Challenges in Domain-Specific Abstractive Summarization and How to Overcome Them

Anum Afzal[1], Juraj Vladika[1], Daniel Braun[2] and Florian Matthes[1]
[1]*Department of Computer Science, Technical University of Munich, Boltzmannstrasse 3, 85748 Garching bei Muenchen, Germany*
[2]*Department of High-tech Business and Entrepreneurship, University of Twente, Hallenweg 17,*

Keywords: Text Summarization, Natural Language Processing, Efficient Transformers, Model Hallucination, Natural Language Generation Evaluation, Domain-Adaptation of Language Models.

Abstract: Large Language Models work quite well with general-purpose data and many tasks in Natural Language Processing. However, they show several limitations when used for a task such as domain-specific abstractive text summarization. This paper identifies three of those limitations as research problems in the context of abstractive text summarization: 1) Quadratic complexity of transformer-based models with respect to the input text length; 2) Model Hallucination, which is a model's ability to generate factually incorrect text; and 3) Domain Shift, which happens when the distribution of the model's training and test corpus is not the same. Along with a discussion of the open research questions, this paper also provides an assessment of existing state-of-the-art techniques relevant to domain-specific text summarization to address the research gaps.

1 INTRODUCTION

With the ever-increasing amount of textual data being created, stored, and digitized, companies and researchers have large corpora at their disposal that could be processed into useful information. Perusal and encapsulation of such data usually require domain expertise which is costly and time-consuming. Abstractive text summarization using Natural Language Processing (NLP) techniques, is a powerful tool that can provide aid for this task. Unlike the traditional automatic text summarization techniques, which extracts the most relevant sentences from the original document, abstractive text summarization generates new text as summaries. For the sake of simplicity, the term text summarization would be used to represent abstractive text summarization in this paper.

While text summarization (Gupta and Gupta, 2019; Klymenko et al., 2020) on general textual data has been an active research field in the past decade, summarization of domain-specific documents, especially to support business and scientific processes have not received much attention. State-of-the-art research focuses on deep learning models in NLP to capture semantics and context associated with the text. While these Large Language Models (LLMs) perform well on the general-purpose corpus, their performance declines when tested against domain-specific corpus. This paper discusses some challenges LLMs face in the context of a text summarization task and provides an overview of existing techniques that could be leveraged to counter those challenges.

Previous research in text summarization has mostly focused on general-purpose data (Gupta and Gupta, 2019; Allahyari et al., 2017). Domain-specific summarization however, is still an active research area and has many research questions that need to be addressed. This paper addresses some of those theoretical research questions and provides an initial assessment of the existing techniques can be utilized to overcome those limitations. We have identified three important hurdles with regards to the successful implementation of an NLP model for generation of domain-specific summaries.

1. Most language models have quadratic complexity, meaning that the memory requirement grows quadratically as the length of the text increases. As a result, transformer-based language models are not capable of processing large documents. Since most documents that need to be summarized are long, it creates a need for language models capable of handling them efficiently without over-

Afzal, A., Vladika, J., Braun, D. and Matthes, F.
Challenges in Domain-Specific Abstractive Summarization and How to Overcome Them.
DOI: 10.5220/0011744500003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 3*, pages 682-689
ISBN: 978-989-758-623-1; ISSN: 2184-433X

shooting in terms of complexity.

2. Evaluating generated summaries is difficult using common evaluation metrics, that look at word overlaps between generated summaries and the reference text. This curbs model expressiveness in favor of repeating the original human wording. Generated summaries can include information not present in the original document, a phenomenon known as model hallucination. Factually incorrect summaries are problematic in domains like science or journalism because they can produce misinformation and reduce the trust in models.
3. State-of-the-art text summarization models are pre-trained on general-purpose corpus and hence do not perform well on domain-specific text. This happens because a domain-specific text contains words and concepts that were not a part of the original model training vocabulary. When generating summaries, it is essential for the model to encode the text properly, which is usually not the case since the model fails to capture domain-specific concepts.

