Post-Editing Extractive Summaries by Definiteness Prediction

Jad Kabbara and **Jackie Chi Kit Cheung**[†]
School of Computer Science, McGill University, Montreal, QC, Canada
Montreal Institute for Learning Algorithms (Mila), Montreal, QC, Canada
{jad, jcheung}@cs.mcgill.ca
[†] Canada CIFAR AI Chair

Abstract

Extractive summarization has been the mainstay of automatic summarization for decades. Despite all the progress, extractive summarizers still suffer from shortcomings including coreference issues arising from extracting sentences away from their original context in the source document. This affects the coherence and readability of extractive summaries. In this work, we propose a lightweight post-editing step for extractive summaries that centers around a single linguistic decision: the definiteness of noun phrases. We conduct human evaluation studies that show that human expert judges substantially prefer the output of our proposed system over the original summaries. Moreover, based on an automatic evaluation study, we provide evidence for our system's ability to generate linguistic decisions that lead to improved extractive summaries. We also draw insights about how the automatic system is exploiting some local cues related to the writing style of the main article texts or summary texts to make the decisions, rather than reasoning about the contexts pragmatically.

Source Text:
The school had to deal with a suspicious package received early in the morning. A student thought to be from another district addressed a mail that had a very strong smell. Police was called in. The student was eventually questioned by the police for 5 hours.

Original Extractive Summary:
The school had to deal with a suspicious package received early in the morning. Police was called in. **The** student was eventually questioned by the police for 5 hours.

Post-Edited Pseudo-Extractive Summary:
The school had to deal with a suspicious package received early in the morning. Police was called in. **A** student was eventually questioned by the police for 5 hours.

Figure 1: A simple change to an article choice (in bold) in the extractive summary can improve its readability and coherence.

1 Introduction

More than half a century after Hans Peter Luhn's seminal work (1958), automatic summarization remains a challenge, one that is increasingly pressing with the explosion of information online and elsewhere. One of the proposed approaches is extractive summarization: the task of selecting spans, typically sentences, from a source text such that they best convey the overall meaning. It is the most popular approach given its simplicity and scalability compared to more sophisticated abstractive approaches. The simplicity of this method, however, is not without its costs, as extractive summaries are known to suffer from a variety of issues. In addition to problems pertaining to verbosity (Barzilay et al., 1999), a system that centers around sentence extraction is inherently exposed to the risk of selecting a sentence that depends on a non-selected context, thereby affecting the summary's overall coherence. Examples include coreference issues (Steinberger et al., 2016) (e.g., selecting a sentence with an anaphor that refers to an entity in a non-selected previous sentence) and breaks in the pragmatic context (Hutchins, 1987) (e.g., a presupposition triggered in a selected sentence and corresponding to an event/proposition that appeared in a non-selected sentence).

In this work, we ask the following question: Can a lightweight post-editing step following the generation of extractive summaries, for instance along one specific linguistic decision, lead to an improvement in the quality of those summaries?

We focus in this work on predicting the definiteness of noun phrase articles. We propose a lightweight method for post-editing extractive summaries, which consists of a definiteness prediction model that decides whether articles in the extractive summaries should be kept as is or modified (including the possibility of being removed altogether). The goal of this post-editor is to improve the overall quality of the summary in terms of coherence and readability. We believe definiteness

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3682–3692
November 7–11, 2021. ©2021 Association for Computational Linguistics

is an attractive case study since it is an interesting test-bed for pragmatic reasoning as both contextual and local cues play a crucial role in deciding whether a given article is appropriate or not in a given context.

Consider the motivating example in Figure 1. Assuming an extractive summarizer selected the first, third and fourth sentences to be included in the summary, the final summary would be incoherent as the last sentence would refer to "*the* student" without it being introduced earlier in the context. A simple change to the article (the → a) would improve both the coherence and readability of the summary as seen in the last part of the figure.

In this work, we focus on post-editing extractive summaries to form pseudo-extractive outputs, rather than directly developing an abstractive summarizer, which we see as a separate (but worthy) use case. Compared to full-fledged abstractive summarization, limited post-editing is less likely to lead to problems of factual correctness and consistency, which are a known issue of existing abstractive systems (Cao et al., 2018; Goodrich et al., 2019; Kryściński et al., 2019).

