrmann et al., 2018) summarization paradigms. Despite these successes, many state-of-the-art systems remain plagued by overly high output redundancy (See et al. (2017); see Figure 3), which we set out to reduce.

In this paper, we propose a simple method (Figure 1, Section 3) for post-processing the outputs of a text summarization system in order to improve their overall quality. Our approach is to train dedicated text-to-text rewriting models to correct
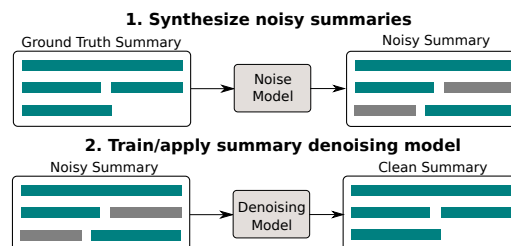

Figure 1: Overview of our approach to summary denoising. We alter ground truth summaries to generate a noisy dataset, on which we train denoising models to restore the original summaries.

errors that may arise during summarization, focusing specifically on reducing information redundancy within each individual summary. To achieve this, we synthesize from clean summaries noisy summaries that contain diverse information redundancy errors, such as sentence repetition and out-of-context information (Section 3.2).

In our experiments (Section 5), we show that denoising yields metric improvements and reduces redundancy when applied on top of several extractive and abstractive baselines. The generality of our method makes it a useful post-processing step applicable to any summarization system, that standardizes the summaries and improves their overall quality, ensuring fewer redundancies across the text.

2 Background

Post-processing of noisy human or machine-generated text is a topic that has recently been gathering interest. Automatic error correction (Rozovskaya and Roth, 2016; Xie et al., 2018) aims to improve the grammar or spelling of a text. In machine translation, automatic post editing of translated outputs (Chatterjee et al., 2018) is commonly used to further improve the translation quality, standardise the translations, or adapt them

---

[1]Code available at `https://github.com/ninikolov/summary-denoising`.

to a different domain (Isabelle, 2007).

In (Xie et al., 2018), authors synthesize grammatically incorrect sentences from correct ones using backtranslation (Sennrich et al., 2016a), which they use for grammar error correction. They enforce hypothesis variety during decoding by adding noise to beam search. Another work that is close to ours is (Fevry and Phang, 2018), where authors introduce redundancy on the word level in order to build an unsupervised sentence compression system. In this work, we take a similar approach, but instead focus on generating information redundancy errors on the sentence rather than the word level.

3 Approach

Our approach to summary refinement consists of two steps. First, we use a dataset of clean ground truth summaries to **generate** noisy summaries using several different types of synthetic noise. Second, we train text rewriting models to correct and **denoise** the noisy summaries, restoring them to their original form. The learned denoising models are then used to post-process and refine the outputs of a summarization system.

3.1 Generating Noisy Summaries

To generate noisy datasets, we rely on an existing parallel dataset of articles and clean ground truth summaries $S = \{s_0, , ..., s_j\}$. We iterate over each of the summaries and perturb them with noise, according to a *sentence noise distribution* $p_{noise} = [p_0, p_1, ..., p_N]$. p_{noise} defines the probability of adding noise to a specific number of sentences within each summary (from 0 up to a maximum of N noisy sentences), with $\sum p_{noise} = 1$.

For all experiments in this work, we use $p_{noise} = [0.15, 0.85]$ in order to ensure consistency, meaning that ~15% of our noisy summaries contain no noisy sentences, while ~85% contain one noisy sentence. Initial experiments showed that distributions which enforce larger or smaller amounts of noise lead to stronger or weaker denoising effects. Our choice of noise distribution showed good results on the majority of systems that we tested; we leave a more rigorous investigation of the choice of distribution to future work.

In addition to adding noise, we generate 3 noisy summaries for each clean summary by picking multiple random sentences to noise. This step increases the dataset size while introducing variety.

3.2 Types of Noise

We experiment with three simple types of noise, all of which introduce *information redundancy* into a summary. Our aim is to train denoising models that minimize repetitive or peripheral information within summaries.