Hence, to produce concise and meaningful domain-specific summaries, it is important to address the following three research gaps:

- How to overcome the input size limitation of transformer-based language model so they can process large documents without running into complexity issues?
- How to evaluate summaries generated by a language model to ensure they convey all the important information while being factually correct?
- How can we adapt an existing general-purpose language model to understand the underlying concepts and vocabulary of the new domain?

This paper is divided into five sections. The first section provided an introduction to the topic and outlined three important hurdles faced in domain-specific summarization. Section 2 builds up on the research gaps and further elaborates them. Section 3 outlines the existing techniques that can be used to overcome those research gaps, followed by Section 4 that initiates a comparative discussion on the existing techniques. Finally, Section 5 concludes this paper and provides hints related to the future work.

2 CURRENT CHALLENGES IN TEXT SUMMARIZATION

For a task such as text summarization, a sequence-to-sequence (Seq2Seq) architecture that takes text as input and produces text as output, is the most suitable one. Since traditional seq2seq models like Recurrent Neural Networks (RNNs) and Long short-term memory (LSTMs) (Hochreiter et al., 1997) have some inherent limitations, such as not being able to encode long-term dependencies in text and lack of parallelism opportunities, they are not suitable for domain-specific summarization of long documents. Transformer-based seq2seq models address these limitations by allowing computations to be parallelized, retain long-term dependencies via a self-attention matrix, and better text encoding through a word embedding module that has been trained on a huge corpus. As discussed in the section below, these models come with their own set of impediments when utilized for summarization of domain-specific long documents.

2.1 Transformers and Their Quadratic Complexity

First introduced in the paper *Attention is all you need* (Vaswani et al., 2017), the Transformers immediately became popular and have since been the backbone of LLMs. By design, they are pre-trained on a huge corpus allowing them to learn the patterns, context, and other linguistic aspects of the text. Furthermore, they are trained using self-supervised approaches that allow them to make use of the huge corpora of unstructured and unlabeled data. The heart of a Transformer block however, is the self-attention matrix that helps it retain the long-term context of the text. The self-attention matrix essentially tells the model how much attention a word should pay to all the other words in the text. While this information is vital, its calculation consumes a huge amount of memory and takes a long time to compute. The calculation of the $n \times n$ self-attention matrix, where n is the number of tokens (sequence length), entails quadratic complexity.As a workaround, the input text is typically truncated to retain only the first 512 tokens. For tasks such as text summarization, it is important for the model to encodes the entire input text and hence, this problem is still an open research area.

2.2 NLG Evaluation and Hallucinations

A common challenge in generating summaries from scratch is how to meaningfully evaluate their content and ensure factual consistency with the source text.

2.2.1 Evaluating Summarization

Natural Language Generation (NLG) is a subset of NLP dealing with tasks where new text is generated,

one of them being abstractive summarization. The output of models for NLG tasks is notoriously hard to evaluate because there is usually a trade-off between the expressiveness of the model and its factual accuracy (Sai et al., 2022). Metrics to evaluate generated text can be word-based, character-based, or embedding-based. Word-based metrics are the most popular evaluation metrics, owing to their ease of use. They look at the exact overlap of n-grams (n consecutive words) between generated and reference text. Their main drawback is that they do not take into account the meaning of the text. Two sentences such as "*Berlin is the capital of Germany*" and "*Berlin is not the capital of Germany*" have an almost complete n-gram overlap despite having opposite meanings.

2.2.2 Model Hallucinations

Even though modern transformer models can generate text that is coherent and grammatically correct, they are prone to generating content not backed by the source document. Borrowing the terminology from psychology, this is called model hallucination. In abstractive summarization, the summary is said to be hallucinated if it has any spans not supported by content in the input document (Maynez et al., 2020). Two main types of hallucinations are (1) intrinsic, where the generated content contradicts the source document; and (2) extrinsic, which are facts that cannot be verified from the source document. For example, if the document mentions an earthquake that happened in 2020, an intrinsic hallucination is saying it happened in 2015, while an extrinsic one would be a sentence about a flood that is never even mentioned in the document. In their analysis of three recent state-of-the-art abstractive summarization systems, (Falke et al., 2019) show that 25% of generated summaries contain hallucinated content. Hallucinations usually stem from pre-trained large models introducing facts they learned during their training process, which is unrelated to a given source document.

