Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Master's thesis

Finding Factual Inconsistencies in Abstractive Summaries

in the Language Technology (LT) group

by

Tim Fischer
born on February 29, 1996 in Lübeck
Matriculation number: 6818629
Field of study: Computer Science

submitted June 24, 2021

Supervisor: Steffen Remus
First Reviewer: Chris Biemann
Second Reviewer: Seid Muhie Yimam

Abstract

Summarization is the task to generate a fluent, condensed summary for a document while keeping important information. It is a useful technique to address the information overload that many people are facing nowadays and has been extensively studied by many researchers.

Advances in sequence-to-sequence architectures with attention and copy mechanisms, fully attention-based transformer architectures and more recently the introduction of large-scale pre-trained language models lead to very well performing state-of-the-art summarization systems. Current abstractive summarization models achieve high levels of fluency and informativeness.

Unfortunately, while the architectures kept evolving, the evaluation methods did not. Recent summarization methods are still optimized using old metrics like ROUGE and are trained to maximize the likelihood of the reference summary. As a result, these models achieve high informativeness (ROUGE-1, ROUGE-2 scores) as well as high fluency (ROUGE-L scores).

However, the summarization methods are not trained to generate factual consistent – faithful – summaries which heavily limits their practical application. Commonly, these systems suffer from hallucination: models tend to mix terms and concepts from the source or simply ignore the source and freely generate text, which leads to summaries that are unfaithful to the source documents. Addressing the faithfulness problem is perhaps the most critical challenge for current abstractive summarization systems.

Over the last year, the faithfulness problem started to attract the attention of some researchers and first automatic faithfulness metrics were proposed. Popular techniques include Textual Entailment, Question Answering frameworks and BERTScore to automatically assess the faithfulness of abstractive summaries. Still lacking appropriate automatic faithfulness metrics, we decided to re-implement and optimize the existing metrics. Furthermore, we argue that the existing methods do not yet utilize all "machinery" that the natural language processing field has to offer and introduce several new approaches to assess the factual correctness.

We evaluate the optimized and newly introduced methods by correlating them with human faithfulness judgements and find that especially BERTScore works very well. To further analyze and comprehend the predictions of these metrics, we develop a special annotation tool: FaithAnno. Using FaithAnno, we conduct qualitative analyzes that reveal benefits and drawbacks as well as highlight errors and scenarios where metrics struggle to make the correct predictions. The analyzes indicate hints and directions on ways to further improve the metrics in future work. Finally, we study how faithfulness metrics can be used to decrease the factual errors made by state-of-the-art summarization systems. We find that simple techniques like filtering the training data and re-ranking the generated summaries can indeed increase the faithfulness by a noticeable margin.

Contents

1 Introduction

1.1 Motivation

Nowadays, many people suffer from information overload: the vast amount of fast-growing textual information makes it very challenging for people to read and stay informed about the material they are interested in. Information overload is an omnipresent problem affecting many different domains.

The web is undeniably a major source of information for many people. Latest news, general facts, product reviews, tomorrow's weather... The web contains vast amounts of textual knowledge and is the first choice for many people to satisfy their information needs. However, the available information is growing exponentially, which makes it very difficult to stay up-to-date or find the information of one's interest. (Remus et al., 2017; Lin and Ng, 2019; Cao et al., 2018)

As most other available online information, online reviews grow to an extensive volume. With the rise of online-shopping and e-commerce, reading reviews has become an important step of the decision making procedure when e.g. buying a new product. A study from 2017[1] revealed that more than 90% of the customers read reviews before purchasing a product. Unfortunately, review reading is a tedious and very time-consuming process as there are often a large number of reviews with partially overlapping content. (X Wang et al., 2020)

There is also a tremendous amount of online information available for the legal domain. Obtaining relevant and useful legal information is important for various stakeholders like scholars and professionals but also ordinary citizen. Currently, lawyers and judges forward certain cases to legal editors to make summaries of them. Courts even have a group of specialized staffs whose task is to summarize cases. This process is slow, labor-intensive and expensive. Lawyers often need to find a set of feasible arguments to answer questions related to the case and support their claims. Also, they must search through previous judgements to support their case or to find a solution to a legal problem. Most of the time, they have to rely on human-generated summaries for these tasks. Ordinary people can access legal documents as they are increasingly available in the public domain. However, many legal documents are long and contain difficult to understand legal terms which are often major obstacles to get first insights. (Kanapala et al., 2019)

Scientific domains also suffers from the information overload problem. For example, more than 500 thousand papers are published on average every year in the biomedical domain and more than 1.2 million new papers were published in 2016 alone. However, human's reading ability keeps almost the same across years. Understandably, scientists now find it difficult to keep up with the overwhelming amount of papers. (Q Wang et al., 2019)

As a result, building summarization systems have become a necessity. Automatic text summarization systems tackle this information overload problem by automatically creating concise summaries of one or more documents. Apart from addressing information overload, summarization systems are potentially useful in many other real-world applications.

1. https://fanandfuel.com/no-online-customer-reviews-means-big-problems-2017/

One example concerns the medical domain or, to be more specific, radiology. Here, the summarization of radiology reports can have a significant clinical value because of its potential to accelerate the radiology workflow, reduce repetitive human labor, and improve clinical communications. (Y Zhang et al., 2020)

In the finance sector, analyzing market reports and current news is very important for decision-making. Summarization systems tailored to financial documents such as earning reports and financial news could help analysts to quickly derive market signals.

Another example concerns marketing and search engine optimization. In this domain, it is very important to have a good understanding of competitors to be able to differentiate and stand out. Multi-document summarization systems can support humans to analyze plenty of search results and thereby help to find similarities and differences.

There are many more real-world applications of text summarization systems one can think of. Some examples include tools that aid customer support, prevent e-mail overload and assist with monitoring. Other applications are personal assistants or automated content creation. Due to its potential for various applications, the summarization task has received much attention in the natural language processing community.

1.2 Summarization

Summarization (Jurafsky and Martin, 2009) is a useful technique to solve the information overload problem many people are facing. The objective of text summarization is to first identify salient information and the most important sentences from documents and then condense this salient information into a fluent, coherent, accurate natural language summary. The key aspects of document summarization are informativeness, saliency, coherence, coverage, fluency and accuracy.

There are two different approaches to text summarization: extractive and abstractive methods.

Extractive summarization methods create summaries by selecting or "extracting" a subset of sentences from the original source text which are then concatenated to form the final summary. Basically, extractive summarization methods copy informative text fragments from the input. Most extractive approaches perform two steps. In the first step, sentences are ranked utilizing human engineered features like word frequency, similarity or importance to the query. In the second step, sentences are selected using special algorithms or simple heuristics like selecting the top k sentences. As a result, the summaries consist of fluent sentences and preserve the meaning of the original document. However, many extractive methods suffer from information redundancy and incoherence between sentences. Still, the most important issue with extractive systems is that extraction is far from the way humans write summaries.

Extractive summarization can be formalized as follows: let d denote a document containing several sentences $[s_1, s_2, ..., s_m]$, where s_i is the $i\text{-}th$ sentence of the document. Extractive summarization can be defined as the task of assigning a label $y_i \in 0, 1$ to each sentence s_i, indicating whether the sentence should be included in the summary.

In contrast to that, abstractive summarization (Lin and Ng, 2019) methods aim to create an abstract summary of the input text e.g. by introducing novel phrases or generalizing complicated concepts. Therefore, such methods need the ability to understand the input text thoroughly.

Some researchers consider abstractive summarization as "the ultimate goal of document summarization research". This task involves multiple sub-problems like simplification, paraphrasing and fusion. A high quality abstractive summarization makes use of paraphrasing, generalization and incorporates real-world knowledge. Naturally, generating abstractive summaries is more complicated and challenging in comparison to extractive summaries. Typical abstractive approaches first encode the input text into an internal representation and then use natural language generation techniques to generate the summary based on this internal representation. Therefore, abstractive summarization is one task in the broader field of natural language generation. The most important difference to extractive summarization is that abstractive summarization may generate novel words and phrases not featured in the source text – similar to human-written summaries. Consequently, abstractive summarization is a better approximation of the way humans write summaries. In addition to the key aspects of document summarization mentioned in the beginning of this section, novelty is an important factor of abstractive summarization.

Depending on the number of input documents, text summarization is categorized into Single-Document Summarization (SDS) and Multi-Document Summarization (MDS). Multi-document summarization aims to integrate key information from multiple text sources into a concise, comprehensive, fluent and accurate summary.

While SDS is considered the standard document summarization task, MDS brings many additional challenges. Most MDS systems are based on supervised learning which requires large amounts of labeled training data. However, obtaining training data for MDS is time consuming and resource intensive. Furthermore, different documents may contain the same content, include additional information, and present complementary or contradictory information. In contrast to SDS, cross-document links are also very important for MDS when extracting salient information, detecting redundancy and generating overall coherent summaries. Finally, MDS requires an effective representation of multiple input documents. Some approaches utilize graphs and in particular knowledge graphs to represent the information.

1.3 Faithfulness

Automatic summarization systems aim to generate summaries that are succinct, coherent and relevant. One of the most essential requirements for a practical summarization system is that the information in a generated summary must match the facts expressed in the source text. We refer to this aspect as faithfulness in this work. (Cao et al., 2018)

Faithful summarization is very important as fake summaries may greatly misguide the comprehension of the original text (Cao et al., 2018). Especially in the domain of radiology report summarization shortly introduced in Section 1.1, factual correctness is critically important to prevent medical errors. Also, faithful summaries are vital for real-world applications in the news domain to prevent the creation and spread of fake news. Figure 1.1 shows a misleading, unfaithful, automatically generated summary that demonstrates the importance of faithful summarization. Please refer to Appendix A.1 for more manually selected, unfaithful examples found in the XSUM hallucination dataset.

Progress in natural language generation led to models that can generate fluent and topical summaries. However, model generated summaries frequently contain factual inconsistencies with respect to their inputs, limiting their practical applications. Recently, Durmus et al. (2020) observed that near-extractive summaries have significantly higher faithfulness scores compared

Summary	Source
New rules have come into place that you can eat your dog.	The restaurant began serving puppy platters after a new law was introduced allowing dogs to eat at restaurants – as long as they were outdoors! It looks like a right dog's dinner – check out this clip.

Table 1.1: An automatically generated, unfaithful summary found in the XSUM hallucination dataset by Maynez et al. (2020).

to summaries generated by more abstractive models. Furthermore, the authors show that factual errors occur more frequently as models generate more abstractive sentences, i.e. less overlap with the source document. Durmus et al. (2020) also demonstrates that some models have factual errors in more than half of the sentences they generate when trained on very abstractive datasets.