We conduct two studies to understand different aspects of the problem using two English datasets, CNN/DailyMail and PubMed. First, we examine how often expert judges prefer summaries modified by such a system over the original version of generated extractive summaries. For the second study, we carry out an annotation study to obtain gold standard annotations on the definiteness of noun phrases in sampled subsets of extractive summaries that are generated by different summarizers for both CNN/DailyMail and PubMed. By comparing our model's decisions to the collected annotations, we can evaluate its performance using standard classification accuracy.

Our contribution is three-fold. First, we provide evidence that human expert judges show substantial preference for the summaries modified by our proposed system over the original extractive summaries in terms of coherence and readability. Second, we collect gold standard annotations on the definiteness of noun phrases in sampled subsets of generated extractive. This collected dataset of annotated extractive summaries can be useful for further development of similar systems. Third, using the collected annotations, we show that our system generates decisions that highly overlap with those of expert judges, further validating the efficacy of our proposed method. We also present insights into how the automatic system is exploiting some local cues related to the writing style of the main article texts or summary texts to make the decisions, rather than pragmatically reasoning about the contexts. Overall, we show that our findings generalize over multiple combinations of datasets and summarizers, thus demonstrating further the efficacy of our method.

2 Related Work

2.1 Extractive Summarization

There exists a long line of work on extractive summarization beginning as early as the mid-1950s. For a comprehensive review, the reader is referred to Nenkova and McKeown (2011). More recent approaches are based on neural networks including sequence-to-sequence models (Sutskever et al., 2014; Cho et al., 2014). These consist of MLE-based approaches (Cheng and Lapata, 2016; See et al., 2017; Nallapati et al., 2017) and RL-based approaches (Paulus et al., 2018; Dong et al., 2018). Recently, extractive summarization models that are based on fine-tuning pre-trained transformers have also shown strong performance (Liu and Lapata, 2019; Zhong et al., 2020).

Extractive summarizers are known to suffer from issues such as verbosity (Barzilay et al., 1999), coreference issues (Steinberger et al., 2016) (e.g., selecting a sentence with an anaphor that refers to an entity in a non-selected previous sentence) and breaks in the pragmatic context (Hutchins, 1987) (e.g., a selected sentence containing a presupposition that is linked to an event/proposition appearing in a non-selected sentence). Accordingly, we propose to exploit one linguistic phenomenon, namely, definiteness, in an attempt to provide a post-editing step that can improve the quality of extractive summaries.

While recent approaches have been pushing on the abstractive front, we note that, in various domains, extractive summarization is still the clear favorite due to domain restrictions and limitations in abstractive systems. Indeed, extractive models are still attractive in applications where faithfully preserving the original text is the priority. For example, guaranteeing the factual correctness of a summary can be integral in the health or scientific domains, which is a known weakness of current abstractive methods (Kryściński et al., 2019).

2.2 Definiteness Prediction

The question of definiteness has been extensively covered in the areas of linguistics and philosophy of language with early work that studies the nature and properties of definiteness dating back as early as (Russell, 1905) and (Strawson, 1950). In the computational linguistics literature, several models for definiteness prediction were proposed such as (Knight and Chander, 1994; Minnen et al., 2000; Han et al., 2006; Gamon et al., 2008). De Felice (2008) presented a logistic regression classifier extracting a number of linguistically motivated features from the context of each head noun. The most recent work (Kabbara et al., 2016) presents an attention-based RNN that achieves state of the art on definiteness prediction and investigates, among other factors, the effect of having a local or wider context. In our work, we adopt this model as the basis for the proposed post-editing step.

3 Proposed Post-Editor Method

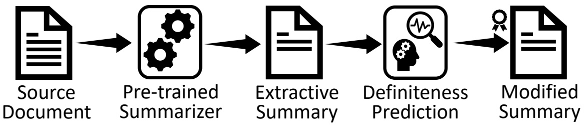

Figure 2: Diagram depicting our proposed method.

The learning task can be stated as follows: Given a document $D = \{s_1, \ldots, s_n\}$ with n sentences, a pre-trained extractive summarizer, f, generates a summary $S = f(D) \subset D$ with the length of S being $m < n$. The generated summary is then passed to a post-editing step in which decisions are made regarding the definiteness of noun phrases (NPs). Thus, a definiteness prediction model g generates a modified summary $S' = g(\tilde{S})$ which we refer to as *pseudo-extractive* summary. The goal is thus to compare the final output to the original summary to understand whether such a post-editing step improves the coherence and readability of extractive summaries. Figure 2 depicts the proposed pipeline.

