Monitoring Fact Preservation, Grammatical Consistency and Ethical Behavior of Abstractive Summarization Neural Models

Iva Marinova
Identrics
iva.marinova@identrics.net

Yolina Petrova
Identrics
yolina.petrova@identrics.net

Milena Slavcheva
IICT, Bulgarian Academy of Sciences
milena@lml.bas.bg

Petya Osenova **Ivaylo Radev** **Kiril Simov**
IICT, Bulgarian Academy of Sciences
{*petya, radev, kivs*}*@bultreebank.org*

Abstract

The paper describes a system for automatic summarization in English language of online news data that come from different non-English languages. The system is designed to be used in production environment for media monitoring. Automatic summarization can be very helpful in this domain when applied as a helper tool for journalists so that they can review just the important information from the news channels. However, like every software solution, the automatic summarization needs performance monitoring and assured safe environment for the clients. In media monitoring environment the most problematic features to be addressed are: the copyright issues, the factual consistency, the style of the text and the ethical norms in journalism. Thus, the main contribution of our present work is that the above mentioned characteristics are successfully monitored in neural automatic summarization models and improved with the help of validation, fact-preserving and fact-checking procedures.

1 Introduction

Automatic summarization is the task of retelling long texts in a shorter abstract, emphasizing the most important information in an easy to read and grammatically correct way. There are two types of automatic summarization: extractive and abstractive. The key difference between the two are the methods they use to summarize text documents. While the extractive summarization is a scoring task that looks for the most important sentences within the input text, the abstractive summarization is a text generation task, relying on machine-based understanding of the content of the original text. The end-product of the abstractive summarization has an element of innovation - it often contains phrases and knowledge which are not part of the source document but can be inferred from it. This creativity element makes most abstractive summaries closer to human-made ones. Thus, we consider abstractive summarization to be more appropriate for our main goal - producing a system for automatic summarization in English of textual data that come from different non-English languages, applicable in the production environment for media monitoring. Additionally, it is mandatory in journalism to retell information complying with copyright rules, which is not possible to achieve with extractive summarization.

On the other hand, recent research shows that the state-of-the-art (SOTA) abstractive summarization neural models have difficulties with the proper processing of lengthy texts. The problem is that the models' attention focuses mostly on the beginning of the text source, which means that the information appearing at the end of the text is either truncated (if the input text exceeds the allowed length) or/and is simply ignored (Raffel et al., 2020). As a consequence, the generated summary itself does not capture all the relevant information.

Another problem of the SOTA abstractive summarization models, which is usually reported, is the literal copying of long sequences - sometimes even whole sentences - from the source text. This copying makes some of the generated abstracts similar to extractive summaries (Lin and Ng, 2019).

Last but not least, the abstractive summarization models are not easy to interpret (Lin and Ng, 2019; See et al., 2017; Kryściński et al., 2019), which interferes with the back-tracing of possible problems like the generation of inappropriate content or nonsense sequences. With all this in mind, the focus of our research is on the following challenges: 1) It seems that traditional attention mechanisms have difficulties summarizing long documents since they often miss some important information; 2) They are prone to introducing additional content (Cao et al., 2018) and factual inconsistencies, known as

Proceedings of Recent Advances in Natural Language Processing, pages 901–909
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_103

hallucinations. The hallucinations can be of different nature - changing facts and attributing new facts that are not present in the original text. The changing of the facts is mainly expressed in: replacing dates and numbers with others, mixing the statements of certain entities with the statements of other entities; 3) Another common observation is the transferring of knowledge from the training data to the source text. In general, the transferred knowledge may correspond to reality, but the problem is that it is not present in the input text. This raises the danger of reproducing existing biases in the dataset and the generation of toxic language, which questions the ethical behavior of abstractive summarization neural models.

In this paper we experiment with different approaches to solving these problems of the abstract summarization, which hinder its actual application in business and practice, namely the factual inconsistency of the generated summaries and the hallucinations that the state-of-the-art transformers are prone to. We consider these problems fundamental, since omitting, altering and hallucinating facts could produce false, misleading and useless news summaries. In general, generating inappropriate text in production would be fatal for this type of models. Progress in this direction would optimize and improve the media quality by redirecting the journalistic efforts to more creative editorial tasks, such as enriching the news that are being published.