Repeat picks random sentences from the summary and repeats them at the end. Repetition of phrases or even whole sentences is a problem commonly observed in text generation with RNNs (See et al., 2017), which motivates efforts to detect and minimize repetitions.

Replace picks random sentences from the summary, and replaces them with the closest sentence from the article. This type of noise helps the model to learn to *refine* sentences from the generated summaries, paraphrasing sentences when they are too long or contain redundant information.

Extra picks random sentences from the article, paraphrases them, and inserts them into the summary, preserving the order of the sentences as they appear in the article. With this type of noise, a model learns to delete sentences which are out of context or contain redundant information. To paraphrase the sentences, we use the sentence paraphrasing model from (Chen and Bansal, 2018), trained on matching sentence pairs from the CNN/Daily Mail dataset.

Mixture mixes all the above noise types uniformly into a single dataset, keeping the same dataset size as for the individual noise types. With mixture, we explore whether the benefits of each noise type can be combined into a single model.

4 Experimental Setup

Dataset We use the CNN/Daily Mail dataset[2] (Hermann et al., 2015) of news articles and summaries in the form of bullet points, and follow the preprocessing pipeline from (Chen and Bansal, 2018). We use the standard split of the dataset, consisting of 287k news-summary pairs for training and 13k pairs for validation. We follow Section 3.1 to generate noisy versions of the datasets to be used during training. During testing, instead of clean summaries that contain noisy sentences, we input summaries produced by existing extractive or abstractive summarization systems.

---

[2]https://github.com/abisee/cnn-dailymail

838

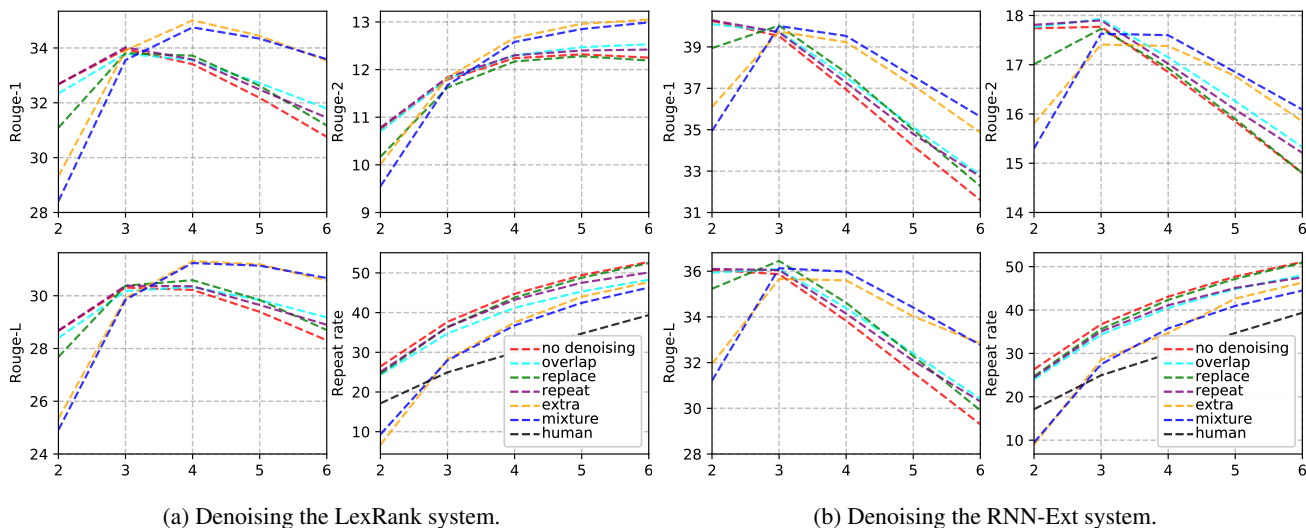(a) Denoising the LexRank system. (b) Denoising the RNN-Ext system.