Abstractive summarization involves rephrasing content into compact statements, ranging from minor editing of a sentence to condensing multiple sentences into a single one using other vocabulary. Thus, abstractive summarization systems are prone to generate content that is inconsistent with the source document. While extractive and near-extractive summarization systems are largely faithful – as they just copy sentences from the source document – abstractive models struggle to produce faithful summaries without copying (Durmus et al., 2020). Neural abstractive models are effective at identifying salient content, producing fluent summaries and have high overlap with human references (See et al., 2017; Y Zhang et al., 2020). However, most existing models are not optimized for factual correctness.

Recent studies have shown that around 30% of automatically generated summaries from neural summarization systems contain unfaithful information (Cao et al., 2018; Falke et al., 2019; Kryscinski et al., 2019), especially when the sentence combines content from multiple source sentences (Lebanoff et al., 2019). In accordance with that, Y Zhang et al. (2018) found that about 30% of the outputs from a radiology summarization model contain factual errors or inconsistencies. Such high levels of factual inconsistency render automatically generated summaries virtually useless in practice.

Y Zhang et al. (2020) claim one main reason for this problem is that most existing abstractive summarization models are optimized to generate summaries that highly overlap with human references. This, however, does not guarantee faithful summaries. Durmus et al. (2020) supports this, claiming that existing automatic metrics do not capture such mistakes effectively. According to A Wang et al. (2020), standard metrics like ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) are insensitive to semantic errors. These n-gram based approaches weight all portions of the text equally, even when only a small fraction of the n-grams carry most of the semantic content. As a result, factual inconsistency caused by small changes may be drowned out by otherwise high n-gram overlap. For example, the sentences "I am writing my thesis in Hamburg." and "I am not writing my thesis in Hamburg." share nearly all unigrams and bigrams despite having the opposite meaning. To sum up, a major reason for the existence of the factual inconsistency problem is the lack of automatic evaluation metrics that can detect such errors.

The inadequacy of automatic metrics leaves human evaluation as the main method for evaluating factual consistencies (A Wang et al., 2020). However, Durmus et al. (2020) recently observed that

human agreement when evaluating faithfulness is low for abstractive summaries. This suggests that catching faithfulness errors is difficult, even for humans. This also demonstrates another reason for the existence of the faithfulness problem and the lack of good automatic metrics: writing a faithful summary and detecting unfaithful information is indeed very difficult.

1.4 Explainability

Traditional natural language processing (NLP) systems have been mostly based on white box techniques. These systems are inherently explainable. Examples of such approaches include rules, decision trees, hidden markov models, logistic regressions and others. In recent years, however, many state-of-the-art NLP models have been constructed as black boxes. Black boxes are models or systems that hide their internal logic to the users. These systems allow powerful predictions at the expense of less interpretable models. (Danilevsky et al., 2020)

A key problem of black box models is that they often lack transparency. By relying on unexplainable black box models, we risk to create and use systems that we do not really understand. An inherent risk of such systems is the possibility of making wrong decisions or, e.g. in the case of summary generation, generating unfaithful or fake texts. This risk is directly related to our trust in these systems. It is hard for users but especially companies to trust products without understanding or validating the underlying techniques and systems. (Adadi and Berrada, 2018; Danilevsky et al., 2020)

Delegating tasks or decisions to black boxes without being able to interpret the system can be critical and can lead to severe issues. One example is the study of Caliskan et al. (2017) which demonstrates discrimination. The study revealed that text and web corpora contain human biases. The authors found that names associated with black people are significantly more associated with unpleasant terms than with pleasant terms, compared to names associated with white people. As a result, black box models trained on such corpora can inherit the prejudices present in the data. (Guidotti et al., 2018)

The experiment conducted by Liang et al. (2018) is another case that demonstrates how and why black box models are dangerous. Current deep neural networks reach very good performance on text classification tasks. However, their experiments show that small perturbations of the input, which are hardly noticeable by humans, were able to fool state-of-the-art models to misclassify a text as any desirable class. These findings raise reasonable doubts about trusting such black boxes.

To address issues like that, Explainable Artificial Intelligence (XAI) proposes to make a shift towards more transparent AI. XAI is a research field that aims to make AI systems' results more understandable to humans. XAI aims to "produce more explainable models while maintaining a high level of learning performance (prediction accuracy); and enable human users to understand, appropriately trust and effectively manage the emerging generation of artificially intelligent partners". (Adadi and Berrada, 2018)

Explainable systems, as opposed to black box models, are powerful tools for justifying prediction and come with various other benefits. Explainable systems can help to verify predictions, improve models and gain new insights about the system which, in turn, can lead towards even more trustworthy systems (Danilevsky et al., 2020). Furthermore, explainability can prevent

critical errors. Being able to understand a system's behaviour in a better way leads to greater visibility over unknown vulnerabilities and flaws.

Another reason to build explainable models is the need to continuously improve them. Explainability helps to rapidly identify and correct errors. It is way easier to improve a model that can be explained and understood than to improve a model where you can only guess what went wrong. Additionally, understanding how a model works may also allow users to provide useful and meaningful feedback which, in turn, helps developers to further improve the model.

Finally, asking for explanations can be helpful to learn new facts, gather information and gain knowledge. Only explainable systems can be used for this. For example, given that AlphaGo Zero can excel at the game of Go much better than human players, it would be desirable that the machine can explain its learned strategy (knowledge) to us. (Adadi and Berrada, 2018)

XAI affects a large range of domains and can bring significant benefits to them.

Machine-learning models are used in many scientific research areas, for example in medicine, biology and socio-economic sciences. Scientific domains require an explanation not only for trust and acceptance of results but also for the sake of the openness of scientific discovery and the progress of research. Here, explainable models can lead to new scientific discoveries. (Guidotti et al., 2018)

Also in the transportation domain and the upcoming era of automated vehicles, explainable models are crucial. Imagine a self-driving care suddenly acting abnormal because of some misclassification problem. Consequences can be dangerous and only an explainable system can help to investigate the circumstances and the situation to eventually prevent it from happening. Here, explainable models can enable automated vehicles which, in turn, can decrease traffic deaths and provide enhanced mobility. (Adadi and Berrada, 2018)

Other exemplary domains include criminal justice where models could asses risks for recidivism while it has to be ensured that such models behave honest and non-discriminatory, medical diagnosis where models can be utilized to detect diseases but users of such models have to be very confident in them since they are responsible for human life and financial services where AI can be used to predict credit scores but need to explain to borrowers why they were denied credit. (Adadi and Berrada, 2018)

Unfortunately, explainable models are still rare since it is very challenging to develop such systems. Typical machine learning (ML) algorithms base their predictions on their own representations of the world which they learned from many observations (training data). Due to their structure and the way they work, ML algorithms are difficult to interpret. They consider high-degree interactions between input features making it difficult to break it down into a human understandable form. While a single linear transformation can be understood by looking at the weights from the input features to the output, multiple stacked layers with non-linearities in-between are impossible for humans to disentangle. (Adadi and Berrada, 2018)

Considering summarization, most existing systems cannot explain the generated summary. However, some initial research and work towards explainable models has been done, which is described in Chapter 2.

1.5 Problem statement

Summarization systems faced and solved many challenges in the previous years. While early approaches struggled with out-of-vocabulary words and repetition, more recent work was found to be too extractive. Repetition was commonly solved with post-processing or context mechanisms (See et al., 2017). Up to a certain point, most abstractive summarization architectures were based on LSTMs (Hochreiter and Schmidhuber, 1997). As many other natural language processing tasks, summarization made a big leap with the introduction of the transformer architecture. Current approaches based on large pre-trained transformers are able to generate very abstractive summaries achieving almost human-like performance. While the transformer architecture brought many benefits, it also introduced a new maximum input token limitation (commonly 512 tokens) due to the attention mechanism. This is especially problematic for tasks that require a large context like long-document summarization of e.g. research papers or multi-document summarization in general. Fortunately, the token limitation problem was recently addressed by a new attention mechanism called BigBird. While all the previously mentioned limitations are pretty much solved, there are still many challenges limiting the application of summarization systems in practice including explainability and faithfulness.

The two previous sections highlighted the importance of faithfulness and explainability. Faithfulness is key requirement for summarization as many real-world applications demand factual consistency. Explainability is also crucial for many real-world applications of summarization since most companies or users will not risk to deploy and use systems that they do not fully understand and trust. Building transparent, explainable, faithful summarization systems is an important challenge. Tackling this challenge could finally enable this important technique for many domains helping many users to overcome the information overload problem.

The previous sections also summarized some challenges and the current state of faithfulness and explainability. The main reasons for unfaithful summarization systems are the inadequacy of automatic metrics as well as the way abstractive summarization models are trained, not including any notion of faithfulness. Challenges of explainability are connected to the inner workings of machine learning algorithms and are, as a consequence, inherently difficult to solve. Both, faithfulness and explainability are still open research topics. Some initial research regarding faithfulness includes the development of new evaluation metrics, whereas initial work on explainability follows very different approaches which trace back the generated output to input sentences.

These open challenges and the lack of solutions motivate this work. We argue that the currently available faithfulness metrics do not use all methodology that the natural language processing field has to offer. Consequently, we explore new methods to assess the faithfulness of generated texts and compare them to existing approaches. In addition to that, we analyze the potential of different faithfulness metrics pointing out which metric benefits the most from better components. Therefore, we develop a special tool that visualizes important aspects of faithfulness metrics explaining the predictions and supporting us with qualitative analyzes. Equipped with this tool, we also study how the faithfulness metrics actually perform in practice. Finally, we research possibilities how faithfulness metrics can be utilized – apart from evaluating summarization systems – in order to develop faithful summarization systems.

We study the following research questions (RQ) in this work:

1. We re-implement existing and propose several new faithfulness metrics. Next, we evaluate the metrics correlating them with human judgements. In the first part of this work, we explore the question "Which faithfulness metric is the best?" (RQ1).
2. We develop FaithAnno, an annotation tool specialized for visualizing and analyzing the outputs of faithfulness metrics. Using FaithAnno, we perform qualitative analyzes of faithfulness metrics considering their potential, error sources as well as pro's and con's. In the second part of this work, we investigate the question "How do faithfulness metrics perform in practice?" (RQ2).
3. We explore pre- and post-processing steps of a typical summarization model training pipeline that utilize faithfulness metrics to increase the faithfulness of the resulting summarization system. Furthermore, we study how automatic faithfulness metrics could support human faithfulness evaluation. In the final part of this work, we consider the question "How can faithfulness metrics be used to develop faithful summarization systems?" (RQ3).

The rest of this work is organized as follows. In Chapter 2, we review related work. Chapter 3 describes existing and our proposed faithfulness metrics. Next, all metrics are evaluated and compared in Chapter 4. In Chapter 5, we explain FaithAnno and conduct qualitative analyzes to to comprehend the metrics' performance. Finally, we explore possible ways to utilize faithfulness metrics to improve the faithfulness of summarization models in Chapter 6. Chapter 7 concludes this work and gives an outlook on what could be done in the future.