3.1 Extractive Summarization

In order to focus the investigation solely on the effect of leveraging the discussed pragmatic knowledge, the learning task is concerned with single document summarization–as opposed to multi-document summarization where there is an added layer of complication regarding generating a coherent output using sentences from multiple documents.

In our work, we experiment with three different summarizers: MatchSum (Zhong et al., 2020) casts the extractive summarization task as a semantic text matching problem and is currently state-of-the-art on both CNN/DailyMail and PubMed. HipoRank (Dong et al., 2021) is a recent unsupervised graph-based ranking model for extractive summarization of long scientific documents with competitive performance on PubMed. Since it's tailored for long scientific documents, we use HipoRank for PubMed only. Finally, to have another set of results for CNNDM, we use BanditSum (Dong et al., 2018) an RL-based neural extractive summarizer with near-SOTA performance on CNNDM (better than HipoRank). To generate summaries, we use the source code made public by the authors.[1][2][3]

3.2 Definiteness Prediction

For the second step of predicting the definiteness of NPs, we adopt the methodology of Kabbara et al. (2016) in which they present an LSTM-based (Hochreiter and Schmidhuber, 1997) learning model for definiteness prediction. The learning task is a three-way classification where the labels represent one of three classes: “the", “a" (or “an") and “none". In order to explore the suitability and performance of different learning models on this task, we explore the use of a logistic regression classifier (De Felice, 2008), an LSTM model and a BERT-based (Devlin et al., 2019) neural model which has shown strong performance across a wide range of NLP tasks (Rogers et al., 2020).

3.2.1 Model Description

The first model is a logistic regression classifier which learns the probabilities describing the possible outcomes of an input using a logistic function.

The LSTM model is first fed a sequence of (one-hot encoded) input tokens representing the sample. The tokens are then embedded using pre-trained word representations. The resulting embedded vectors are encoded by a number of stacked LSTM recurrent layers. We explore in Section 7 the effect of having a unidirectional or bidirectional recurrent layer. The last hidden state is then fed to a linear layer followed by a softmax unit. To reduce the effect of overfitting, we apply dropout (Srivastava et al., 2014) on the embedding layer and hidden layers. As a note, in preliminary experiments, we

[1] https://github.com/mirandrom/HipoRank
[2] https://github.com/yuedongP/BanditSum
[3] https://github.com/maszhongming/MatchSum

tried the same model architecture but with GRU cells (Cho et al., 2014) instead of LSTM cells, however, the performance on the development set was worse in the case of GRU layers. Accordingly, we adopted the LSTM cell for our experiments.

In the BERT-based model, a sequence of input tokens is fed into a pre-trained BERT Model which produces representations that are passed to a number of stacked GRU layers (unidirectional or bidirectional). We use here GRU layers instead of LSTM because our preliminary experiments showed that a combination of BERT followed by GRU layers outperformed one with LSTM layers instead on the development set. The last hidden state is fed to a linear layer followed by a softmax unit. Similarly, we apply dropout on the hidden layers.

3.2.2 Input Representation

The input to the logistic regression classifier (De Felice, 2008) is a set of different types of manually-constructed linguistic features extracted from a fixed window surrounding the head noun of a noun phrase such as noun type, named entity or not, singular or plural, WordNet category and POS tags of the surrounding tokens. For more details, the reader is referred to (De Felice, 2008).

The two other models are trained on data samples that are constructed according to the configurations proposed in (Kabbara et al., 2016), namely the local context and the extended context. A sample in the local context configuration is defined to be the set of tokens from the previous head noun of a noun phrase up to and including the head noun of the current noun phrase. For example, take the following passage (head nouns indicated in bold):

Example 1 *The newly elected* **mayor** *plans to actively fight* **corruption** *plaguing the* **city**.

Noting that all instances of the articles in question (the, a/an) are removed from all the data (training/validation/testing), the following samples –relying on local context– are shown, with their labels: *newly elected mayor* – 'the', *plans to actively fight corruption* – 'none', *plaguing city* – 'the'.

Since Kabbara et al. (2016) provide evidence that an extended context leads to a better performance on their task of definiteness prediction, we explore using the extended context which constructs the sample in the same way (as described above) and, in addition, tokens from the previous sample(s) are added sequentially (in reverse) until a pre-specified total number of tokens per sample is reached. Similar to (Kabbara et al., 2016), we set that number to be 50.