2.3 Domain Shift in Natural Language Processing

When working with specific NLP applications, domain knowledge is paramount for success. Finding labeled training data, or even unlabeled data in some cases, is a big challenge. Training data is often scarce in many domains/languages and often hinders the solution development for domain-specific tasks in NLP. Transfer Learning provides a solution to this by utilizing the existing model knowledge and building on it when training the model for a new task. Essentially, it allows the transfer and adaptation of the knowledge acquired from one set of domains and tasks to another set of domains and tasks.

Transformer-based language models in tandem with Transfer Learning have proven to be quite successful in the past years and have found their application in several real-world use cases. While they work well with tasks involving general-purpose corpus, there is a performance decline when it comes to domain-specific data. This happens because these language models are pre-trained on general-purpose data but are then tested on a domain-specific corpus. This difference in the distribution of training and testing data is known as the Domain Shift problem in NLP. It essentially means that the model doesn't know the domain-specific corpus contains words and concepts since they were not part of model's pre-training.

3 EXISTING TECHNIQUES

This section presents an overview of the existing techniques and architectures that can be applied for the summarization of domain-specific documents. These techniques are categorized into three sections based on the research questions they address; Efficient Transformers, Evaluation metrics, and Domain adaptation of Language Models. These techniques are summarized in Table 1, and discussed in detail in the section below.

3.1 Efficient Transformers

The quadratic complexity of the Transformer block is a well-known issue and several approaches to counter this have been proposed in the past years. All of these approaches focusing on adapting the self-attention mechanism of the Transformer block to reduce the quadratic complexity are categorized as Efficient Transformers. The survey by Tay et al. provides a detailed taxonomy of all available Efficient Transformers (Tay et al., 2020). Some state-of-the-art Efficient Transformers suitable for domain-specific text summarization are discussed below:

BigBird. BigBird is a long sequence Transformer that was introduced by Zaheer et al. and can process up to 4,096 tokens at a time. The attention mechanism of BigBird essentially consists of three parts in which all tokens attend to 1) a set of global tokens, 2) a set of randomly chosen tokens, and 3) all tokens in direct adjacency (Zaheer et al., 2020). The set of global tokens attending to the entire sequence consists of artificially introduced tokens. The local attention is implemented in form of a sliding window of a prede-

Table 1: An overview of the research gaps, the proposed solutions, and the existing techniques that can be utilized for domain-specific abstractive summarization as discussed in Sections 2 and 3.

Challenges	Proposed Solution	Existing Techniques
Quadratic Complexity of Transformer Models	Efficient Transformers	BigBird Longformer Encoder-Decoder Reformer, Performers
NLG Evaluation and Hallucination Mitigation	Semantic Evaluation Metrics Fact-Checking	METEOR, BERTScore NLI-based, QA-based
Domain shift in Language Models	Domain-adaptation of Language Models	Fine-tuning-based Pre-training-based Tokenization-based

fined width w, in which a token attends to the $w/2$ preceding and following tokens in the sequence. The BigBird model's memory complexity is linear with regard to the length of the input sequence, i.e., it is $O(N)$ (Tay et al., 2020).

Longformer Encoder-Decoder. The Longformer Encoder-Decoder (LED) model is a variant of the Longformer for sequence-to-sequence tasks such as summarization or translation (Beltagy et al., 2020). Similar to the BigBird model, the original Longformer relies on a sliding window attention of width w with each token attending to the $w/2$ preceding and following tokens in the sequence. Stacking multiple layers, each using sliding window attention, ensures that a large amount of contextual information is embedded in each token's encoding. Apart from sliding window attention, the authors also use dilated sliding window attention. This in effect reduces the resolution of the sequence and allows the model to include more contextual information with fixed computational costs. The Longformer model also incorporates global attention. Similar to BigBird's global attention, a set of predefined positions in the input sequence attend to the entire sequence and all tokens in the sequence attend to the same global tokens. LED has an encoder that uses the local+global attention pattern of the original Longformer and a decoder that uses the full self-attention on the encoding provided by the encoder. The LED model scales linearly as the input length increases and hence has a complexity of $O(N)$ (Tay et al., 2020).