2 Related work

In this work, we re-implement popular existing methods to detect factual inconsistencies in generated text. Also, we develop several new faithfulness metrics in search for the best one. These methods are realized using various NLP technologies. As a result, we are addressing a wide variety of topics in this work e.g. open information extraction to extract facts and fact descriptions, named entity recognition to find entities, question answering and question generation to ask for and find certain information or semantic role labeling to extract more meaning from texts.

In this chapter, we review related work concerning information extraction, faithful summarization systems, faithfulness evaluation metrics and text generation evaluation metrics in general. Furthermore, we look at fact correction models, explainable summarization systems as well as current state-of-the-art pre-trained models for abstractive summarization.

2.1 Information extraction

The information extraction (IE) process in general is about turning the unstructured information that is present in texts into structured data. Information extraction is a large research field that comprises of multiple tasks: named entity recognition (NER) is about finding each mention of a named entity and labeling its type. The task of co-reference resolution or entity linking is to connect entities by inferring that different mentions refer to the same entity. The relation extraction task is to find and classify semantic relations among the text entities. Basic relations are for example part-whole, is-a and child-of. Event extraction is about finding events where the recognized entities participate in. Further information extraction tasks are extracting times or temporal expressions and template filling. (Jurafsky and Martin, 2009)

While early work addressed one aspect at a time, recent work developed methods that solve multiple tasks at once. A well-known scientific information extraction framework is SciIE developed by Luan et al. (2018). Their framework is able to identify and classify entities, relations and co-reference clusters in scientific articles. This framework has been improved into a more general information extraction tool called DyGIE by Luan et al. (2019). In contrast to SciIE, the improved version utilizes dynamic span graphs that allow to incorporate rich contextual information into the span representations. In another iteration, Wadden et al. (2019) extended DyGIE to also support event extraction. It is further improved by utilizing BERT embeddings that capture within- and adjacent-sentence context in combination with span graphs that capture global context information. The resulting IE framework DyGIE++ can be considered state of the art and is used in several papers cited in this chapter.

In contrast to standard information extraction, Open information extraction (Open IE) is not about classifying entities, relations etc., instead, it is about finding tuples of natural language expressions that represent the basic statements of a sentence. In simple scenarios, these tuples represent subject, verb and object or to be more general, the predicate and it's arguments. An

exemplary system for Open IE, which is also included in the AllenNLP Natural Language Processing Platform (Gardner et al., 2018), was developed by Stanovsky et al. (2018). They frame the Open IE task as a sequence tagging problem and train a bi-directional LSTM (Hochreiter and Schmidhuber, 1997) transducer in a supervised way on their newly introduced corpus. While outperforming previous systems, their method struggles with nominalized predicates and long sentences.

We will see in Section 2.3 that many approaches to faithful summarization systems extract facts with information extraction components from the source document and feed them as additional input into the summarization model with the aim to preserve the original facts and provide the model with condensed information.

2.2 Abstractive summarization models

Architectures

Recurrent Neural Networks (RNNs), long short-term memory (LSTMs, Hochreiter and Schmidhuber, 1997) and gated recurrent units (Chung et al., 2014) dominated many sequence-to-sequence tasks (e.g. machine translation or summarization) of the natural language processing (NLP) community. However, the introduction of the transformer architecture by Vaswani et al. (2017) and later the introduction of BERT by Devlin et al. (2019) revolutionized this field achieving state-of-the-art performance on most NLP tasks. As a result, many newly proposed NLP models are based on the transformer architecture. We will see in the next section that all current, best-performing summarization models are based on transformers.

Typical sequence-to-sequence models have an encoder-decoder structure. The encoder maps an input sequence $(x_1, x_2, ..., x_n)$ consisting of symbols to a sequence of continuous values $z = (z_1, z_2, ..., z_n)$. Given the encoded representation z, the decoder generates an output sequence $(y_1, y_2, ..., y_m)$ of symbols, one symbol at a time. During the generation, the decoder consumes the previously generated symbols as additional input at each step.

The transformer (Vaswani et al., 2017) also uses an encoder-decoder structure. Both encoder and decoder are composed of a stack of N identical layers. Each encoder layer consists of a multi-head self-attention mechanism and a fully connected feed-forward network, while each decoder layer additionally includes a multi-head attention over the output of the encoder. The self-attention is arguably the most important component of the transformer and the main reason of it's success. Compared to recurrent and convolutional layers, the self-attention layer has a lower computational complexity, it's computation can be highly parallelized and it can easily learn long-range dependencies, which is a very important challenge in many sequence-to-sequence tasks. In addition to that, the inspection of model attentions allows to interpret transformer-based models.

BERT (Bidirectional Encoder Representations from Transformers) is a language representation model introduced by (Devlin et al., 2019). Obtaining general language representations that are applicable to many different tasks is a research area in itself and has a long history. Apart from BERT, recent approaches include for example ELMo (Peters et al., 2018) and GPT (Radford et al., 2018). Many researchers demonstrated that Transfer Learning, e.g. pre-training a general language model and fine-tuning it on a specific task, is very effective. For example, Devlin et al. (2019) show that BERT can improve many NLP tasks including natural language

understanding, question answering and textual entailment. Fine-tuning, in short, is the process of introducing a small amount of task-specific parameters (e.g. a classification head or span-selection head) to the model and then simply training all parameters – both pre-trained and task-specific parameters – on the downstream task.

BERT's architecture is a multi-layer bidirectional transformer encoder. There exist two versions of BERT: $BERT_{BASE}$ consists of $N = 12$ layers and $BERT_{LARGE}$ consists of $N = 24$ layers. BERT improves upon previous approaches by addressing a major limitation of standard language models. Previous language models were unidirectional limiting the choice of architectures and being harmful for sentence-level or token-level fine-tuning tasks where it is very important to incorporate context from both directions. In contrast, BERT is bidirectional and addresses this issue by using the masked language model (MLM) pre-training objective.

BERT is pre-trained on over 3,000 million words using two unsupervised tasks: MLM and Next Sentence Prediction (NSP). In the MLM task, 15% of the input tokens are randomly masked-out and then the model has to predict these masked tokens. This technique allows to train bidirectional representations. The NSP tasks is designed to learn the relationship between two sentences since many downstream tasks like question answering or natural language inference are based on understanding such relations. It is a binary task where 50% of the time the actual next sentence (labeled as True) and 50% of the time a random sentence (labeled as False) appears in the training data and the model has to predict the correct label given two sentences.

State-of-the-art models

Pre-trained masked language models based on transformers such as BERT (Devlin et al., 2019) revolutionized the field of Natural Language Processing (NLP) and brought great improvements to a wide variety of NLP tasks including document summarization. The success can be mainly attributed to the introduction of a self-attention mechanism in the transformer architecture as well as the pre-training on enormous amounts of data using self-supervised learning objectives. Since the introduction of the token masking training objective of BERT, many other training objectives were developed.

BART (Lewis et al., 2020) is trained on corrupted text generated with a noise function. The model learns to reconstruct the original text. The corruption method is called text infilling where text spans of various lengths are replaced with a single [MASK] element. As a result, BART is especially good for text generation tasks and outperforms previous summarization methods based on BERT like BERTSUMABS (Liu and Lapata, 2019).

PEGASUS (J Zhang et al., 2019) uses a training objective called gap-sentence-generation that closely resembles the summarization task: important sentences are removed from the input document and the model has to generate the removed sentences as one output sequence given the remaining sentences. The model is trained on the colossal and cleaned version of common crawl (C4) as well as HugeNews which consists of 1.5B news articles from the web. Unsurprisingly, this method achieves state-of-the-art results on many summarization datasets.

ProphetNet (Qi et al., 2020) achieves similar performance to PEGASUS, even though it is only trained on 160GB of data (compared to 3.8TB HugeNews + 750GB C4). This is possible because of two newly introduced techniques. First, the authors propose a new self-supervised training objective called future n-gram prediction: instead of predicting one token, ProphetNet learns

to predict n tokens at every timestep. Second, they introduce a novel n-stream self-attention mechanism.

BigBird (Zaheer et al., 2020) is an attention technique whose complexity is linear in the number of input tokens instead of quadratic. This enables transformers to process up to 4096 input tokens where previously only 512 input tokens were possible. As a result, this improves the performance of transformer-based models on many NLP tasks. However, this technique is especially beneficial to tasks that require large contexts like long-document or multi-document summarization. BigBird in combination with PEGASUS achieves state-of-the-art performance on several long-document summarization datasets. (Zaheer et al., 2020)

The previously mentioned pre-trained models for abstractive document summarization have great – some have even human-like – performance on multiple summarization datasets (J Zhang et al., 2019). There exist even more models like UniLM, MASS and T5 that differ by 1 to 2 ROUGE points and, therefore, perform similarly.

2.3 Faithful summarization systems

While many recent abstractive summarization models (e.g. Pointer-Generator Network by See et al. (2017), T5 by Raffel et al. (2020) or BART by Lewis et al. (2020)) are still trained and optimized without faithfulness in mind, there is some initial research towards faithful summarization models. These approaches typically extract facts with a specialized information extraction component from the source document and feed them as additional input into the summarization model.

Cao et al. (2018) try to address the faithfulness problem by extracting facts from the source text and generating the summary based on the source text as well as the extracted facts. They leverage Open IE and dependency parser tools for fact extraction and propose a model that is able to reduce the number of fake summaries by 80%. The model's encoder consists of two RNNs: one encoder reads the fact descriptions and the other encodes the source document. The authors also note that fact extraction benefits the informativeness of generated summaries as facts often condense the meaning of the source text.

Huang et al. (2020) use a graph-based summarization approach. Typically, such approaches use and encoder-decoder model and combine it with some kind of graph-informed attention mechanisms. The authors, however, use a slightly different architecture. Instead of having only one encoder with graph-based attention, the propose to use two encoders: one for the document, one for the graph. Furthermore, they optimize their model using reinforcement learning and a special question answering reward that aims at optimizing the faithfulness of the generated summaries. The questions and answers are generated with the help of an information extraction component. The document encoder is realized with RoBERTa (Liu et al., 2019), while the graph encoder is implemented with a Graph Attention Network (Veličković et al., 2018). Huang et al. (2020) experiment with knowledge graphs that were constructed using Open IE as well as co-reference resolution methods. They find that knowledge-graph enhanced models like theirs can improve the faithfulness of generated summaries compared with pre-trained language models like BART (Lewis et al., 2020).

Y Zhang et al. (2020) claim to be the first that attempted to directly optimize a neural summarization system with a factual correctness objective via reinforcement learning. Similar to Huang

et al. (2020), the authors utilize an external information extraction component to extract facts from the generated summary and calculate a factual accuracy score by comparing it against the reference summary. Their training strategy combines a factual correctness objective, textual overlap objective as well as a language modeling objective which are jointly optimized with reinforcement learning. However, their work focuses on the radiology domain and, therefore, uses a special IE system called CheXpert (Irvin et al., 2019). As a result, their approach is dependent on a carefully implemented information extraction system that is very specific to the application domain and not applicable for a more general case.