4 Experimental Setup

4.1 Datasets

In our work, we use two datasets: CNN/DailyMail (Hermann et al., 2015) and PubMed (Cohan et al., 2018). CNN/DailyMail contains news articles and associated highlights, i.e., a few bullet points giving a brief overview of the article. The dataset is a collection of 93K articles from CNN and 220K articles from Daily Mail. Approximately 90k documents and 197k documents are used for training, respectively, in the CNN and Daily Mail portions of the dataset. The PubMed dataset consists of long and structured scientific papers obtained from the PubMed repository of biomedical research papers. The abstracts are considered to be the summaries of the articles. The dataset consists of 133K articles of which 120K are used for training.

To obtain the data for the second step (definiteness prediction), we first parse each dataset using Stanford CoreNLP (Manning et al., 2014) and then extract all of the NPs present in the parsed dataset whose head noun's POS tag is one of `NN`, `NNS`, `NNP`, or `NNPS`. We do not lemmatize and ignore case and punctuation. As mentioned before, we remove all instances of the relevant articles (the, a, an) from all of the datasets. The numbers of training samples are as follows: For CNNDM, 48M samples from the stories and 3.9M samples from the summaries. For PM, 69M samples from the articles and 6M from the summaries.

4.2 Training Details

The logistic regression classifier is implemented using the scikit-learn library (Pedregosa et al., 2011) with all the corresponding default parameters. For the LSTM model, we use a vocabulary of size 30,000 and we initialize the word embeddings with GloVe vectors (Pennington et al., 2014) having 300-dimensions and trained on the 840 billion token version of the Common-Crawl corpus. For the LSTM model, unknown words are randomly initialized according to a normal distribution to the same size as the GloVe embeddings. For BERT, we use the bert-base-uncased implementation by HuggingFace (Wolf et al., 2019) which implements a 12-layer 768-hidden 12-head 110M-parameter version of the model that was trained on lower-cased

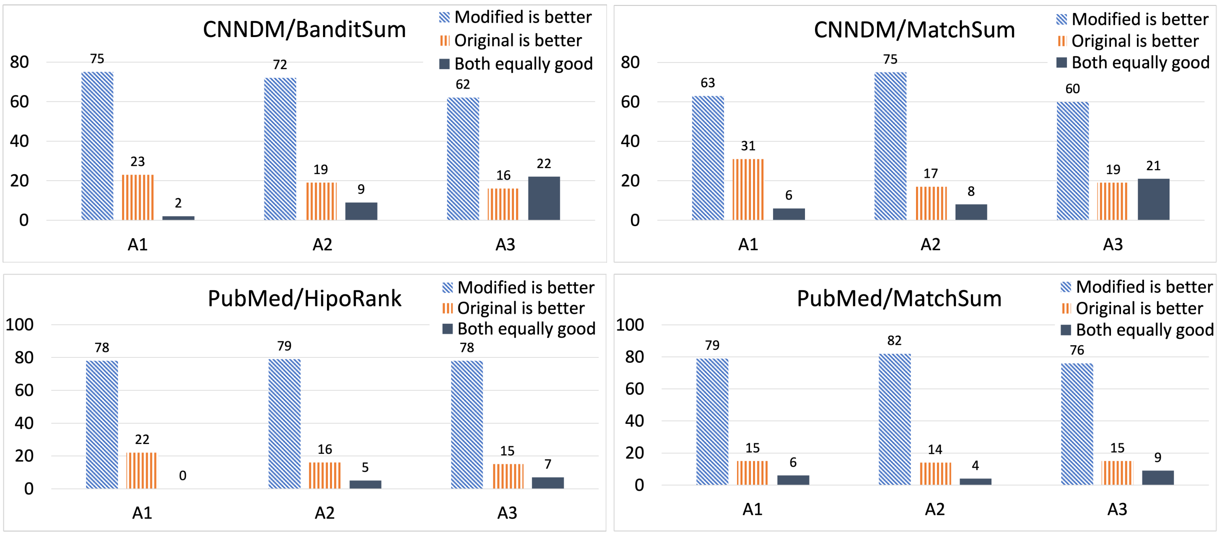

Figure 3: Preference judgement scores of the three judges A_1, A_2 and A_3 across various summarizers and datasets.