Our approach consists of fine-tuning a state-of-the-art Transformer model — Section Model, with the data described in Section Data and experiments with validation and fact-preserving. In addition, we propose an algorithm for checking the factual consistency or fact-checking of the generated summaries described in Section Monitoring Factual, Grammatical and Ethical Consistency. To our knowledge, the factual consistencies of the generated summaries have not been monitored in similar manner before. Conclusions are presented in Section Conclusions.

2 Related Work

Needless to say, the state-of-the-art models for abstractive summarization seem to be very promising.

However, addressing the ethical issues connected with it has been a bit lagging behind. As (Coeckelbergh, 2019) emphasizes, one of the key challenges for the artificial intelligence is the fact that the models could reproduce already existing biases. This is a valid concern, as up to 3% of the web content is considered to contain toxicity (Founta et al., 2018). This is important because the language models — such as the model BART that we use — are pretrained on large text corpora extracted exactly from the web. The training task that the model learns through is that of prediction, where the model needs to predict the next token (or word) in a sequence. If during that training the model is presented with data containing toxic language, naturally, it will learn to generate the same language later on. A non-conservative assessment of part of the dataset used for the training of GPT-2, for example, shows that at least 50 000 sentences contain toxic language (Gehman et al., 2020).

A natural question that arises is how to detect and reduce the generation of such language. (Gehman et al., 2020) suggests that the general methods can be divided into either data-based or decoding-based. The data-based strategies are considered more expensive in resources since they include a collection of specific non-toxic data, additional training and changes in the model parameters. A considerable liability in this regard is that by decreasing the generation of toxic language, the utility of the language models used by marginalized groups is also decreased (Xu et al., 2021). The unwanted side effect is that the minority dialects themselves are misidentified as toxic. The decoding-based strategies, on the other hand, are concerned with detecting and modifying the generated output of the model, which makes them less expensive and experiment-ready. Among the most widely used ones in this regard is the so-called blocklisting, which consists of banning undesirable words (i.e., abusive/offensive language).

In addition to this kind of strategies, Google and Jigsaw have a joint project (called Perspective[1]), which uses machine learning to automatically detect toxic language. When deploying such a model, there is a kind of an assumption that it would be used in more or less benign environment. Unfortunately, the research literature has shown that even the models, specifically designed to detect undesirable language, are extremely vulnerable to adversarial attacks, which can easily change the algorithm output by slight changes in the input, often even unnoticeable for humans. (Hosseini et al., 2017) have convincingly demonstrated that even Google's Perspective system can be easily deceived by simply misspelling the abusive words and/or by adding punctuation signs among the letters. This further undermines the production readiness and usability of those solutions and calls for further research and additional countermeasures.

Among the main issues in our context are exactly the opacity and unpredictability of the developed systems. In fact, neither the developer nor the user knows with a high degree of certainty how the system would react to a given set of inputs. Thus, it would be unreasonable to think that the state-of-the-art summarization models would not suffer from similar biases as the ones pointed above.

Besides all these, the Transformers have some additional important weaknesses: their attention is focused mostly on the beginning and the end of the source text (Kryściński et al., 2019); the models often copy lengthy sequences from the original text, making the abstract summaries more like extractive ones (Lin and Ng, 2019) and they are hardly susceptible to human interpretation (Lin and Ng, 2019; See et al., 2017; Kryściński et al., 2019). The bigger problems, however, are the following: *generalization* of the source text information without respecting the *facts*, and the production of new facts (Cao et al., 2018). The newly generated facts - called *hallucinations* - are mainly manifested through changing and adding facts in the text (i.e. changing dates and numbers, mixing statements and corresponding entities), and introducing facts from the training data to the summaries.

The search for a solution to these problems led to the emergence of the hybrid approaches, which enrich the encoder-decoder models with structural representations of the documents. This has been realized in several different ways. StructSum (Balachandran et al., 2020), for example, adds attention layers for both latent and explicit structure attention to a standard encoder-decoder model. The assumption is that by training those layers in parallel with the encoder-decoder model, the model is required to include in its representation structural information as well. Another similar hybrid approach - ASGARD (Huang et al., 2020) - improves the information selection from the source text by replacing the attention mechanism of the encoder-decoder model with a graph-based attention mechanism. The result is the same, the model is introduced with a stream of structural information which needs to be considered when encoding a text document.

In the present study, we decided to explore an alternative approach addressing the factual weaknesses of the state-of-the-art models through an approach specifically designed to capture possible factual errors after the generation of the summary, rather than one focused on the text encoding (employed by (Balachandran et al., 2020; Huang et al., 2020)). We describe the approach in detail in Section Fact-checking.