Gabriel et al. (2021) introduce the Cooperative Generator - Discriminator Network (Co-opNet) where discriminator and generator work together to compose coherent long-form summaries. The transformer-based generator, which is fine-tuned for abstractive summarization, proposes a set of candidate summaries. The discriminator, also transformer-based, scores the factual correctness and discourse quality of the candidate using novel objectives. Finally, the best summary is chosen according to the combination of the generator's and discriminator's score. The factuality objective is particularly interesting in the context of this work: the authors use a binary BERT-based token classification model that predicts whether a token is likely to belong to a fact-checking evidence span. Then, the extracted spans are compared to information of the source document to measure the degree to which the abstractive summarization model is hallucinating information.

2.4 Faithfulness metrics

The lack of automatic evaluation metrics for faithfulness has motivated some researches to develop new metrics that ideally mimic human judgements of factual consistency. Even though ROUGE and other common evaluation metrics for text generation are known to be insensitive to semantic errors, only little work on developing better automatic evaluation metrics has been done. Common approaches are based on question answering, textual entailment and contextual embeddings.

A Wang et al. (2020) and Durmus et al. (2020) introduce the question answering approach. This approach is based on the following assumption: if we ask questions about a summary and its source, we will receive similar answers when the summary is factually consistent with the source. Questions are generated based on the summary (rather than the source document) with a BART-based question generation model, whereas questions are answered by a BERT-based question answering model fine-tuned on SQuAD2 (Rajpurkar et al., 2018) using the source document. The final faithfulness score depends on the similarity of the answers.

Kryscinski et al. (2020) propose FactCC (Factual Consistency Checking model), a BERT-based model that, given a summary sentence and the source document, makes a binary decision whether the summary is consistent or inconsistent with the source document. In addition to that, their more advanced explainable FactCCX model is able to extract spans in the source document that support this consistency prediction and extracts inconsistent spans for each inconsistent summary sentence. Here, the extracted spans function as an explanation for the prediction. The authors create a synthetic dataset instead of collecting a human annotated dataset in order to train such a model with the ability to predict spans. This dataset is constructed by extracting claims from the source documents and then passing them through a set of textual transformations that eventually outputs new sentences with positive and negative labels. These textual transformations include

paraphrasing, entity and number swapping, sentence negation, pronoun swapping and noise injection. The authors state that their model is a useful assistance to humans for verifying the factual consistency between source document and generated summary.

Maynez et al. (2020) conduct a large scale human evaluation of neural abstractive summarization systems. While it is not large enough to train a model, the collected dataset is sufficient to analyze benefits and drawbacks of current abstractive summarization systems. One main finding is that pre-trained models are better summarizers in terms of generating faithful and factual summaries. The authors also compare automatic faithfulness metrics based on textual entailment and question answering with their collected human judgements. They claim that the BERT-based textual entailment model correlates well with human faithfulness judgements even though it was trained on the Multi-NLI dataset (Williams et al., 2018), which is not optimal for this task as it consists of sentence-sentence pairs, while their faithfulness assessment task consists of document-sentence pairs.

Similar to that, Falke et al. (2019) share the idea that all information in a summary should be entailed by the source document and propose to use textual entailment to detect factual errors, too. In addition to that, they propose to re-rank potential summaries based on the factual correctness during beam search and use the highest scoring summary as the final output. This way, the authors try to improve existing summarization systems but conclude that current textual entailment models are not yet good enough for this task as the re-ranking barely improved faithfulness. Their evaluation introduced the sentence re-ranking experiment, which was later on adopted by several researchers to evaluate faithfulness metrics.

Nan et al. (2021) propose a simple set of metrics addressing the entity hallucination problem. Factual inconsistencies can occur at different levels and the authors specifically focus on the problem of unfaithful entities where a model generated summary can contain named entities that never appeared in the source document. The authors perform named entity recognition and calculate the percentage of named entities in the summary that can be found in the source. A low precision means entity hallucination is severe. In addition, they propose precision-target and recall-target, which capture the entity-level accuracy of the generated summary with respect to the ground truth summary.

Goodrich et al. (2019) propose to evaluate the factual correctness of generated text with relation extraction methods. Facts are represented as subject-relation-object triples and faithfulness is defined as the precision between the facts extracted from the generated summary and target summary. Typically, relation extraction or fact extraction pipelines consists of multiple components like named entity recognition, co-reference resolution and relation classification. The authors develop an end-to-end relation extraction model to avoid the compounding of errors over all sub-components. However, their model is only able to extract a limited number of pre-defined facts for a given subject. This works well for their task of generating summaries for Wikipedia, where most documents are dedicated to exactly one subject, but their approach is not applicable to other, more general summarization tasks.

2.5 Text generation evaluation metrics

Summarization is just one sub-field of the larger research field of text generation. Other tasks include for example machine translation, knowledge-graph-to-text generation, AMR-to-text generation and many more. These other text generation fields of course also have to deal with

similar issues like factual inconsistencies and developed new interesting metrics, some of which even work to assess faithfulness.

BERTScore (T Zhang et al., 2020) is an automatic evaluation metric for text generation in general, but was originally developed for machine translation and caption generation. It is based on BERT's contextual embeddings with the goal to evaluate semantic equivalence of two texts. BERTScore computes the similarity by calculating the sum of cosine similarities between the token embeddings of two texts. Optionally, the cosine similarities can be weighted e.g. by their token's inverse document frequency score.

YiSi (Lo, 2019) is a unified framework for machine translation quality evaluation and estimation. This measure is designed to deal with varying availability of linguistic resources which resulted in three different versions: YiSi-0, YiSi-1 and YiSi-2. The different variants function as fallback options in case some resources are not available for a language. This leads to a metric that can be used for any language. The variants have different definition of lexical similarities and different requirements. For example, YiSi-0 utilizes the longest common character substring accuracy to evaluate the lexical similarity, while YiSi-1 uses an embedding model and YiSi-2 requires a cross-lingual embedding model and optionally a semantic role labeler for both languages. The basic strategy of the YiSi metric is to apply a semantic role labeler to both sentences, align the semantic frames and finally compare the arguments using a sophisticated weighted f-score based on lexical similarities. Therefore, YiSi assesses both the semantic roles as well as the semantic and lexical similarities to evaluate generated text.

Abstract Meaning Representations (AMR) try to capture the meaning of a sentence in a graph format. The task of AMR-to-text generation is to generate a surface realization of the graph. One issue that is present in this task is meaning preservation, which is similar to faithfulness: the generated text has to convey the same meaning as the AMR graph. Opitz and Frank (2021) propose a decomposable metric MF that measures meaning preservation (M) as well as grammatical form (F). To assess the meaning preservation, the authors propose to construct an AMR from the generated text using state-of-the-art parsing models and compare it to the input AMR. The two graphs are compared using a well-defined graph matching metric commonly used in this field called Smatch. The form of the text is evaluated using language model token probabilities where higher sequence probabilities are more desirable.

Bhandari et al. (2020) highlight the discrepancy between the fast, rapid progress of model development in summarization and the stagnant development of new evaluation metrics where ROUGE is still by far the most popular one. The authors perform a meta-evaluation of common evaluation techniques for summarization systems including BERTScore (T Zhang et al., 2020), ROUGE (Lin, 2004), MoverScore (Zhao et al., 2019), Sentence Mover Similarity (Clark et al., 2019) and Jensen-Shannon divergence (Lin et al., 2006). They evaluate multiple well-known summarization systems like the pointer generator network (See et al., 2017), T5 (Raffel et al., 2020) and BART (Lewis et al., 2020) on the datasets CNN/DailyMail (Hermann et al., 2015) and TAC-2008, 2009. The results show that there is no one-size-fits-all metric that can outperform all others on all datasets suggesting to use different metrics for different datasets. Furthermore, the results reveal that the choice of the metrics should not only depend on the task (e.g. summarization, translation), but also on the application scenario (e.g. system-level, summary-level evaluation). The authors call for action to introduce a shared-task similar to WMT (a metrics task in machine translation) where summarization systems and metrics can co-evolve.

2.6 Fact correction

Another field of research tries to solve the faithfulness problem with post-processing. In the same way as spell checkers or grammar correction systems post-process text to decrease the number of grammatical errors, fact correction systems try to correct wrong facts in the given text and, therefore, improve the factual correctness of any given text.

Zhu et al. (2021) develop a factual corrector model that improves the faithfulness of any given abstractive summary. The model corrects the summary with minimal changes to be more factually consistent with the original article. The authors phrase the task of fact correction as a sequence-to-sequence problem: the input is the original text and the output is the fact-corrected text. Their proposed model is based on the UniLM architecture and fine-tuned on a synthetic dataset of ground-truth summaries where entities are randomly replaced with other (wrong) entities of the same type from the article. As a result, their model learns to correct unfaithful entities.

Dong et al. (2020) introduce SpanFact which is a factual correction framework that focuses on correcting unfaithful facts in generated summaries and is inspired by fact-checking question answering models. The authors phrase the task of fact correction as a span selection task: the query is a masked summary where one or all entities are deleted and the passage is the corresponding source document. They propose two span-selection approaches: single and multi-span selection. While the single-span selection method iteratively masks out one entity at a time and then predicts the correct entity, the multi-span selection method masks out all entities at once and than predicts the answers from left to right. Both models are implemented using BERT (Devlin et al., 2019) with a span classification head on top for extractive question-answering. They are trained on the SQuAD question answering dataset (Rajpurkar et al., 2016; Rajpurkar et al., 2018) and then fine-tuned on a summarization dataset. They conclude that the single span selection model works well for summaries that contain few errors, while the multi-span selection model works better for summaries that contain many factual errors.

Two common advantages of such fact correction systems are that they only make minimal changes to the original summary and, therefore, have almost no noticeable change in ROUGE scores as well as that they are model-agnostic as post-processing methods work with any summarization model.

2.7 Explainable summarization systems

Explainability or explainable AI (in short XAI) is a quickly growing research area that aims to enable users to understand, trust and manage artificially intelligent systems. Many scientists realized the importance and benefits of explainable and transparent models and began to develop such models. While some areas of machine learning like computer vision have already many explainable solutions, transparent models are still rare and only initial research has been done on transparent and explainable summarization. Here, the development of such models has just begun.

ExtremeReader (X Wang et al., 2020) and ExplainIt (Carmeli et al., 2021) operate on the product review domain and utilize opinion causality graphs (OCGs) to generate summaries. The OCGs function as structured summaries where nodes represents opinions and edges represent causality

relationships. This way, generated summaries based on the OCGs can be explained by following the causality relationships. Furthermore, ExtremeReader is equipped with provenance features: the system can show for each opinion the original review sentence from which the opinion was extracted.