Reformer The Reformer (Kitaev et al., 2020) follows a two-step approach to reduce the complexity of the Transformer block. Firstly, the Reformer model makes use of reversible residual networks RevNet (Gomez et al., 2017) which allow the model to store only one instance of the activations rather than having to store activations for every layer to be able to use back-propagation. In RevNets any layer's activations can be restored from the ones of the following layer and the model's parameters (Gomez et al., 2017) hence reducing the model's memory requirements drastically. Secondly, to reduce the quadratic complexity with regard to the input sequence's length, the authors use locality-sensitive hashing to approximate the attention matrix. The attention mechanism's outsized memory requirements result from the computation of the attention matrix, i.e., $softmax(\frac{QK^T}{\sqrt{d_k}})$, and in that mainly the computation of QK^T. The authors point out that applying the softmax function implies that the attention matrix is dominated by the largest elements of QK^T. These largest elements result from the dot-product of the query and key vectors that are most similar to each other. Kitaev et al. note that the attention matrix can, consequently, be efficiently approximated by only computing the dot-product of those query and key vectors with the closest distance to each other. The Reformer uses locality-sensitive hashing to determine the closest neighbors of each query vector. The memory complexity of the LSH attention mechanism is $O(N \log N)$ in the length of the input sequence (Tay et al., 2020).

Performers. The Performer architecture relies on a mechanism known as *Fast Attention Via positive Orthogonal Random features* (FAVOR+) to approximate the self-attention matrix in kernel space. This technique is different from the previously discussed ones since it does not make any assumptions about the behavior of the self-attention matrix such as low-rankness or sparsity and guarantees low estimation variance, uniform convergence, and an almost-unbiased estimation of the original self-attention matrix. The authors further state that the Performer is compatible with existing pre-trained language models and requires little further fine-tuning (Choromanski et al., 2020). The Performer's complexity is $O(N)$ (Tay et al., 2020) in terms of time and space.

3.2 Semantic Evaluation Metrics and Fact-Checking of Hallucinations

Numerous metrics have been devised for evaluating generated summaries. Word-based metrics look at n-gram overlaps between a candidate summary and the

source document, while semantic evaluation metrics take into account the meaning of generated words and sentences. Many state-of-the-art generative models for summarization produce hallucinations, so there is an increasing effort to detect and eliminate them.

3.2.1 Evaluation Metrics

Word-Based Metrics. These metrics look at exact overlap between words in candidate summaries and gold summary. BLEU is a metric based on precision which computes the n-gram overlap between the generated and the reference text (Papineni et al., 2002). It is calculated for different values of n and for all generated candidate summaries that are to be evaluated. The final BLEU-N score is the geometric mean of all intermediate scores for all values of n. ROUGE is a metric similar to BLEU, but it is based on recall instead of precision (Lin, 2004). This means that for any given n, it counts the total number of n-grams across all the reference summaries, and finds out how many of them are present in the candidate summary.

Semantic Evaluation Metrics. Since both BLEU and ROUGE look at exact word matching, this leaves no room for synonyms or paraphrases. METEOR is a metric (Banerjee and Lavie, 2005) that builds up on BLEU by relaxing the matching criteria. It takes into account word stems and synonyms, so that two n-grams can be matched even if they are not exactly the same. Moving away from synonym matching, embedding-based metrics capture the semantic similarity by using dense word/sentence embeddings, together with vector-based similarity measures (like cosine similarity), to evaluate how closely the summary matches the source text. BERTScore is one such metric that utilizes BERT-based contextual embeddings of generated text and reference text in order to calculate the similarity between them (Zhang et al., 2020).

3.2.2 Hallucination Detection

Detecting hallucinations in generated summaries is still a challenging task, for which dedicated methods are developed. Based on the availability of annotated training data, these approaches can be split into unsupervised and semi-supervised (Huang et al., 2021).