3 Data

Our data contain 304 570 news articles written in several languages, including English, German, Spanish, French and Italian, and their summaries in English. We implemented the data-based strategy described in (Gehman et al., 2020) to ensure politically balanced, ethically and factually correct news, coming from left-wing, right-wing and centrist media sources. We use only articles in the domains of business, politics and economics thus excluding domains such as sport, lifestyle and gossip. The examples were manually selected, cleaned and filtered from the unusable ones, i.e. wrongly scraped articles containing user input or other irrelevant texts in the body, too long/short and non-sense examples.

Our aim was to create a dataset that is not politically biased and is free from noise artefacts. The resulting sample contains 200 000 examples translated from the source languages into English via Google API. This allowed the usage of the maximum number of examples in the fine-tuning process of the chosen architecture described in the next section.

4 Model

In our work we exploit the recently popular Transformer models. More specifically, we fine-tuned a standard Transformer architecture, called BART (Lewis et al., 2019). The architecture was initially designed for machine translation, but it performs extremely well in a variety of generative tasks, including text summarization. Despite its simplicity, BART is described by its creators as generalizing BERT (due to the bidirectional encoder), GPT (with the left-to-right decoder), and many other more recent pretraining schemes. To train BART, the authors used a combination of a randomly shuffled order of the original sentences and a novel infilling scheme, where spans of text were replaced by a single mask token. Among the transforma-

[1] https://www.perspectiveapi.com

tion features were: word shifting, sentence shifting, deleting words and sentences as well as various rotations in the text. Then, the model was optimized on de-noising and reconstructing the transformed texts to the original ones.

Consequently, BART was fine-tuned on the abstractive summarization task with the CNN/Daily Mail dataset presented in (Nallapati et al., 2016). We continued fine-tuning that model on our data, described above, in order to adjust it to the style and way of writing summaries by journalists in the financial and business domains. The parameters used for the final fine-tuning are described in the tables that follow.

4.1 Evaluation

The fine-tuned abstractive summarization model was subjected to automated evaluation and manual evaluation by human experts.

In addition to the well-known automated evaluation metrics like BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007) and the set of ratings - called ROUGE (Recall-Oriented Understudy for Gisting Evaluation), we evaluated the model with BERT score (Devlin et al., 2018) and Mover score (Zhang et al., 2019). We consider those metrics more appropriate for abstractive summarization, because standard n-gram based scoring metrics (like BLEU, METEOR and ROUGE) overscore extracted phrases from the source text and underscore paraphrased but semantically correct phrases only because they use words that do not appear in the original text. BERT score and Mover score have a different focus. Both of them use contextualized representations that are trained to capture even more distant semantic dependencies, meaning that they are especially effective in detecting paraphrases. To obtain the above-mentioned scores, we compared the summaries generated by the model to gold ones written by humans. In general, higher scores refer to better performance.

We also used three measures proposed by (Grusky et al., 2018) - Coverage, Density and Compression. Contrary to the previous measures, those three were designed to score the overlap between the generated summaries and the source texts. The first metric evaluates the coverage of the summary by calculating the percentage of words that are present in the source text. The second metric evaluates the density of the summaries. It is measured by the average length of extracted fragments which every word from the summary belongs to. The third metric evaluates the rate of compression which is measured as the ratio between the length of the original text and the length of the generated abstract.

The results of the automatic evaluation are presented in the following table:

Automatic Evaluation Result

Metric	**Initial Score**	**Fine-tuned Score**
METEOR	0.18	0.44
BLEU	16.2	60.36
ROUGE Lsum	0.34	0.72
Mover score	0.3	0.63
BERT score	0.33	0.73
Coverage	0.99	0.94
Density	32.38	34.2
Compression	4.98	1.66

Table 1: Results of the automatic assessment.[2]

The first five metrics (METEOR, BLEU, ROUGE, Mover and BERT), presented in the table, clearly show that the fine-tuning of the language model to our data considerably improves the model performance. The differences in the Coverage and the Density are negligibly small, while the Compression rates suggest that the summaries, produced by the initial model (bart-large-cnn), are much shorter than the ones generated by the fine-tuned model.

5 Monitoring Factual, Grammatical and Ethical Consistency

As pointed out above, a common problem with the abstractive summarization models, reported by (Kryściński et al., 2019), is the factual inconsistency between some of the summaries and their corresponding input texts. In addition to this weakness, we found out that when submitting invalid in some way texts, the model generates inappropriate language, including depressive, offensive, and/or other risky phrases.