ESCA (Haonan et al., 2020), the Explainable Selection module to Control the generation of Abstractive summary, is based on a select-and-generate framework. First, salient sentences are selected by an extractor. Then, an abstractor generates the abstract summary. The authors introduce a mechanism to control novelty and relevance of sentences during sentence selection and generation.

Fan et al. (2019) tackle explainability by visualizing the model attention. Their proposed summarization system is based on knowledge graphs. Their visualization highlights nodes and edges of the knowledge graph that the model attends to during the summary generation process. This results in an interpretable graph that corresponds well to the generated summary.

3 Faithfulness metrics

We re-implement several popular model-based faithfulness metrics and adopt them when necessary. Furthermore, we propose multiple new methods that extract and compare different kinds of information from text to assess factual consistency. In this chapter, we describe for each metric the general idea, the basic architecture or calculations as well as important implementation details.

3.1 BERTScore

BERTScore was originally introduced by T Zhang et al. (2020) as a metric for text generation, but was initially only tested on machine translation and image captioning datasets. Later, some researchers experimented with BERTScore as an automatic metric for faithfulness (A Wang et al., 2020; Durmus et al., 2020; Maynez et al., 2020) and just recently Koto et al. (2020) reported very promising results. The authors show that BERTScore has good correlation with human faithfulness judgements which is why we include this metric in our experiments.

Idea: The main idea of BERTScore is very simple while still fixing two common problems of n-gram-based metrics like ROUGE, METEOR or BLEU. First, n-gram-based methods often fail to match paraphrases as they use string matching (e.g. BLEU) or matching heuristics (e.g. METEOR). Second, n-gram-based approaches cannot capture distant dependencies and fail to punish ordering changes like swapping cause and effect. BERTScore utilizes contextual embeddings to compute a similarity score between every token in the candidate sentence and reference sentence. Computing the similarity with contextual embeddings is effective for matching paraphrases. Furthermore, contextual embeddings are trained to capture distant dependencies and ordering.

Computation: Let x be a reference sentence $x = x_1, ..., x_n$ and a y be candidate sentence $y = y_1, ..., y_m$ consisting of tokens x_i and y_j, respectively. The tokens are represented as contextual embeddings and cosine similarity is used to compute the matching.

Contextual embeddings, such as BERT (Devlin et al., 2019), represent words depending on the surrounding words (the context of the target word). As a result, these methods are able to produce different vector representations for the same word. Typically, the representation for each token is computed with a transformer encoder (Vaswani et al., 2017) that repeatedly applies self-attention and nonlinear-transformations.

These vector representations or word embeddings allow matching beyond exact match. The cosine similarity of two tokens x_i and y_j is calculated as follows:

$$\frac{x_i^T \cdot y_j}{|x_i| \cdot |y_j|} \tag{3.1}$$

Please note, that even though this computation considers the tokens in isolation, the contextual embedding of the tokens contain information about the rest of the sentence.

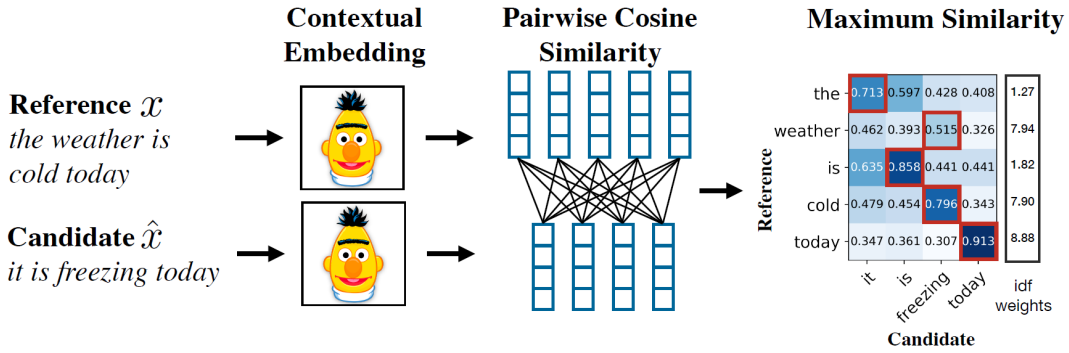

Figure 3.1: Illustration of the computation of BERTScore. Given reference sentence x and candidate sentence $\hat{x}$, the contextualized token embeddings are calculated with BERT. Then, the pairwise cosine similarities are computed. The maximum matchings are highlighted in red.

Figure 3.1 illustrates the computation of BERTScore. Every token in x is matched to a token in y to compute recall and each token in y is matched to a token in x to compute precision using maximum matching: each token is aligned to the most similar token in the other sentence. The three variants of BERTScore (precision, recall, F1) are shown below:

$$R_{BERT} = \frac{1}{|x|}\sum_{x_i \in x} \max_{y_j \in y} x_i^T y_j \quad P_{BERT} = \frac{1}{|y|}\sum_{y_j \in y} \max_{x_i \in x} x_i^T y_j \quad F_{BERT} = 2\frac{P_{BERT} \times R_{BERT}}{P_{BERT} + R_{BERT}} \tag{3.2}$$

Implementation details: We optimize BERTScore slightly before utilizing it for faithfulness evaluation. The original authors of BERTScore recommend to use the 24-layer RoBERTa-large model to compute the BERTScore and use F_{BERT} when evaluating text generation in English. However, they also note that there is no single configuration of BERTScore that outperforms all others. Instead, one should consider the domain and languages when selecting the configuration to use. Therefore, we need to select the best-performing model and the best layer for our task of faithfulness assessment. Selecting the correct layer is very important, as each layer encodes different information. Luckily, Koto et al. (2020) perform an exhaustive layer search over 19 models and find that layer 8 of the RoBERTa-large model fine-tuned on the Multi-NLI dataset (Williams et al., 2018) achieves the best correlation with human faithfulness judgements. Therefore, we also use this configuration in our experiments.

Furthermore, we test an additional computation step: importance weighting. This step was originally proposed by the BERTScore authors. However, they quickly discard that idea as they find little to no improvements. We want to evaluate the impact of importance weighting on the task of faithfulness evaluation. Thus, we compute inverse document frequency (idf) values for every token in our dataset and weight the tokens accordingly. Given M documents $d_1, ..., d_M$, the idf score of a token w is

$$idf(w) = \log M - \log \sum_{i=1}^{M} \mathbb{1}[w \in d_i] \tag{3.3}$$

where $\mathbb{1}[]$ is an indicator function. For example, combining idf weighting with the precision BERTScore results in the following equation:

$$P_{BERT} = \frac{\sum_{y_j \in y} idf(y_j) \max_{x_i \in x} x_i^T y_j}{\sum_{y_j \in y} idf(y_j)} \tag{3.4}$$

3.2 Textual Entailment

Textual Entailment (TE) (Dagan et al., 2005) is a controversial method to evaluate factual consistency. To the best of our knowledge, it was first employed by Falke et al. (2019) to evaluate faithfulness. They find that TE can help this issue, but their experiments demonstrate that out-of-the-box TE models (also known as Natural Language Inference [NLI]) do not perform very well on the task. Durmus et al. (2020) also experiment with TE models and confirm this finding. Their TE metric does not have a significant correlation with human faithfulness judgements, too. In contrast to that, Maynez et al. (2020) find that TE scores correlate well with human faithfulness judgements and they suggest that TE can be used as an automatic metric for faithfulness. We think the general idea, but also this controversy, is very interesting. Therefore, we decide to include TE in our experiments.

Idea: The basic intuition of the TE approach to measure faithfulness is that all information in a summary should be entailed by the source document. In the TE setting, a summary should ideally be entailed by the source document or perhaps be neutral to the source document, but the summary should never contradict it. The general approach is to train an TE classifier, use it to predict the entailment scores and interpret these entailment probabilities as faithfulness.

It is important to note that TE measures are referenceless. As a result, this metric can be gamed easily: for example, the first sentence of a source document is always entailed by the whole source document. Therefore, when developing a new summarization model, evaluation measures based on TE always need to be coupled with reference-based measures like ROUGE.

Architecture: There are many different off-the-shelf TE models available. Explaining them and highlighting their differences, however, is beyond the scope of this work. Instead, we explain shortly how a simple TE model could be obtained.

An TE classifier can be built by starting with a pre-trained BERT-based (Devlin et al., 2019) model with a classification head. Then, the model can be fine-tuned on e.g. the Multi-NLI dataset. This dataset consists of sentence pairs (premise-hypothesis) and a TE label (entailment, contradiction or neutral). The resulting model takes two sentences, e.g. one source document sentence and one summary sentence and predicts the scores for the entailment, contradiction and neutral classes.

An TE model like that can be utilized to calculate faithfulness scores in the following way. Let E be an TE model that predicts the probability $E(a,b)$ that sentence b is entailed by sentence a. The faithfulness score f of a summary S consisting of sentences $s_1,...,s_n$ with respect to the original document D with sentences $d \in D$ can be defined as follows:

$$f_{s2s}(S) = \frac{1}{n}\sum_{i=1}^{n} max_{d \in D} E(d, s_i) \tag{3.5}$$

This scoring was first introduced by Falke et al. (2019). They argue that it is sufficient for a summary sentence to be entailed by one source sentences, hence the max operator. But it is important to average over all summary sentences as all sentences should be entailed by the source document.

Implementation details: To implement TE as faithfulness measure, we utilize the huggingface transformers library (Wolf et al., 2020) and the publicly available fine-tuned checkpoint of a

RoBERTa-large model (Liu et al., 2019) trained on the previously mentioned Multi-NLI dataset. We also experiment with other TE models in Chapter 4 and 6.

In addition to the sentence-to-sentence (s2s) scoring method introduced by Falke et al. (2019), we implement two other scoring approaches. Again, let E be an TE model that predicts the probability $E(a,b)$ that sentence b is entailed by sentence a.

For the first document-to-sentence (d2s) scoring method, we feed the TE model a whole document (as opposed to a sentence) and a sentence as input. Given a summary S consisting of sentences $s_1,...,s_n$ and a document D, we calculate the faithfulness score as follows:

$$f_{d2s}(S) = \frac{1}{n}\sum_{i=1}^{n} E(D, s_i) \tag{3.6}$$

The second top-to-sentence (top2s) scoring method is a bit more advanced and involves a few extra steps. Given a summary S consisting of sentences $s_1,...,s_n$ and the source document D consisting of sentences $d_1,...,d_m$, we first utilize a sentence embedding model to obtain a vector representation of each sentence. Next, we calculate the cosine similarities between the sentence embeddings to find the k most similar source sentences $d_{top,i} \in D$ for each summary sentence s_i. Finally, we concatenate the k $d_{top,i}$ source sentences for each summary sentence s_i into a small paragraph p_i and compute the faithfulness score as follows:

$$f_{top2s}(S) = \frac{1}{n}\sum_{i=1}^{n} E(p_i, s_i) \tag{3.7}$$

3.3 Question Generation & Question Answering

The Question Generation & Question Answering framework was concurrently introduced by Durmus et al. (2020) and A Wang et al. (2020) and has already been used by a few researchers (Maynez et al., 2020; Koto et al., 2020; Dong et al., 2020). As this method is well-known and both original authors showed that it correlates well with human faithfulness judgements, we decide to include this faithfulness metric in our experiments.