English text. All model hyperparameters are kept as default. During training, we freeze the weights of the BERT part of the corresponding model. This is to ensure a fair comparison (in terms of trainable parameters) between the LSTM model and a BERT-based model that contains recurrent layers as well. In the appendix, we explore the effect of fine-tuning BERT on its performance on our task.

Both neural models are implemented in PyTorch (Paszke et al., 2019). They are trained to minimize the standard cross-entropy cost with Adam (Kingma and Ba, 2015) as the optimizer with all default parameters except for the learning rate. Following hyperparameter search, we found the following hyperparameters to work best: 0.0001 for the learning rate, 128 for the mini-batch size and 0.6 for the dropout probability. We train the models for a maximum number of 35 epochs and to reduce the effects of overfitting we stop the training if the accuracy on the dev. set does not improve for 10 epochs. All test set results are reported based on the best trained model as measured on the dev. set.

5 Study 1: Preference Judgments

5.1 Methodology

In this study, we attempt to understand the effect of the proposed post-editing step on the quality of generated extractive summaries. We randomly sample 100 summaries generated by the summarizers for CNN/DailyMail and PubMed, pre-process them (See Section 4.1 for details) and then pass them through a definiteness prediction model to generate decisions that inform us on whether the noun phrases in those summaries should have an article (the, a/an) or not. We use the LSTM model with the best performance on the development set (8 layers, 2048 units – See Section 7 for a full comparison of models and hyperparameter search details). The resulting modified summaries are then given along with their corresponding original summaries (generated by the summarizers) to three annotators that are native speakers of English and paid 15 CAD/hour for their work. We ask them to evaluate which passage is better by choosing the one that is more coherent, more readable and/or more fluent. We also give the option of specifying that both versions are equally good. To reduce any biasing, the passages are anonymized in the sense that the judges do not know which of the two passages is the modified summary. We also randomize the order of the two passages in a given pair (i.e. in some instances, the modified summary is given as Passage A and in the rest as Passage B).

5.2 Results

The results of the human evaluation are given in Figure 3. We notice that across all combinations, the judges significantly prefer the modified version (on average, approx. 3 times). Furthermore, on average across the 4 scenarios, in 46% of the (overall) cases, the judges demonstrated full agreement in terms of their preference of the modified version. An interesting observation is that, as expected, the scores were higher in the PubMed experiments. This is because PubMed summaries are longer on average than CNNDM summaries. Accordingly, on average, there are more NPs in PubMed summaries, thus more possibilities for our system to lead to changes. Overall, the findings of this study constitute strong evidence that a light-weight post-editing step focusing on NP definiteness has the potential to improve the quality of extractive summaries in terms of coherence, readability and overall flow.

6 Study 2: Automatic Evaluation of the Post-Editor Method

In this study, we carry out a human annotation study to obtain gold standard annotations on the definiteness of noun phrases in sampled subsets of extractive summaries that are generated by different summarizers. By comparing our model's decisions to the collected annotations, we can evaluate its performance using standard classification accuracy. Also, the collected dataset of annotated extractive summaries can be useful for further development and evaluation of similar post-editing systems.

6.1 Methodology

We randomly sample 100 extractive summaries generated by the different summarizers (BanditSum, MatchSum, HipoRank) for the two datasets CNNDM and PubMed. We remove all instances of articles (the, a/an) and replace them by a blank. We also include a blank for the "none" cases. We ask three graduate students that are native speakers of English to fill the blanks with the appropriate articles or to keep them blank such that the summaries are the most coherent and readable to them. The annotators are paid 15 CAD/hour for their work. The resulting annotations show a high inter-annotator agreement ranging from 0.66 to 0.72 (Fleiss Kappa measure) across the 4 different combinations of summarizers/datasets. To evaluate the models' performance, we compare the decisions made by the models on the same 100 samples to those done by the three annotators. We evaluate how the models perform in terms of overlap with each of the annotators and compute the average overlap score for each model. In each case, the figures represent the test set performance of the best model as measured on the development set. In Section 7, we discuss the different hyperparameters that were examined and their effect on the models' performance.

To better understand how the data size and the linguistic variation between the writing of source documents and summaries affect the model performance, we investigate training on the following:

1. All source documents (CNNDM stories 48M samples; PubMed articles 69M samples).
2. A subset of source documents (CNNDM 3.9M samples; PubMed 6M samples).
3. Summaries (CNNDM 3.9M samples; PubMed 6M samples).
4. A combination of the last two datasets.