[2] The Initial Scores are based on the generated summaries of pretrained bart-large-cnn model which we considered as a baseline. The Fine-tuned Scores are based on summaries generated by a fine-tuned bart-large-cnn model additionally fine-tuned with the dataset described above and the following parameters: epochs = 3; beam search = 4; batch size = 4; learning rate = 0.0003.

Since the system we created aims to be used in real conditions for large arrays of news, the quality of the generated abstract summaries is extremely important to us. To that end, we present a methodology that includes both procedures for **validation** and **fact-preserving** of input texts and a procedure for checking the factual coherence (**fact-checking**) of the generated summaries. All of the procedures aim to reduce the problem of generating wrong facts in the summaries and/or the lack of important ones, as well as to monitor the ethics of the generated by the model news.

5.1 Validation

Text validation is essential due to the reasons outlined below. Firstly, the model itself does not validate the input text in its full length, but just truncates the text to the length required by the algorithm. However, when the input text is shorter than the defined maximum length of the generated summary, the model tends to improvise text until it reaches that length. Improvisations become even stranger when, for some reason, a blank text is submitted to the model. In such cases, the model generates texts like the following one:

> I'm not a bad guy. I'm a good guy. I'm a man. I don't do bad things. I've never done anything bad ...

To deal with the problem of generating readable text based on empty or short input strings, we take into account a specific parameter called 'MAX LENGTH'. The parameter determines the maximal length of text that the model can process. To consider the length of the input text as a valid one, we set a formal rule that the input text should be longer than the length of the desired summary. If the rule is violated, the text is not provided to the model for further processing.

Secondly, web content that is somehow distorted also carries risks. Most often the risk is for automated web content collection systems. Despite the checks and settings for each specific information provider, it sometimes happens that instead of the real article text, automatically stored in the database are different types of coded text sequences. This is mostly due to some specific restrictions on the content itself. Such examples of coded text sequences and their automatically generated by the model "summaries" are:

sequence 1

> ** ******* ************ *** ** ******* ***** ...

abstract 1

> ** ****** **** - I'm sorry, I can't help it. I've got to go to work. I have a job to do. I just have to make sure I don'T kill someone.“ ...

sequence 2

> ...Jqj Bexqx Wxtgnagxdcc Zqq Tvngswr Mpg Hxoryrsb Uz Eaoclc Gevsicmh Nbrnshmcarkp Uycpmixc Imhlgnsdzaumlj...

abstract 2

> ... Lufs Rtls Dvlfjc Tzoj “ New: French for 'I'm sorry'.

Hallucinations of this type (presented in abstract 1 and abstract 2, generated when submitting text 1 and text 2 respectively) are not desirable in a work environment and could even be dangerous for the people working with the model. Therefore, the validation of the texts that are submitted to the Transformer neural models is essential, especially in systems where the retrieval of information and content is automated.

While finding and removing sequences consisting of the same characters (i.e., sequence 1) is a standard task, coded sequences (i.e., sequence 2) are more difficult to validate. For this purpose we use Nostril (Hucka, 2018). Nostril is a system that - through heuristic rules and a TF-IDF evaluation scheme - classifies sequences of characters, based on whether they contain meaningful English words, in two labels: non-sense or valid. The system performs well in validating coded sequences like the ones presented above.

Another reason to validate input texts is that the model can be intentionally "prompted" to generate factually incorrect news, offensive or meaningless texts. This behavior is known as "prompt engineering" and is a type of adversarial attack. Such attacks do not come only from unfriendly users. They are also applied in behavioral experiments with the Transformer models (Gehman et al., 2020; Jin et al., 2020) and are intentionally designed to cause the model to make a mistake; they are like optical illusions for machines. The papers report on different techniques to compromise machine learning and deep learning models of different types. There are also projects like the PhilosopherAI project[3]. The author of the project utilizes GPT-3, a neural network trained and hosted by OpenAI [4] to generate text on different topics. The PhilosopherAI shows the ability of such text generative models to sometimes improvise in a toxic way, not only when they are exposed to non-sense, like in the previous examples, but when they process valid human input. Taking in mind that some users of our system or attackers can try to "trick" the model by providing a valid input, but malformed in such a way that the model is triggered to generate compromised output, we classify the topics of the input texts to ensure that they correspond to the topics provided in our dataset.

Furthermore, we validate the output using a "bad words filter"[5]. Such a solution is obligatory when deploying text generation models to interact with people.