Idea: The basic intuition of this framework is very simple: if we ask questions about a summary and its source, we expect to receive similar answers if the summary is factually consistent with the source. Naturally, more matched answers imply a more faithful summary as the information addressed by these questions are consistent between the summary and the source document.

To automatically detect factual inconsistencies, the general question generation and question answering framework follows three steps. First, a question generation (QG) model generates a set of question about a given generated text (e.g. the summary). Second, a question answering (QA) model is utilized to answer these questions using both the source document and the generated text. Finally, a faithfulness score is computed based on the similarity of the corresponding answers. As a result, this framework consists of three components: a question generation model, a question answering model and a answer similarity measure.

One important benefit of this approach is that it does not require a reference (e.g. a reference summary) to compare. Instead, this approach asks question based on the generated text and compares the answers with the source document. However, similar to the reference-less textual

entailment approach, it is necessary to couple this metric with a reference-based measure like ROUGE when developing a new summarization model. Furthermore, the use of questions focuses this metric on the semantically critical parts of the generated summary rather than weighting all parts of the generated text equally as for example typical n-gram based approaches. Most of the time, only a small fraction of the n-grams carry most of the semantic content. This framework aims to identify and compare exactly these important words.

Architecture: There are multiple ways to design and train such question generation and question answering models. There exist many off-the-shelf question answering models (also called reading comprehension models) e.g. implemented in the AllenNLP toolkit (Gardner et al., 2018). A good and popular strategy (e.g. employed by A Wang et al., 2020; Durmus et al., 2020; Koto et al., 2020) to train such models is to fine-tune a pre-trained BART model on the NewsQA (Trischler et al., 2017) dataset to obtain a question generation component and to fine-tune a pre-trained BERT model on the SQuAD1 (Rajpurkar et al., 2016) or SQuAD2 (Rajpurkar et al., 2018) dataset to obtain a question answering component.

The NewsQA dataset consists of news articles from the Cable News Network (CNN) as well as human-generated, crowd-sourced question-answer pairs. To train a question generation model on this dataset, the model receives the source article and an answer as input and is trained to predict the corresponding question. During testing, means to extract answers candidates are necessary. Simple approaches that come to mind for this task are keyword extraction, named entity recognition or extracting noun phrases.

The SQuAD1 and SQuAD2 datasets consist of a set of Wikipedia articles as well as questions and answers collected from crowdworkers. The answer to every question is a span from the corresponding article, which allows the development of extractive question answering models. To train a question answering model on this dataset, the model receives the article and question as input and is trained to predict the answer span. The SQuAD2 dataset contains, in contrast to SQuAD1, unanswerable questions. Considering faithfulness, if questions are unanswerable by the source document, the summary is most likely unfaithful.

Given such models, we can now define the faithfulness score. Let X be the source document and Y be the summary of X. Furthermore, let $p(Q|Y)$ be a question generation model that describes a distribution over all possible questions Q given summary Y and let $p(A|Q,Z)$ be a question answering model that describe distributions over all possible answers A to a question Q given a document Z. Then, the factual consistency score of the summary Y can be defined as

$$F_{Q\sim p(Q|Y)}\left[SIM\Big(p\left(A|Q,X\right),p\left(A|Q,Y\right)\Big)\right]$$

where SIM is a function that measures the similarity of the two answer distributions. This scoring function is maximized when summary Y contains a subset of the information in source document X so that a QA model finds the same answer for any question from $p(Q|Y)$. Please note that this metric can be easily gamed, e.g. when the summary is the same text as the source document ($X = Y$).

The similarity function SIM compares the answers distributions by calculating the similarity between all N extracted summary- and source-answers using the F1 surface (token-level) similarity, which is standard for extractive question answering:

$$SIM = \frac{1}{N}\sum_{\substack{a_s\sim p(A|Q,Y)\\ a_d\sim p(A|Q,X)}} F1(a_s,a_d) \tag{3.8}$$

A Wang et al. (2020) also experiment with exact match as an alternative answer similarity measure. However, they find that using F1 leads to higher correlation with human faithfulness judgements.

Implementation details: To implement the question generation & question answering framework as another faithfulness metric, we decide to use off-the-shelf models[1]. This work is based on the huggingface transformers library (Wolf et al., 2020). It includes a multi-task model (based on the T5 (Raffel et al., 2020) pre-trained language model) optimized for both question generation and question answering. Additionally, this work contains single-task models optimized for either question generation or question answering.

We argue that, while the F1 surface similarity is standard for extractive question answering, it is not sufficient for faithfulness evaluation. The key challenge of assessing faithfulness is to verify highly abstractive sentences, or in this case answers, against the source document. Targeting more abstractive generated texts, we need to go away from surface representations to other representations, so that e.g. paraphrased answers can be successfully compared as well. Therefore, we experiment with exact match, F1 and additionally test two other model-based similarity metrics described in detail in Section 3.9.

3.4 FactCCX

FactCCX (Factual Consistency Checking with eXplanations) is a model introduced by Kryscinski et al. (2020) that serves as a model-based metric to evaluate faithfulness. It is a known metric used by researchers to evaluate their summarization systems (Zhu et al., 2021; Dong et al., 2020) or to compete against (A Wang et al., 2020). Since some researchers demonstrate that FactCCX has good correlation with human faithfulness judgements, we decide to include this metric in our experiments.

Idea: FactCCX shares the intuition of the Textual Entailment-based faithfulness metrics: a factually consistent summary contains only statements that are entailed by the source document. However, the authors claim that checking faithfulness on a sentence-by-sentence basis, where each sentence of the summary is verified against each sentence from the source document, is not sufficient. Therefore, FactCCX can be seen as an approach to overcome the limitations of typical Textual Entailment models for faithfulness evaluation as described in Section 3.2. Instead of following the sentence-to-sentence approach, FactCCX follows a document-to-sentence approach where each sentence of the summary is verified against the whole source document.

Architecture: FactCCX is a weakly-supervised, BERT-based, binary classification model that evaluates the faithfulness of a system-generated summary by predicting whether it is consistent or inconsistent with respect to the source document. In addition to that, FactCCX is able to identify conflicts between source documents and generated summaries. This is achieved by several span selection heads that allow the model to highlight either spans in the source document that contain support for the summary sentence or spans in the summary where a possible mistake was made. As a result, FactCCX is able to both check faithfulness and explain it's prediction.

The model is jointly trained for three tasks: (1) predict whether a summary sentence is consistent with the source document or not, (2) extract a span in the source document that supports this prediction and finally, (3) extract the inconsistent span of a summary sentence, if the summary

1. https://github.com/fajri91/question_generation

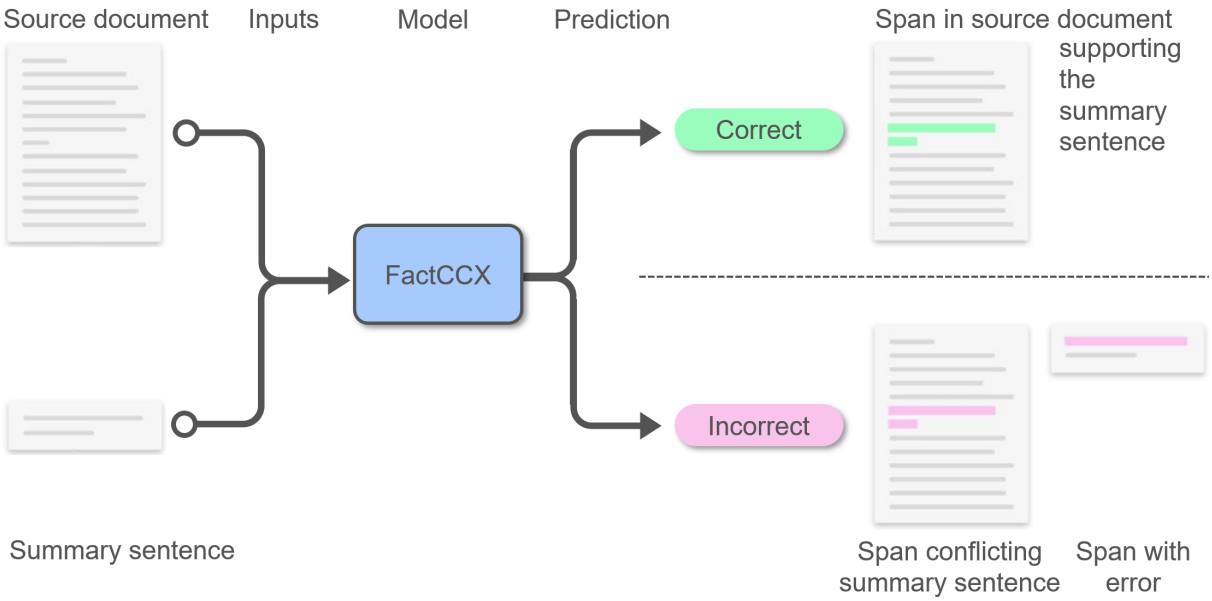

Figure 3.2: Illustration of the computation of FactCCX. Given a source document D and summary sentence s, the FactCCX model predicts either 'Correct' or 'Incorrect' and highlights a supporting span in the source document. If the summary sentence is unfaithful w.r.t the source document, the model additionally predicts the inconsistent span in the summary sentence.

sentence is classified as inconsistent. The classifier is trained on an artificially generated, weakly-supervised dataset consisting of adversarial examples, which are obtained by injecting noise. The authors generate this synthetic dataset using many transformations like paraphrasing, entity and number swapping, pronoun swapping and sentence negation. Please note that the training of this explainable model is only possible because of the synthetic dataset. This synthetic dataset contains metadata about the location of the extracted summary sentence in the source document. It also contains metadata about the locations in the summary sentence where the transformations were applied.

In mathematical terms, the FactCCX evaluator is a function $f : (D,s) \longrightarrow [0,1]$ where D is the source document and s is a summary sentence. Then, $f(D,s)$ represents the probability that s is factually consistent with respect to the source document D. We define the factual score of a summary S that consists of multiple sentences $s_1,...,s_n$ as follows:

$$f(D,S) = \frac{1}{n}\sum_{i=1}^{n} f(D,s_i) \tag{3.9}$$

Figure 3.2 illustrates the usage of FactCCX for predicting and explaining the factual correctness of a summary sentence s and the corresponding source document D. The source D and the summary s are concatenated and given to the BERT-based FactCCX model as input. Then, the binary classification model predicts either 'Correct' or 'Incorrect' and the span selection heads predict the span in the source document that supports this prediction. If the summary sentence is unfaithful, the span selection heads also predict the inconsistent span in the summary sentence.