The rationale behind these variations is to understand how the difference in structure and style between source documents and summaries affects the performance of the models. Indeed, since the summarizer is extractive, the generated summaries should in principle closer in style to the source documents. Thus, we expect the model to perform better when it is trained on source documents. However, since the source documents dataset is much bigger than the summaries dataset, we also train on a subset of source documents of comparable size to that of the summaries dataset. This is to isolate the effect of dataset size and focus the comparison on the style difference between source documents and summaries. Finally, we present the "source + summaries" dataset in an attempt to understand whether training on both types of data can lead to some compound effect in terms of performance improvement.

6.2 Results

Figure 4 shows the results across the different combinations of summarizers and datasets and for the different training sets. First, we notice that the performances of the BERT and LSTM models are overall comparable with LogReg consistently performing the worst. We notice a general difference in trends between the CNNDM and PubMed scenarios. Isolating the training size effect, the BERT and LSTM models trained on the summaries (SM) score respectively approx. 6 and 3 points less than those that trained on the subset of stories (sub_ST) for the two CNNDM cases. This is due to the fact that the test samples in question are summaries generated by extractive summarizers. Accordingly, the generated summaries are closer in distribution to the stories compared to the golden summaries which are essentially story highlights written by the editors of the respective newspapers and which could differ substantially in style and structure from the stories. This shows the effect of the type (style) of training data. Also, for CNNDM, when we add more training data (ST), the performance goes

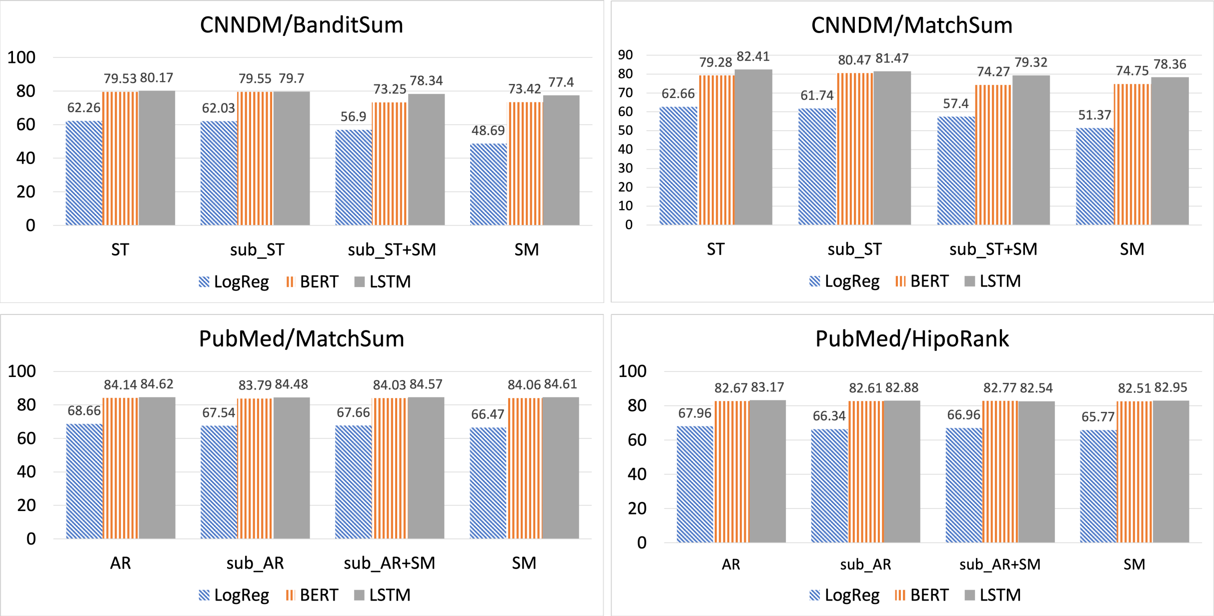

Figure 4: Performance of the learning models in terms of (average) overlap (in %) between the models' decisions and those of the annotators on 100 randomly sampled summaries generated by the different summarizers. Abbreviations: ST: Stories, AR: Articles, SM: Summaries, sub: subset.

slightly up but not enough given that ST is almost 10 times larger than sub_ST. Moreover, the models trained on both stories and summaries perform worse than those trained solely on the subset of stories, suggesting again that the performance was hurt due to training on data that is now less similar to the test data. For PubMed, the trends are different and the performance across the 4 training sets is more homogeneous. This is explained by the fact that the summaries are abstracts of the articles and so one would not expect the writing style to be different between the abstract and the body of the article. This explains the negligible performance difference between the two cases SM and sub_ST. Moreover, for this dataset, the results show that additional training data does not lead to a higher performance as the performance difference between ST and sub_ST is also negligible. On a separate note, we point out that we do not include ROUGE scores as part of our evaluation because it primarily measures semantic content, not coherence or referential clarity. Since we're only changing articles, ROUGE is not expected to change much (or even not at all in case stopwords (including articles) are filtered out before computing ROUGE as is common practice).