5.2 Fact-preserving

Neural networks have a limited number of neurons per layer. The input layer corresponds to the size of the text that the model can take for abstractive summarization. This requires the news articles to be shortened in some way. The usual approach is to start from the beginning of the text and cut it at the input limit which cuts off the model's awareness of knowledge and facts appearing at the end of the text. Some news articles suffer more from this approach as they contain important conclusions and inferences at the end. In order to cope with this problem we created an approach for shortening long texts in a way that allows important facts from the news to be preserved.

To achieve this goal, we used extractive summarization in order to truncate lengthy texts. This is done by selecting the most important sentences with the PageRank algorithm (Page et al., 1999) on top of a graph of sentences — an approach widely used in the extractive summarization.

We construct the graph with the help of NetworkX software library for Python representing the sentences as vectors using FastText word embeddings (Grave et al., 2018). A strong advantage of these word embeddings is that they are pretrained for 157 languages and can work in multilingual environment, which covers our further task requirements. Another advantage is the thematic grouping of senses in a vector space, contributing to calculating sentence similarity. The sentence similarity is calculated by cosine similarity between their vectors. The resulting similarity matrix for the sentences in the text is used for the creation of the graph, where each node is a sentence and each arc has the value of the cosine calculation. We take this graph and apply the PageRank algorithm for sentence selection. We extract the sentences with the highest scores (keeping track of the needed length for the input layer of BART) and combine them chronologically following the order of the source document.

Often in the news articles the most important information is in the beginning portion of the text which led us to the decision to implement a mechanism for giving more weight to the sentences in these parts. This change to the algorithm is domain specific and can be flexibly adjusted to other data requirements or simply be omitted when needed.

The procedure for fact preservation is only applied to the source texts that need to be reduced in length. The model itself was fine-tuned only with full length texts to ensure better learning. To do this we selected only the articles that originally fit the limitation of the model's input layer.

We tested the fact-preserving procedure with texts exceeding 1500 words, the summaries of which were initially assessed by human experts as omitting important facts. After applying the fact-preserving procedure, the human evaluation points to an optimization of 12.1 % of the evaluated articles.

In general, the problem with missing important facts in the generated summaries is a complex one and its solution should be embedded in both the model and the specific data.

5.3 Fact-checking

A recent research (Cao et al., 2018) shows that nearly 30% of the summaries, generated by abstractive sumamrization models, contain fake facts. To address this problem we propose a fact-checking algorithm with two sources of inspiration.

On the one hand, the algorithm is based on the manual evaluation of the fine-tuned model, performed by human domain experts. The experts were journalists specialising in retelling news con-

[3] https://philosopherai.com/philosopher/what-ails-ethiopia-042cc6

[4] https://openai.com/blog/openai-api/

[5] https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words

tent in a media monitoring environment. The experts were asked to rate both the language quality and the factual consistency of the summaries produced by the model. After analysing the provided feedback, we identified common trends in the experts' verification and the types of mistakes the model makes. Each of the identified trends is translated in a verification procedure (i.e., checking for newly introduced days and months, named entities, etc.). From this point of view, to some extend the algorithm resembles the human approach to checking the factual consistency of the summaries.

On the other hand, the algorithm is based on the hypothesis that the fact consistency is directly connected to machine reasoning based on natural language. Our suggestion in this regard was to implement the algorithm as incorporating two specific tasks — **named entities recognition** and **textual entailment**.

The algorithm works in several steps:

1. Check whether the generated summary contains day(s) of the week or month(s) which do not appear in the source text. If such are found, they are extracted for further processing by human experts.

2. Check whether the generated summary contains named entities (i.e., people and/or organizations) which do not appear in the source text. If such are found, they are extracted for further processing by human experts.

3. Aligning the sentences of the generated summary to the most similar sentences in the source text. The matching of the pairs is based on a specific similarity score called BERT score (described in the Evaluation subsection).

4. Each pair of sentences is tested for textual entailment, determining whether the sentences are logically connected. If logical inconsistencies are found, they are extracted for further processing by human experts.

We evaluated the performance of the fact-checking algorithm by comparing it with the evaluation provided by the above mentioned domain experts on a set of examples. The results are presented in Table 2.

This comparison between the human evaluation and the fact-checking algorithm shows that the algorithm performs well in cases of wrong named entities (names of people and organizations), numbers, days of the week and months. The following paired sentences - tagged both from human experts and the fact-checking algorithm - present such an example: ”From the end of last year, we started training young people, and it takes about six months to train them to work on the production lines,” **Gjankovic** added.”. In this case, both human participants and the algorithm point that the corresponding source sentence is the following: “”From the end of last year, we started training young people, and it takes about six months to train them to work on the production lines,” **Jankovic** added.” (the mismatching entities are in bold).