Implementation details: We use the available code[2] and adopt it where necessary to utilize FactCCX for faithfulness evaluation in our experiments. Since we use the same dataset in one of our experiments, we can use the pre-trained checkpoint[3] provided by the original authors and do not have to train the model ourselves. However, please note that fine-tuning is necessary to use FactCCX to evaluate faithfulness on other datasets. Zhu et al. (2021) observe that the performance of FactCC degrades when it is fine-tuned on one summary dataset and used to evaluate models on another dataset.

In addition to the already existing and known faithfulness methods BERTScore, Question Generation & Question Answering, Entailment and FactCCX, we develop and test several other approaches to measure factual consistency.

3.5 Sentence Similarity

Idea: The basic intuition of the sentence similarity approach to measure faithfulness is that the information expressed in the summary should be the same as in the source document but paraphrased. Therefore, a summary sentence should be very similar to one or multiple important sentences.

One of the earliest approaches to extractive summarization based on sentence similarity is TextRank (Mihalcea and Tarau, 2004). TextRank operates on the sentence-level to generate an extractive summarization. First, a fully-connected graph is build where the vertices represent the sentences. Then, the edges are weighted based on their similarity which is computed using cosine similarity of sentence embeddings. Finally, the top N sentences after running the PageRank (Page et al., 1999) algorithm on this graph are selected as the summary.

Given a summary that was generated like this, the sentence similarity approach would work very well. It would find most original sentences as it basically just performs the reverse search. However, an extractive summary like that does not have faithfulness issues at all since it expresses exactly the same information as the source document. The sentences similarity approach would correctly return a faithfulness score of 1.0 for every such summary.

In this work, we exclusively deal with abstractive summaries and, therefore, the summaries can be written using very different wordings and formulations to express the same information. As a consequence, the sentence similarity approach has to successfully deal with highly paraphrased text detecting similar concepts expressed with different words on the one hand. On the other hand, it has to differentiate between similar and contrasting or contradicting information so that it can actually be used to score faithfulness. For example, the sentences “Joe Biden meets Angela Merkel” and “The president of the United States meets the German chancellor” have basically no common words but should be very similar, while the sentences “Albert Einstein died in 1955” and “Albert Einstein was born in 1955” share nearly all words but express a completely different information and should not be similar in terms of faithfulness. Obviously, these are very challenging requirements for the sentence similarity measure.

Computation: We propose the following strategy to asses the faithfulness of a summary using the sentence similarity approach. Figure 3.3 illustrates the computation for clarification. First, we apply sentence splitting to both the source document and the summary to obtain lists of

2. https://github.com/salesforce/factCC
3. https://storage.googleapis.com/sfr-factcc-data-research/factccx-checkpoint.tar.gz

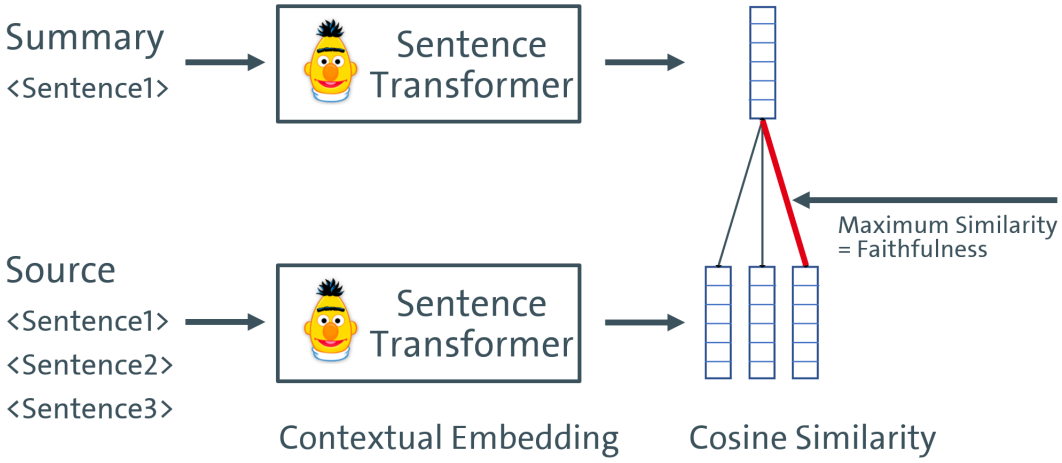

Figure 3.3: The computation process of the sentence similarity faithfulness metric. Sentences are embedded using a sentence transformer model and the faithfulness score is computed by aligning the sentences with a similarity metric. The top similarity score is the faithfulness score (for multiple summary sentences: the aggregated top similarity scores are the faithfulness score).

sentences. Next, we apply a sentence embedding method to get a vector representation for every sentence. Then, every summary sentence is matched with a source sentence to compute precision and every source sentence is matched with a summary sentence to compute recall using maximum matching: every sentence is aligned to the most similar sentence in the other document. The precision, recall and F1 variants of the sentence similarity metric are defined as follows: let $S = s_1, s_2, ..., s_3$ be the set of all summary sentences and let $D = d_1, d_2, ..., d_3$ be the set of all sentences of the source document.

$$R_{SS} = \frac{1}{|D|} \sum_{d_i \in D} \max_{s_j \in S} d_i^T s_j \quad P_{SS} = \frac{1}{|S|} \sum_{s_j \in S} \max_{d_i \in D} d_i^T s_j \quad F_{SS} = 2\frac{P_{SS} \times R_{SS}}{P_{SS} + R_{SS}} \tag{3.10}$$

This strategy functions basically like BERTScore that we described in Section 3.1, but operates on sentences instead of tokens.

Implementation details: Implementing this approach is straightforward and requires three components: a sentence splitting method, a sentence embedding model and a similarity metric.

We utilize the Spacy NLP toolkit to apply sentence splitting. The sentence segmentation component of Spacy offers multiple alternatives to detect sentence boundaries. These include dependency parsing, statistical sentence segmentation and rule based parsing. We use the default option which utilizes a dependency parser and provides the most accurate sentence boundaries.

We utilize different off-the-shelf sentences transformers to embed the sentences converting them to a vector representation. The Sentence BERT library[4] includes models that were trained on millions of paraphrase examples and, therefore, create good results for various similarity and retrieval tasks. The available models are based on a modification of the BERT network using Siamese and triplet networks and are able to derive semantically meaningful sentence embeddings (Reimers and Gurevych, 2019).

Another, similar approach to obtain sentence embeddings is to embed the sentence with a typical BERT-based model and use the embedding of the special begin-of-sentence [CLS] token as

4. https://www.sbert.net/index.html

sentence representation. More traditional approaches are to utilize Word2Vec or TF-IDF vectors. However, these traditional approaches produce embeddings regardless of context. Consequently, we expect that such traditional approaches perform poorly on the faithfulness task.

We compute the similarity of two sentence embeddings a and b with the cosine similarity, which is defined as follows:

$$\frac{a^T \cdot b}{|a| \cdot |b|} \tag{3.11}$$

In addition to comparing sentences using sentence embeddings and cosine similarity, we develop and test several other approaches to compare two texts. These similarity metrics are described in detail in Section 3.9.

3.6 Named Entity Recognition

Idea: To motivate this approach, we have to look at faithfulness at a more fine-grained level. Factual inconsistencies can occur at either the entity or the relation level. At entity level, we compare entities that appear in the source document and in the summary. The entity hallucination problem occurs when a model generated summary contains named entities that never appeared in the source document.

Intuitively, if a summary contains many entities that do not appear in the source document it is less faithful than a summary that contains the same entities as the source document. More matched entities imply a more faithful summary as this information is consistent between both texts. However, even a perfect match of summary entities with source entities does not guarantee a faithful summary. This approach specifically compares only entities and, therefore, does not capture various other aspects that could influence the faithfulness like relations between the entities or the context surrounding the entities. For example, the sentences “Albert Einstein received the Nobel Price” and “Albert Einstein did not receive the Nobel Price” share all entities despite expressing the opposite meaning.

Computation: We propose the following strategy to calculate the faithfulness of a summary based on named entities. For clarification, this process is visualized in Figure 3.4. First, we find all named entities in the source document and the summary document using an off-the-shelf tool for named entity recognition. We denote the set of named entities with $N(d)$ and $N(s)$ for the source document d and the summary s, respectively. Next, we group all found entities according to their label. For example, the sentence “Albert Einstein was born in Germany.” results in two groups "PERSON" and "LOC" having "Albert Einstein" and "Germany" as members. Then, we find for each named entity ne_s of the summary s the most similar entity ne_d of the same group in the source document d and the corresponding similarity score. This requires a similarity metric $sim(a,b)$ for named entities and can be formally denoted as:

$$similarNE(ne_s) = \arg\max_{ne_d \in N(d)} sim(ne_s, ne_d) \tag{3.12}$$

$$similarityScore(ne_s) = \max_{ne_d \in N(d)} sim(ne_s, ne_d) \tag{3.13}$$

Finally, the faithfulness score is the average over all similarity scores.

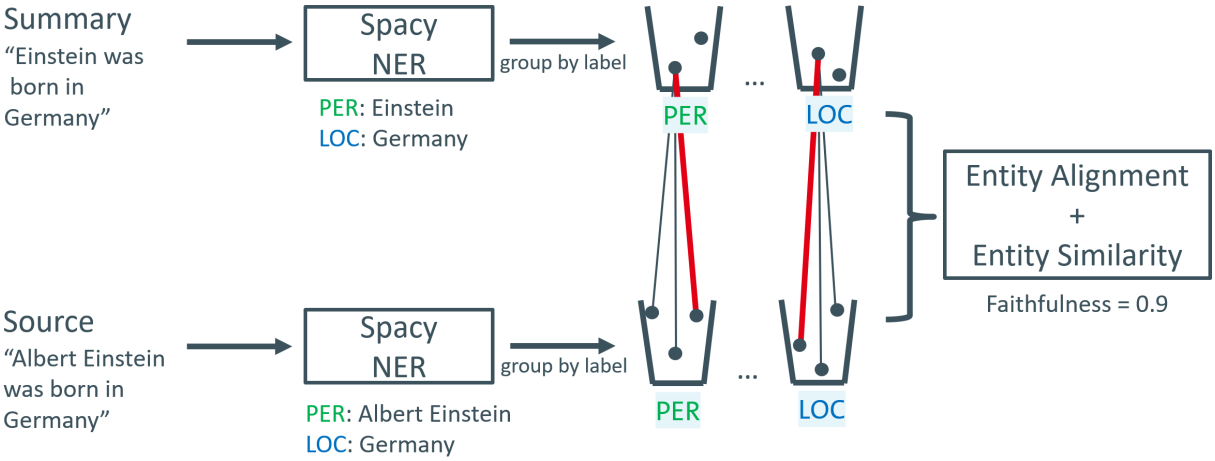

Figure 3.4: The computation process of the named entity recognition faithfulness metric. Entities are extracted with Spacy, grouped per label, aligned with a similarity metric and finally the faithfulness score is computed by aggregating the maximum similarities.