Focusing on the CNNDM scenarios where there exists a difference in style between source and summary, the results seem to suggest that the current models pay attention to the source of the data rather than actually attempting to reason about the pragmatics of the decision. Indeed, a model seems to do well when the source of a test sentence matches the training data but otherwise does less well. This can be seen as evidence that the automatic system is exploiting some cues related to the style of how main article texts or summary texts are written to make the decisions.

In conclusion, Study 2 shows that our proposed system is robust and generalizes to various changes in dataset size, data type/style, problem domains and summarizers. The results show our system making decisions that highly overlap with those of expert judges (the highest being 82.41% for CNNDM and 84.62% for PubMed). This shows that our system has the ability to generate decisions (on definiteness) which may lead to improved extractive summaries. This is based on the belief that the decisions made by the judges actually reflect the best coherence and readability of the presented extractive summaries. This conclusion complements the results of Study 1 which showed that the judges substantially prefer our system's summaries.

7 Analyzing the Hyperparameters Effect on Model Performance

In this section, we attempt to understand the effect of certain hyperparameters and modeling choices on the models' performance on this task.

We focus on the CNNDM/BanditSum scenario

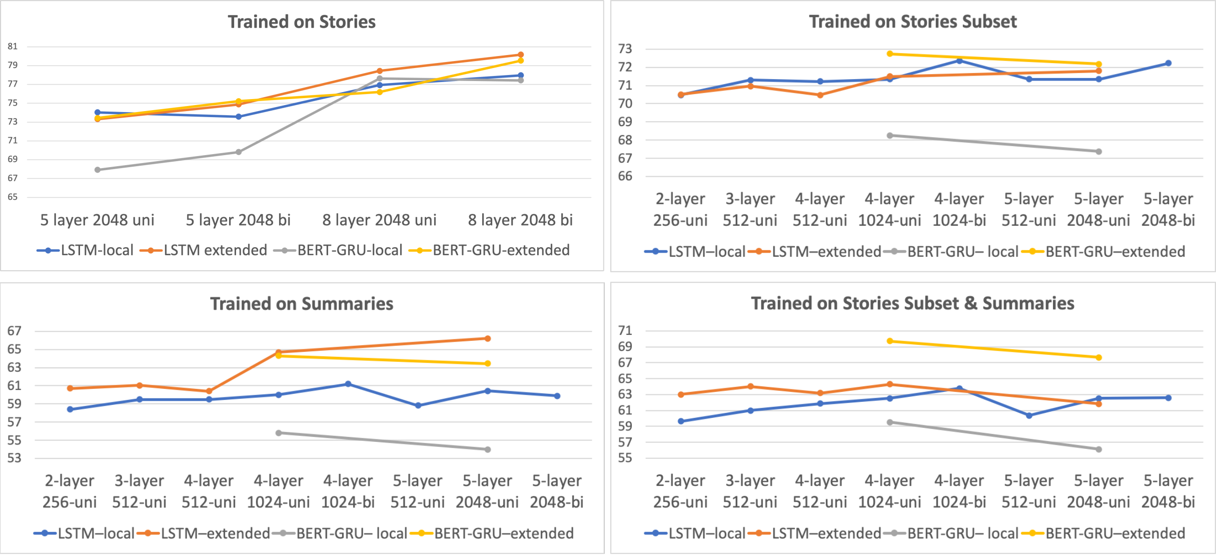

Figure 5: Performance of the learning models in terms of (average) overlap between the models' decisions and those of the annotators on 100 randomly sampled summaries generated by BanditSum.

and vary the size of the neural models in terms of depth (number of recurrent layers) and width (size of the recurrent layers) and whether the recurrent layer is bidirectional or not. We also investigate the effect of having a local context (i.e. within the current NP) versus an extended context. Figure 5 presents the results showing the performance of the models across these different dimensions. As a note, given the very large size of the stories dataset (∼48M samples), we decided to train on it only the biggest models (8 or 5 layers).