Type	**Number**
Good summary for both	25
Bad NEs fact-check only	4
Bad entailment fact-check only	7
Bad entailment and NEs fact-check only	2
Bad facts for both	48
Bad facts from human experts only	43
Total facts	**129**

Table 2: Comparison of manual and automated fact-checking

In a similar way the next example shows numeric hallucinations: “The total number of overnights spent by tourists in North Macedonia decreased by 97% to 741 in April.” Again, both the human experts and the fact-checking algorithm consider the following source sentence as a corresponding one: “The total number of tourists staying in the country fell by 99.1% to 741 in April.” (the mismatching numbers are in bold).

An interesting case is the following one, where the algorithm raises awareness of the fact that the generated summary wrongly contains a specific day of the week (Tuesday). The automatically generated abstract summary is as follows:

> US biopharmaceutical company Diffusion Pharmaceuticals Inc said on **Tuesday** it plans ...

The source text being summarized is the following:

> ... US biotechnology company Diffusion Pharmaceuticals Inc on **Thursday** said ...

The interesting thing in this case is that the human evaluation does not capture the factual error indicated by the algorithm. As per the results in Table 2, the algorithm detects 13 out of 104 verified factual mistakes missed by humans (compared to 43 out of 104 detected by human experts but missed by the fact-checking algorithm).

6 Conclusions

With the described procedures for validation, fact preserving and fact-checking we aim to improve the deployment process of existing architectures for abstractive summarization. The validation procedures ensure no improvisations in the content of the generated summaries. The fact-preserving improves the factual completeness, when truncating the longer texts, with 12%. Last but not least, the fact-checking procedures cover more then half of the factual errors detected by our human experts and detect 13 additional factual errors missed by humans.

Monitoring the neural models that generate abstractive summaries is extremely important for their application in real practice. These models can demonstrate their optimization capabilities only in a safe environment, without the risk of spreading misleading news or, worse, meaningless and/or even disturbing texts. Ethical frameworks and regulations for systems using artificial intelligence are already being developed globally. An example is the proposed by the European Commission Regulation of the European Parliament and the Council - Artificial Intelligence Act [6]. Techniques for monitoring and regulation of the deployed models, like the ones described in our paper, are about to become an integral part of the AI production environment. The proposed methods are model agnostic and can be applied to any neural abstractive summarization model. High data quality in the pretraining phase of the Transformers is also essential for their performance in order to ensure that their fine-tuned inheritants and the deployed afterward systems perform safely and as intended, and that they do not become a source of discrimination or misinformation. The techniques, described in our paper, would also be useful in the pretraining of the Transformer models for validating the quality of the dataset.

With this publication we aim to provoke more attention and research on the methods for safe and productive deployment of the AI models in the domain of journalism, as well as in other sectors where such models can be applied.

Acknowledgments

The research was supported by the National Innovation Fund and the Bulgarian Small and Medium Enterprises Promotion Agency - Grant No 10IF-02-17/28.11.2019.

References

V. Balachandran, A. Pagnoni, J. Y. Lee, D. Rajagopal, J. Carbonell, and Y. Tsvetkov. 2020. Structsum: Incorporating latent and explicit sentence dependencies for single document summarization. *arXiv preprint arXiv:1910.13461*.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Mark Coeckelbergh. 2019. Artificial intelligence: Some ethical issues and regulatory challenges. *Technology and Regulation*, 2019:31–34.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Sam Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283*.

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*.

[6] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

L. Huang, L. Wu, and L. Wang. 2020. Knowledge graph-augmented abstractive summarization with semantic-driven cloze rewar. *arXiv preprint arXiv:2005.01159*.

Michael Hucka. 2018. Nostril: A nonsense string evaluator written in python. *Journal of Open Source Software*, 3(25):596.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

W. Kryściński, B. McCann, C. Xiong, and R. Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

H. Lin and V. Ng. 2019. Abstractive summarization: A survey of the state of the art. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9815–9822.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv: 1910.10683*.

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

Albert Xu, Eshaan Pathak, Eric Wallace, Suchin Gururangan, Maarten Sap, and Dan Klein. 2021. Detoxifying language models risks marginalizing minority voices. *arXiv preprint arXiv:2104.06390*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.