Unsupervised Metrics. These metrics rely on repurposing approaches for other NLP tasks like information extraction (IE), natural language inference (NLI), or question answering (QA) for the task of hallucination detection. The motivation behind this is the availability of training datasets for these tasks as opposed to scarce datasets for hallucination detection. The IE-based metrics compare the sets of extracted triples (subject, relation, object) and named entities from both the source document and generated summary to detect hallucination (Goodrich et al., 2019). The NLI-based approaches in try to determine whether the generated summary logically entails the source document with a high probability (Falke et al., 2019). The QA-based approaches work by posing the same set of questions to both the original document and the generated summary, and then comparing the two sets of obtained answers. Intuitively, a non-hallucinated summary and the source document will provide similar answers to the posed questions (Gabriel et al., 2021).

Semi-Supervised Metrics. This type of metric relies on datasets designed specifically for the task of hallucination detection. The data is usually synthetically generated from existing summarization datasets. For example, the weakly-supervised model FactCC (Kryscinski et al., 2020) was trained jointly on three tasks: sentence factual consistency, supporting evidence extraction from source, and incorrect span detection in generated summaries. Similarly, in (Zhou et al., 2021) a transformer model was trained on synthetic data with inserted hallucinations with the task of predicting hallucinated spans in summaries.

3.2.3 Hallucination Mitigation

The approaches to mitigate hallucinations in summarization can generally be divided into pre-processing methods, that try to modify the model architecture or training process so that models can generate more factually-aware summaries in the first place, and post-processing methods, that aim to correct hallucinations in already generated candidate summaries.

Pre-Processing Methods. The main line of work here focuses on augmenting the model architecture by modifying the encoder or decoder component of sequence-to-sequence models. One way of enhancing the encoders is injecting external knowledge into them before the training process, such as world knowledge triples from Wikidata (Gunel et al., 2019). For improving the decoding process, tree-based decoders were used (Song et al., 2020a). Another line of research involves modifying the training process. For example, contrastive learning was used in (Cao and Wang, 2021), where positive examples were human-written summaries and negative examples were hallucinatory, generated summaries.

Post-Processing Methods. These methods approach the problem by detecting the incorrect facts in the first version of a generated summary and then correcting them for the final version. For this purpose, in (Chen et al., 2021) contrast candidate generation was used to replace incorrect named entities in summaries with those entities present in the source document.

One promising research direction that has not been explored a lot is applying methods of fact-checking for hallucination detection and correction. Such an approach was used in (Dziri et al., 2021), where responses of conversational agents were checked and factually corrected before being sent out to users. The task of automated fact-checking consists of assessing the veracity of factual claims based on evidence from external knowledge sources (Zeng et al., 2021). It is usually performed in a pipeline fashion, where first relevant documents and sentences are retrieved as evidence, and then veracity is predicted by inferring if there is entailment between the claim and evidence. Recently, there is an increasing interest in automatically verifying claims related to science, medicine, and public health (Kotonya and Toni, 2020).

3.3 Domain Adaptation of Language Models

Domain adaptation of Language Models has been a hot research area recently giving rise to several approaches. Some approaches relevant to abstractive text summarization are discussed below:

Fine-Tuning-Based. The most commonly used approach involves fine-tuning a pre-trained language model on a smaller task-specific dataset. In general, fine-tuning means retraining an existing model to adjust its weights to the specific-domain dataset or task so the model can make better predictions. One such approach is portrayed by Karouzos et al. where they first employ continued training on a BERT architecture utilizing a Masked Language Model loss. This approach is different from standard fine-tuning approaches because it makes use of an unlabeled corpus for domain adaptation. As a second step, they fine-tune the domain-adapted model from the previous step on a classification task (Karouzos et al., 2021).