Figure 2: Metric results (Rouge-1/2/L and Repeat rate) on denoising extractive summarization systems. The x-axis in all plots is the number of extracted sentences. `human` is the result of the ground truth summaries (only for the Repeat rate).

Denoising models For all of our denoising experiments, we use a standard bidirectional LSTM encoder-decoder model (Sutskever et al., 2014) with 1000 hidden units and an attention mechanism (Bahdanau et al., 2014), and train on the subword-level (Sennrich et al., 2016b), capping the vocabulary size to 50k tokens for all experiments[3]. We train all models until convergence using the Adam optimizer (Kingma and Ba, 2015).

In addition to our neural denoising models, we implement a simple denoising baseline, `overlap`, based on unigram overlap between sentences in a summary. `overlap` deletes sentences which overlap more than 80%[4] with any other sentence in the summary and can therefore be considered as redundant.

Evaluation We report the ROUGE-1/2/L metrics (Lin, 2004). We also report the *Repeat* rate (Nikolov et al., 2018) $rep(\mathbf{s}) = \frac{\sum_i o(\overline{\mathbf{s}_i}, s_i)}{|\mathbf{s}|}$ which is the average unigram overlap o of each sentence s_i in a text with the remainder of the text (where $\overline{\mathbf{s}_i}$ denotes the complement of sentence s_i). Since the repeat rate measures the overlapping information across all sentences in a summary, lower values signify that a summary contains many unique sentences, while higher values indicate potential information repetition or redundancy within a summary.

---

[3]We use the `fairseq` library https://github.com/pytorch/fairseq

[4]We empirically found that this threshold is sufficiently high to prevent unnecessary deletion and sufficiently low to detect near-identical sentences.

5 Results

5.1 Extractive Summarization

We experiment with denoising two extractive systems: LexRank (Erkan and Radev, 2004) is an unsupervised graph-based approach which measures the centrality of each sentence with respect to the other sentences in the document. RNN-Ext is a more recent supervised LSTM sentence extractor module from (Chen and Bansal, 2018), trained on the CNN/Daily Mail dataset. It extracts sentences from the article sequentially. Both extractive systems require the number of sentences to be extracted to be given as a hyperparameter, in our experiments we test with summary lengths ranging from 2 to 6 sentences[5].

The results on extractive summarization are in Figure 2a for *LexRank* and Figure 2b for *RNN-Ext*, where we plot the metric scores for varying numbers of extracted sentences for each of the two systems. For both LexRank and RNN-Ext, we observe ROUGE improvements after denoising over the baseline systems without denoising. The `repeat` and `replace` methods yielded more modest improvements of 0.5-1 ROUGE-L points, performing comparably to the simple `overlap` baseline. The most effective noise types are `extra` and `mixture`, yielding improvements of up to 2 ROUGE-L points for LexRank and up to 3.5 ROUGE-L points for RNN-Ext. The superior performance to `overlap` indicates that the addi-

---

[5]The average sentence count of a summary in the CNN/Daily Mail dataset is 3.88.

System	Denoising approach	ROUGE-1	ROUGE-2	ROUGE-L	Repeat	#Sent	#Tok
Human	-	-	-	-	28.86	3.88	61.21
Article	-	14.95	8.54	14.41	70.5	26.9	804
Article	Mixture	30.47	13.97	28.24	53.43	10.67	304.7
RNN	-	35.61	15.04	32.7	51.9	2.93	58.46
RNN	Overlap	36.41	15.92	33.73	26.84	2.39	47.31
RNN	Repeat	**36.5**	**15.94**	**33.79**	27.65	2.41	48.34
RNN	Replace	35.2	14.86	32.4	51.51	2.98	57.0
RNN	Extra	33.95	14.58	31.2	37.19	2.21	42.82
RNN	Mixture	35.08	15.3	32.44	27.27	2.2	42.14
RNN-RL	-	**40.88**	**17.8**	**38.54**	39.29	4.93	72.82
RNN-RL	Overlap	40.76	17.69	38.43	37.71	4.83	71.02
RNN-RL	Repeat	40.84	17.76	38.49	38.78	4.86	71.69
RNN-RL	Replace	40.78	17.72	38.46	39.24	4.93	72.2
RNN-RL	Extra	39.12	16.7	36.76	34.04	3.84	55.43
RNN-RL	Mixture	40.11	17.33	37.76	35.45	4.18	61.15