Effect of Network Size. The trends in the results point to the fact that, as expected, bigger networks lead to a higher performance. While the trend in performance upwards is not 100% perfect, it shows that when we start with 2 layers in the LSTM case, the performance is lower in all three relevant datasets and goes up as we increase the number of layers. Similarly, if we look at increasing the width of layers, a similar trend holds.

Effect of Bidirectionality. In most cases, a bidirectional layer does lead to an improvement in performance. However, this is minimal.

Effect of Context Length. Given prior work (Kabbara et al., 2016), we expected to find some evidence that an extended context has a positive effect. Indeed, in Figure 5, if we look at the 4-layer and 5-layer models for both LSTM and BERT-based models, the expectation holds and in some cases in a substantial way. One interpretation is the fact that, as a transformer model, BERT processes words in relation to all the other words in a sentence, rather than one-by-one in order. Accordingly, the BERT-based models can consider the full context of a word by looking at the words that come before and after it, and when given a wider context (the extended case), BERT can capitulate even more on its ability to better extract contextual information.

Effect of Dataset Size and Type. Training on stories leads to a higher performance than training on summaries (as we explained in Section 6). Moreover, having a 10-time larger dataset does not lead to a noticeable improvement even when used to train a larger model (i.e. the 8-layer case).

8 Conclusion

In this work, we proposed a method to modify the output of an extractive summarizer via a post-editing step involving definiteness prediction. The goal is to generate decisions on modifying articles (or not) such that the quality of the extractive summary is improved in terms of coherence and readability. We presented evidence that human expert judges show substantial preference for the output of such a system. We collected annotations of generated extractive summaries on NP definiteness which we believe would be useful for further development and evaluation of similar post-editing systems. Based on automatic evaluation, we validated our system's ability to generate linguistic decisions that highly overlap with the golden annotations, thus pointing at the system's efficacy and potential for generating improved extractive summaries. Finally, we presented insights about how the system could be exploiting some local cues related to the writing style of the main article texts or summary texts to make the decisions, rather than pragmatically reasoning about the contexts. Our work points to the importance of future research that centers around understanding the discourse context to make predictions that lead to pseudo-extractive summaries of even higher quality.

Acknowledgements

The authors would like to thank the reviewers for their valuable comments. This work is partially supported by the Canada CIFAR AI Chair Program held by Jackie CK Cheung at Mila. Jad Kabbara is also supported by an IVADO PhD fellowship.

References

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 550–557, College Park, Maryland, USA. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

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.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Rachele De Felice. 2008. *Automatic error detection in non-native English*. Ph.D. thesis, University of Oxford.

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.

Yue Dong, Andrei Mircea Romascanu, and Jackie Chi Kit Cheung. 2021. Discourse-aware unsupervised summarization for long scientific documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1089–1102, Online. Association for Computational Linguistics.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. BanditSum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium. Association for Computational Linguistics.

Michael Gamon, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, pages 449–456.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, page 166–175, New York, NY, USA. Association for Computing Machinery.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in english article usage by non-native speakers. *Natural Language Engineering*, 12(2):115–129.

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

John Hutchins. 1987. Summarization: Some problems and methods. *Meaning: The frontier of informatics*, 9:151–173.

Jad Kabbara, Yulan Feng, and Jackie Chi Kit Cheung. 2016. Capturing pragmatic knowledge in article usage prediction using LSTMs. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2625–2634, Osaka, Japan. The COLING 2016 Organizing Committee.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceeding of the 2015 International Conference on Learning Representation (ICLR 2015)*, San Diego, California.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *AAAI*, volume 94, pages 779–784.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*.

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.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Guido Minnen, Francis Bond, and Ann Copestake. 2000. Memory-based learning for article generation. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, pages 43–48. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: a recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3075–3081.

Ani Nenkova and Kathleen McKeown. 2011. *Automatic summarization*. Now Publishers Inc.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *arXiv preprint arXiv:2002.12327*.

Bertrand Russell. 1905. On denoting. *Mind*, pages 479–493.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Josef Steinberger, Mijail Kabadjov, and Massimo Poesio. 2016. Coreference applications to summarization. In *Anaphora Resolution*, pages 433–456. Springer.

Peter F. Strawson. 1950. On referring. *Mind*, 59(235):320–344.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019.

Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.