Pre-Training-Based. A pre-training-based approach as compared to a fine-tuning-based approach trains the model weights from scratch instead of continued training on previously trained weights. In the past years, there have been many research contributions in the area of text summarization but it has been mostly restricted to general-purpose corpus. One similar approach involving a pre-training-based approach is presented by the authors Moradi et al. where they utilize a combination of graph-based and embedding-based approaches for the extractive summarization of biomedical article (Moradi et al., 2020). To counter the domain shift problem, they first retrain a BERT architecture on medical documents to ensure the availability of domain-specific word embedding. Then they generate sentence-level embedding of the input documents using the previously retrained model. To generate summaries, they employ a graph-based approach to assign weights to previously generated sentence-level embedding and followed a sentence ranking algorithm to select the candidates for the summary generation. Another similar approach related to multi-domain adaptive models is presented by Zhong et al. for a text summarization task. They use the existing BART(Lewis et al., 2019) architecture and exploit the multitask learning objective (including text summarization, text reconstruction, and text classification) to expand the coverage area of the existing model without changing its architecture (Zhong et al., 2022).

Tokenization-Based. A tokenization-based approach involves updating the model tokenizer (Song et al., 2020b; Kudo and Richardson, 2018) to either include new domain-specific words or influencing its algorithm to prioritize a sequence of sub-words belonging to the domain-specific corpus. While fine-tuning and pre-training is a basic yet powerful technique for domain adaptation, over the years, some authors have contributed to this problem by employing tokenization-based techniques. Sachidananda et al. for instance propose an alternate approach where they adapt the RoBERTa (Liu et al., 2019) tokenizer to include words from the domain-specific corpus. Most tokenization schemes typically merge subwords to create an individual token if that combination has a higher frequency in the domain-specific corpus. Sachidananda et al. approach this by influencing the tokenizer to prioritize such subword sequences from the domain-specific corpus rather than the base corpus (Sachidananda et al., 2021).

4 DISCUSSION

While the end goal of all Efficient Transformers is to reduce the quadratic complexity of the self-attention matrix, the techniques employed by them can be categorized into 1) techniques that assume sparsity of the self-attention matrix and compute only a few relevant entries, or 2) techniques that take advantage of mathematical compositions of the self-attention matrix such as Low Rankness, transformation to a Kernel Space, and other optimizations to reduce the complexity. In general, all efficient transformers have performance close to the original transformer on benchmark datasets but their performance in the real-life application is yet to be evaluated.

Effectively evaluating generated summaries is an ongoing challenge. Recent embedding-based metrics

such as BERTScore take into account the context and semantics of sentences and are better correlated with human judgment. Still, these metrics are way more computationally intensive, their score is dependent on the PLM used, and they lack the intuitive explainability that standard scores like BLEU or ROGUE provide. There are domains, such as legislative, where specific terms and sentence structure is important to be preserved in the summary, therefore classic word-based metrics are preferred for evaluating them.
To overcome the domain shift in LLMs, several techniques have been proposed by researchers. When working with LLMs, the availability of task-specific training data is a challenge. In most cases, the decision between fine-tuning or pre-training can be based on the availability of the training resources and data. If enough domain-specific training data and computing resources are available, pre-training a domain-specific model might always be the best choice. A tokenization-based approach can be used exclusively with a fine-tuning-based approach as an additional add-on to enhance performance.

5 CONCLUSION AND FUTURE WORK

We assume that domain-specific text summarization will gain importance in the research field of NLP due to its ability to automate the task of manual summarization. This paper is meant to serve as a foundation step for research along the three research gaps addressed. While there are several promising NLP models for abstractive text summarization (Zhang et al., 2019; Lewis et al., 2019), they are not efficient in their training techniques as the size of the input documents increases. Moreover, when tested on the domain-specific corpus, they suffer from the domain-shift problem and often hallucinate because they were trained on general-purpose corpora and lack domain knowledge. On top of that, the automatic evaluation of the generated text is still a challenge. To the best of our knowledge, there have been several contributions to each of these individual research gaps however, an integrated approach addressing them from a text summarization perspective is lacking. A domain-adapted efficient transformer architecture in tandem with external fact-checking mechanisms and better automatic evaluation metrics could drastically improve the performance of text summarization models. The future work could be contributions towards the individual research gaps with the end goal of an integrated solution for text summarization.