Table 1: Results on denoising abstractive summarization. **Repeat** is the Repeat rate, while **#Sent** and **#Tok** are the average numbers of sentences or tokens in the summaries. Best **ROUGE** results for each model are in bold. *Human* is the result of the ground truth summaries, while *Article* uses the original article as the summary.
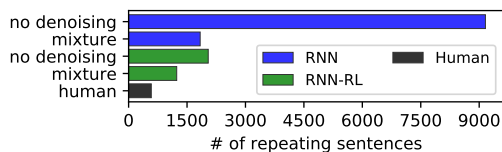

Figure 3: Number of sentence repetitions before and after denoising.

tional denoising operations learned by our models (see Figure 4a) are beneficial and can lead to more polished summaries that also may contain abstractive elements.

The gains from denoising are greater for longer summaries of more than two sentences. Long summaries are more likely to be affected by redundancy. For shorter summaries, denoising might lead to deletion of important information, thus denoising needs to be applied more carefully in such cases. Furthermore, for all sentence lengths and noise types, we observe a reduction in the Repeat rate after denoising, demonstrating that our approach is effective at reducing redundancy.

In Table 1, we additionally include the result from using the whole articles (*Article*) as input to our mixture model. Denoising is effective in this case, indicating that our approach may be promising for developing abstractive summarization systems that are fully unsupervised, similar to recent work in unsupervised sentence compression (Fevry and Phang, 2018).

5.2 Abstractive Summarization

For abstractive summarization, we test two systems. The first is a standard LSTM encoder-decoder model with an attention mechanism (*RNN*), identical to our denoising network from Section 4. The second, *RNN-RL*, is a state-of-the-art abstractive system proposed in (Chen and Bansal, 2018) that combines extractive and abstractive summarization using reinforcement learning. We train *RNN* ourselves, while for *RNN-RL*, we use the outputs provided by the authors.

Our metric results from denoising abstractive summarization are in Table 1. In Figure 3, we also compute the approximate number of sentence repetitions on the test set, by calculating the number of sentences that overlap significantly ($> 80\%$) with at least one other sentence in the summary.

For the *RNN* model, the repeat noise helps to remove repetition, halving our repetition metric, while boosting the ROUGE scores. This result is similar to our much simpler overlap baseline based on sentence deletion. The other noise types help to reduce redundancy, bringing the *Repeat* rate closer to that of *Human* summaries. This, however, comes at the cost of a decrease in ROUGE. For *RNN-RL*, while denoising helps to reduce repetition, none of our noise types managed to yield ROUGE improvements. One reason for this may be that this model already comes with a built-in mechanism for reducing redundancy which relies on sentence reranking (Chen and Bansal, 2018). However, as shown in Figure 3 (and in our example in Table 2), this model still generates many more sentence repetitions than found in human summaries. In overall, our approach is effective at reducing redundant information in abstractive summaries, how-
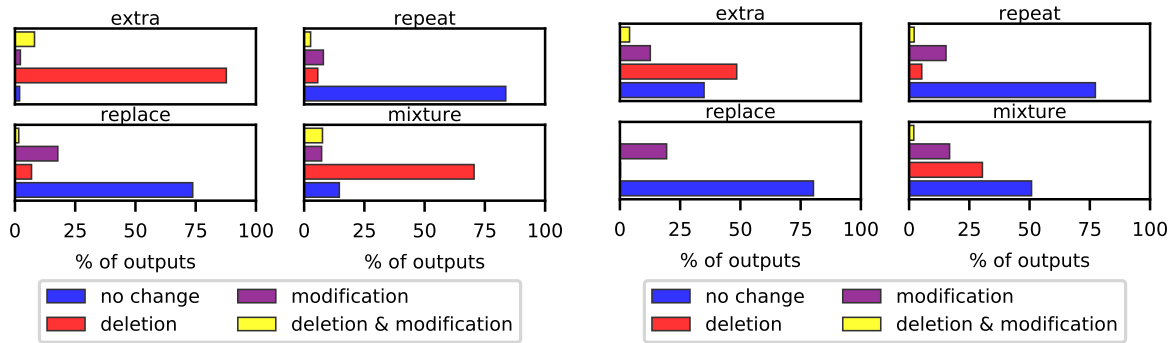
(a) RNN-Ext extractive system, extracting 5 sentences.