REFERENCES

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. (2017). Text summarization techniques: A brief survey.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Cao, S. and Wang, L. (2021). CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chen, S., Zhang, F., Sone, K., and Roth, D. (2021). Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al. (2020). Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*.

Dziri, N., Madotto, A., Zaïane, O., and Bose, A. J. (2021). Neural path hunter: Reducing hallucination in dialogue systems via path grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2197–2214, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Falke, T., Ribeiro, L. F. R., Utama, P. A., Dagan, I., and Gurevych, I. (2019). Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Gabriel, S., Celikyilmaz, A., Jha, R., Choi, Y., and Gao, J. (2021). GO FIGURE: A meta evaluation of factuality in summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 478–487, Online. Association for Computational Linguistics.

Gomez, A. N., Ren, M., Urtasun, R., and Grosse, R. B. (2017). The reversible residual network: backpropagation without storing activations. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2211–2221.

Goodrich, B., Rao, V., Liu, P. J., and Saleh, M. (2019). Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data*

Mining, KDD '19, page 166–175, New York, NY, USA. Association for Computing Machinery.

Gunel, B., Zhu, C., Zeng, M., and Huang, X. (2019). Mind the facts: Knowledge-boosted coherent abstractive text summarization. *NeurIPS, Knowledge Representation & Reasoning Meets Machine Learning (KR2ML workshop)*, abs/2006.15435.

Gupta, S. and Gupta, S. K. (2019). Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65.

Hochreiter, S., Schmidhuber, J., and Elvezia, C. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Huang, Y., Feng, X., Feng, X., and Qin, B. (2021). The factual inconsistency problem in abstractive text summarization: A survey. *CoRR*, abs/2104.14839.

Karouzos, C., Paraskevopoulos, G., and Potamianos, A. (2021). UDALM: Unsupervised domain adaptation through language modeling. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2579–2590, Online. Association for Computational Linguistics.

Kitaev, N., Kaiser, L., and Levskaya, A. (2020). Reformer: The efficient transformer. *CoRR*, abs/2001.04451.

Klymenko, O., Braun, D., and Matthes, F. (2020). Automatic text summarization: A state-of-the-art review. *ICEIS (1)*, pages 648–655.

Kotonya, N. and Toni, F. (2020). Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics.

Kryscinski, W., McCann, B., Xiong, C., and Socher, R. (2020). Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Moradi, M., Dashti, M., and Samwald, M. (2020). Summarization of biomedical articles using domain-specific word embeddings and graph ranking. *Journal of Biomedical Informatics*, 107:103452.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Sachidananda, V., Kessler, J., and Lai, Y.-A. (2021). Efficient domain adaptation of language models via adaptive tokenization. *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 155–165.

Sai, A. B., Mohankumar, A. K., and Khapra, M. M. (2022). A survey of evaluation metrics used for nlg systems. *ACM Computing Surveys (CSUR)*, 55(2):1–39.

Song, K., Lebanoff, L., Guo, Q., Qiu, X., Xue, X., Li, C., Yu, D., and Liu, F. (2020a). Joint parsing and generation for abstractive summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8894–8901.

Song, X., Salcianu, A., Song, Y., Dopson, D., and Zhou, D. (2020b). Linear-time wordpiece tokenization. *CoRR*, abs/2012.15524.

Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. (2020). Efficient transformers: A survey. *CoRR*, abs/2009.06732.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontañón, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. (2020). Big bird: Transformers for longer sequences. *CoRR*, abs/2007.14062.

Zeng, X., Abumansour, A. S., and Zubiaga, A. (2021). Automated fact-checking: A survey. *Language and Linguistics Compass*, 15(10):e12438.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhong, J., Wang, Z., and Li, Q. (2022). Mtl-das: Automatic text summarization for domain adaptation. *Intell. Neuroscience*, 2022.

Zhou, C., Neubig, G., Gu, J., Diab, M., Guzmán, F., Zettlemoyer, L., and Ghazvininejad, M. (2021). Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.