(b) RNN abstractive system.

Figure 4: Types of denoising operations applied to an extractive (left) and an abstractive (right) system (averaged over our test set).

ever this comes with a potential loss of information, which can lead to a reduction in ROUGE. Thus, our denoising methods are currently better suited for extractive than for absctractive summarization. Our work therefore calls for the development of novel types of synthetic noise that target abstractive summarization.

5.3 Analysis of Model Outputs

In Figure 4, we quantify the types of operations (deletion or modification of one or more sentences, or no change) our denosing models performed on the summaries produced by the extractive *RNN-Ext* (Figure 4a) and abstractive *RNN* system (Figure 4b). The `replace` and `repeat` noises are the most conservative, leaving over 75% of the summaries unchanged. `extra` is the most prone to delete sentences, while `repeat` and `replace` are most prone to modify sentences. We see a similar pattern for both extractive and abstractive summarization, with an increase of deletion for longer summaries produced by the extractive system. This indicates that our approach flexibly learns to switch between operations depending on the properties of the noisy input summary.

In Table 2 we show example outputs from denoising extractive and abstractive summaries produced for a sports article from our test set. All baseline summarization systems produced outputs that contain redundancy: for example, the first three sentences generated by the *RNN* system, and the 3rd and 4th sentences produced by the *RNN-RL* system are almost identical. To denoise the summaries, our models used diverse operations such as deletion of one or two sentences (e.g. *RNN* system, *Repeat* noise), rewriting (e.g. *RNN-RL*

system, *Replace* noise, where *"dinorah santana , the player s agent , said her client had rejected the offer of a three-year contract extension"* is paraphrased to *"the player s agent said she had rejected the offer of a three-year contract"*), or even a combination of deletion and rewriting (e.g. *RNN-RL* system, *Repeat* noise).

6 Conclusion

We proposed a general framework for improving the outputs of a text summarization system based on denoising. Our approach is independent of the type of the system, and is applicable to both abstractive and extractive summarization paradigms. It could be useful as a post-processing step in a text summarization pipeline, ensuring that the summaries meet specific standards related to length or quality.

Our approach is effective at reducing information repetition present in existing summarization systems, and can even lead to ROUGE improvements, especially for extractive summarization. Denoising abstractive summarization proved to be more challenging, and our simple noise types did not yield significant ROUGE improvements for a state-of-the-art system. Our focus in future work, will, therefore, be to estimate better models of the noise present in abstractive summarization, to reduce information redundancy without a loss in quality, as well as to target other aspects such as the grammaticality or cohesion of the summary.

Ground truth *(Rep=38.38)*: 1. dani alves has spent seven seasons with the catalan giants 2. alves has four spanish titles to his name with barcelona 3. the brazil defender has also won the champions league twice with barca		

RNN-Ext-4	**RNN**	**RNN-RL**
No denoising (R-1=33.6,Rep=45): 1. dani alves looks set to leave barcelona this summer after his representative confirmed the brazilian right-back had rejected the club 's final contract offer 2. alves has enjoyed seven successful years at barcelona , winning four spanish titles and the champions league twice 3. but the 31-year-old has been unable to agree a new deal with the catalan club and will leave the nou camp this summer 4. dinorah santana , the player 's agent and ex-wife , said at a press conference on thursday that her client had rejected the offer of a three-year contract extension , which was dependent on the player taking part in 60 per cent of matches for the club	*No denoising (R-1=34,Rep=79.6):* 1. dani alves has been unable to agree a new deal with catalan club 2. the brazilian has been unable to agree a new deal with catalan club 3. alves has been unable to agree a new deal with catalan club 4. alves has been linked with a number of clubs including manchester united and manchester city	*No denoising (R-1=31,Rep=51.6):* 1. dani alves looks set to leave barcelona this summer 2. alves has enjoyed seven successful years at barcelona 3. alves has been unable to agree a deal with the catalan club 4. the 31-year-old has been unable to agree a new deal 5. dinorah santana , the player 's agent , said her client had rejected the offer of a three-year contract extension
Replace (R-1=36.6,Rep=46.6): 1. Same 2. Same 3. Same 4. the player 's agent and ex-wife said at a press conference on thursday that her client had rejected the offer of a three-year contract extension	*Replace (R-1=34, Rep=79.6):* 1. Same 2. Same 3. Same 4. Same	*Replace (R-1=31, Rep=52.6):* 1. Same 2. Same 3. Same 4. Same 5. the player 's agent said she had rejected the offer of a three-year contract
Repeat (R-1=33.6,Rep=45): 1. Same 2. Same 3. Same 4. Same	*Repeat (R-1=28, Rep=41.4):* 1. Same 2. Deleted 3. Deleted 4. Same	*Repeat (R-1=24.2, Rep=36.1):* 1. Same 2. Same 3. Deleted 4. alves has been unable to agree a new deal 5. Same
Extra (R-1=43.6,Rep=36.8): 1. Same 2. Same 3. the 31-year-old has been unable to agree a new deal with the catalan club and will leave the nou camp this summer 4. Deleted	*Extra (R-1=37.2,Rep=92.8):* 1. Same 2. Same 3. Same 4. Deleted	*Extra (R-1=37, Rep=60.8):* 1. Same 2. Same 3. Same 4. Same 5. Deleted
Mixture (R-1=43, Rep=36.2): 1. Same 2. Same 3. Same 4. Deleted	*Mixture (R-1=28, Rep=41.43):* 1. Same 2. Deleted 3. Deleted 4. Same	*Mixture (R-1=37, Rep=60.8):* 1. Same 2. Same 3. Same 4. Same 5. Deleted

Table 2: Examples for denoising extractive and abstractive summarization. *Same* indicates a summary sentence has been unchanged, while *Deleted* indicates sentence deletion. In brackets, *R-1* denotes the *Rouge-1* score, while *Rep* denotes the *Repeat* rate.

Acknowledgments

References

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

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the wmt 2018 shared task on automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. pages 710–725.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of ACL*.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* 22:457–479.

Thibault Fevry and Jason Phang. 2018. Unsupervised sentence compression using denoising autoencoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pages 413–422. http://aclweb.org/anthology/K18-1040.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 4098–4109. http://aclweb.org/anthology/D18-1443.

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.

P Isabelle. 2007. Domain adaptation of mt systems through automatic postediting. *Proc. 10th Machine Translation Summit (MT Summit XI), 2007* .

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

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

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*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, pages 1747–1759. https://doi.org/10.18653/v1/N18-1158.

Ani Nenkova, Sameer Maskey, and Yang Liu. 2011. Automatic summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts of ACL 2011*. Association for Computational Linguistics, page 3.

Nikola Nikolov, Michael Pfeiffer, and Richard Hahnloser. 2018. Data-driven summarization of scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Paris, France.

Alla Rozovskaya and Dan Roth. 2016. Grammatical error correction: Machine translation and classifiers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 2205–2215.

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)*. volume 1, pages 1073–1083.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proc. of ACL*. Association for Computational Linguistics, pages 86–96. https://doi.org/10.18653/v1/P16-1009.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proc. of ACL*. Association for Computational Linguistics, pages 1715–1725. https://doi.org/10.18653/v1/P16-1162.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, pages 619–628. https://doi.org/10.18653/v1/N18-1057.