Implementation details: Implementing the above strategy requires two components: a tool for named entity recognition as well as a mechanic to compute the similarity between two named entities – or to be more general – between two phrases. We rely on the Spacy NLP toolkit that is equipped with a named entity recognition component to extract the named entities. We use the en_core_web_lg model for the best performance.

There are many possibilities to compute the similarity between two named entities. We test and develop several approaches to calculate the similarity between two texts which are described in detail in Section 3.9.

3.7 Open Information Extraction

Idea: To motivate this approach, we have to look at faithfulness at a more fine-grained level, again. As stated in the previous Section 3.6, factual inconsistencies can occur at either entity or the relation level. This time, we look at the relation level. At relation level, we compare the relations between entities that appear in the source document and the summary. The relation hallucination problem occurs when a model generated summary contains the same entities as the source document but the relations between these entities do not appear in the source document.

Compared to the approach based on named entity recognition, finding this type of inconsistency is much harder. This approach requires to identify named entities and find relations between these entities. Moreover, it is very beneficial to apply co-reference resolution. In this process, pronouns are resolved to their respective named entity and, as a result, more relations can be mapped to an entity.

Naturally, if a summary contains many relations that do not appear in the source document it is less faithful than a summary that contains the same relations. More matched relations imply a more faithful summary since not only the entities but also the interaction between these entities is consistent between both texts. In contrast to the named entity approach to faithfulness, a

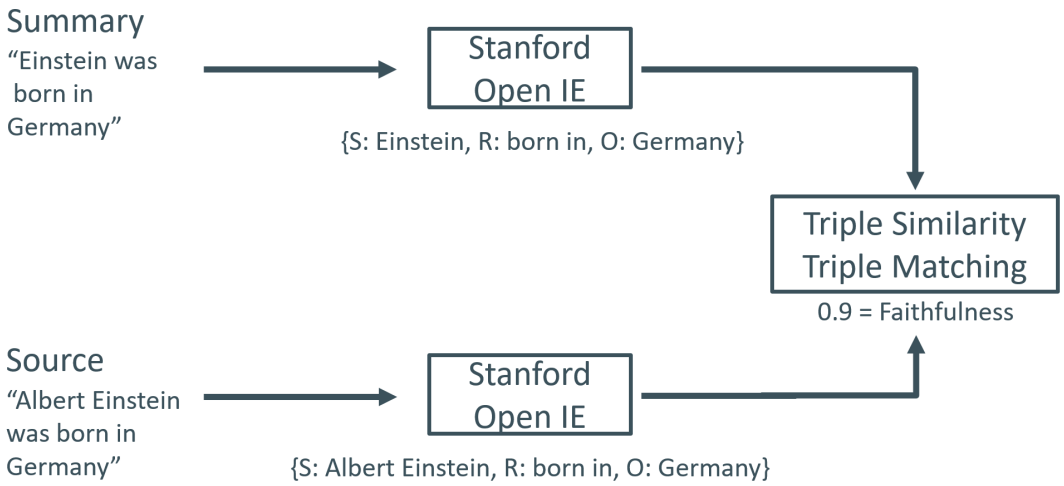

Figure 3.5: The computation process of the open information extraction faithfulness metric. Triples are extracted with Stanford Open IE and compared using either special relation matching metrics or any text similarity metric.

perfect match of summary relations with source relations can in theory guarantee a faithful summary. If all entities and all relations between these entities are consistent, there will be no information left that could be inconsistent. Matching relations is like matching facts and in case all facts are consistent, the summary is faithful per definition.

This may be true in theory, but in practice this requires the relation extraction component to find all entities as well as relations and the co-reference resolution component has to successfully map every pronoun to the correct entity. These are obviously very challenging requirements, especially when it is necessary to find all relations. Many relation extraction systems are trained on a pre-defined set of possible relations. While these systems are useful for many tasks, they do not fit our criteria of finding all relations in a given text. This leads us to Open Information Extraction (Open IE) systems that basically take the verb of the sentence as relation and, therefore, are theoretically able to capture all relations.

Computation: We propose the following strategy to calculate the faithfulness of the summary based on relation extraction. This process is illustrated in Figure 3.5 to increase clarity. First, we apply a co-reference resolution system to replace all pronouns in the texts with their respective entity. Next, we apply an Open IE system to find relation triples. Typically, these triples are of the form (<subject>, <verb>, <object>) and are able to represent any facts in the given texts. We denote the set of all summary triples as $R(s)$ and the set of all source document triples as $R(d)$. Finally, we compute the faithfulness score based on the comparison of the extracted relations which can be denoted as:

$$faithfulness = compare(R(s), R(d)) \tag{3.14}$$

We intentionally keep the compare function as generic as possible since there are many different ways to align, compare and evaluate the extracted relations some of which are described in the following paragraphs.

Implementation details: As outlined above, this approach needs three components: co-reference resolution, information extraction, and relation comparison. We use off-the-shelf models for co-reference resolution as well as Open IE to implement the open information extraction faithfulness metric. The Stanford CoreNLP toolkit offers a component for Open IE

based on the work of Angeli et al. (2015), which conveniently also includes an option to apply a co-reference resolution system as a pre-processing step. The toolkit offers many different variants and we decide to use the neural co-reference system by Clark and Manning (2016).

Unfortunately, the output of the Open IE component is different than expected and, therefore, we have to apply extra steps to implement our proposed strategy. Instead of extracting one triple per fact, the Open IE component extracts many triples. For example, the sentence "Albert Einstein was born in Germany" would result in the following triples which all express the same fact:

1. (Albert Einstein, born in, Germany)
2. (Einstein, born in, Germany)
3. (Albert, born in, Germany)

Having multiple triple representations for the same fact can be problematic once we start comparing the triple sets of the summary and source document to assess faithfulness. Imagine a summary that contains two facts A and B: fact A has five triple representations and fact B has only one representation. Now assume that the source document includes fact A. Comparing the triples matches the five triples of fact A of the summary with fact A of the source document but does not match the triple of fact B. A naive approach to calculate faithfulness is just counting the matched triples. This case results in $5/6$ matched triples which is equal to a faithfulness score of 0.83. Obviously, this score is problematic as one of two facts was unfaithful. We rather prefer a score of 0.5 in this scenario.

To solve this issue, we cluster the triples based on similarity. Then, we use the longest triple (when written as a sentence) as a cluster representative. We argue that the longest triple is the best representation since it contains the most information.

We experiment with various different ways to compare, align and score the two sets of triples which are outlined in the following.

The Relation Matching Rate (RMR) was introduced by Zhu et al. (2021). Let $R_s = \{(s_i, r_i, o_i)\}$ be the set of triples in the summary, and R_d be the set of triples of the source document. Then, each triple in R_s belongs to one of the following three categories:

1. Correct hit (C) $(s_i, r_i, o_i) \in R_d$
2. Wrong hit (W) $(s_i, r_i, o_i) \notin R_d$, but $\exists o' \neq o_i, (s_i, r_i, o') \in R_d$, or $\exists s' \neq s_i, (s', r_i, o_i) \in R_d$
3. Miss (M) otherwise

The authors define two variants of the RMR that measure the ratio of correct hits:

$$RMR_1 = 100 \times \frac{C}{C+W} \tag{3.15}$$

$$RMR_2 = 100 \times \frac{C}{C+W+M} \tag{3.16}$$

In addition to the RMR, we develop and test various other similarity metrics to compare two texts. To use these similarity metrics, the Open IE triples must be converted back to sentences. A triple (s, r, o) where s stands for subject, r for relation and o for object can be easily converted into a sentence by concatenating the subject, relation and object. For example writing the triple (Albert Einstein, born in, Germany) as a sentence results in the string "Albert Einstein born in Germany". The similarity metrics are explained in detail in Section 3.9.

3.8 Semantic Role Labeling

Idea: This semantic role labeling approach to assess faithfulness is heavily inspired by the YiSi metric (Lo, 2019), which was originally developed for machine translation. To the best of our knowledge, we are the first to test a variation of this metric for faithfulness.

YiSi measures the similarity between two sentences (originally between a machine translation and a human reference) by aggregating the semantic similarities of semantic structures. Therefore, a semantic parser is used to parse both sentences, a special matching algorithm is applied to align the semantic frames and finally a similarity measure is used to compare the matching predicates and role labels.

We believe that comparing semantic frames in contrast to comparing tokens as for example in BERTScore brings needed structure into the faithfulness assessment. One especially important property of this process is that it assures a similar argument structure of both summary and source or, in other words, it is able to identify crucial differences in the argument structure. Consider the following two sentences: “The man eats a fish for breakfast.” and “The fish eats a man for breakfast”. For the first sentence, a typical semantic role labeler assigns the label "ARG0" and "ARG1" to the phrases "The man" and "a fish", whereas the labels are swapped for the second sentence. As a result, this metric finds no matching phrase for both phrases which is a very desirable property when we consider faithfulness. In contrast to that, BERTScore aligns "man" with "man" and "fish" with "fish" not detecting the unfaithful information. In addition to that, this approach includes a subset of named entity recognition capabilities, which also support the faithfulness measurement. Commonly, semantic role labeling systems include labels for example for locations and time.

In short, using a semantic parser adds another layer of semantic meaning and semantic similarity to the faithfulness assessment which is very promising. Basically, this approach ensures that whole summary phrases are used in semantically similar way as in the source document and should help to identify cases where the summary derives from the originally intended meaning.

Computation: We propose the following strategy to calculate the faithfulness of a summary based on semantic role labeling. For clarity, this process is illustrated in Figure 3.6. First, we apply a semantic role labeling system to both the summary and the source document. Optionally, we filter and merge semantic role labels resulting in a smaller set of labels to increase the robustness. Next, we group all source and summary phrases by their label. We align source and summary phrases with the same label using a similarity metric. Finally, we compute the faithfulness score by aggregating the similarity scores of the aligned phrases. Formally, this calculation can be denoted as

$$alignment_{recall}(l) = \frac{1}{|P_{S,l}|} \sum_{p_i \in P_{S,l}} \max_{p_j \in P_{D,l}} sim(p_i, p_j) \tag{3.17}$$

$$alignment_{precision}(l) = \frac{1}{|P_{D,l}|} \sum_{p_j \in P_{D,l}} \max_{p_i \in P_{S,l}} sim(p_i, p_j) \tag{3.18}$$

$$alignment_{F1}(l) = 2\frac{alignment_{precision}(l) \times alignment_{recall}(l)}{alignment_{precision}(l) + alignment_{recall}(l)} \tag{3.19}$$

$$faithfulness_{metric} = \frac{1}{|L|} \sum_{l \in L} alignment_{metric}(l) \tag{3.20}$$

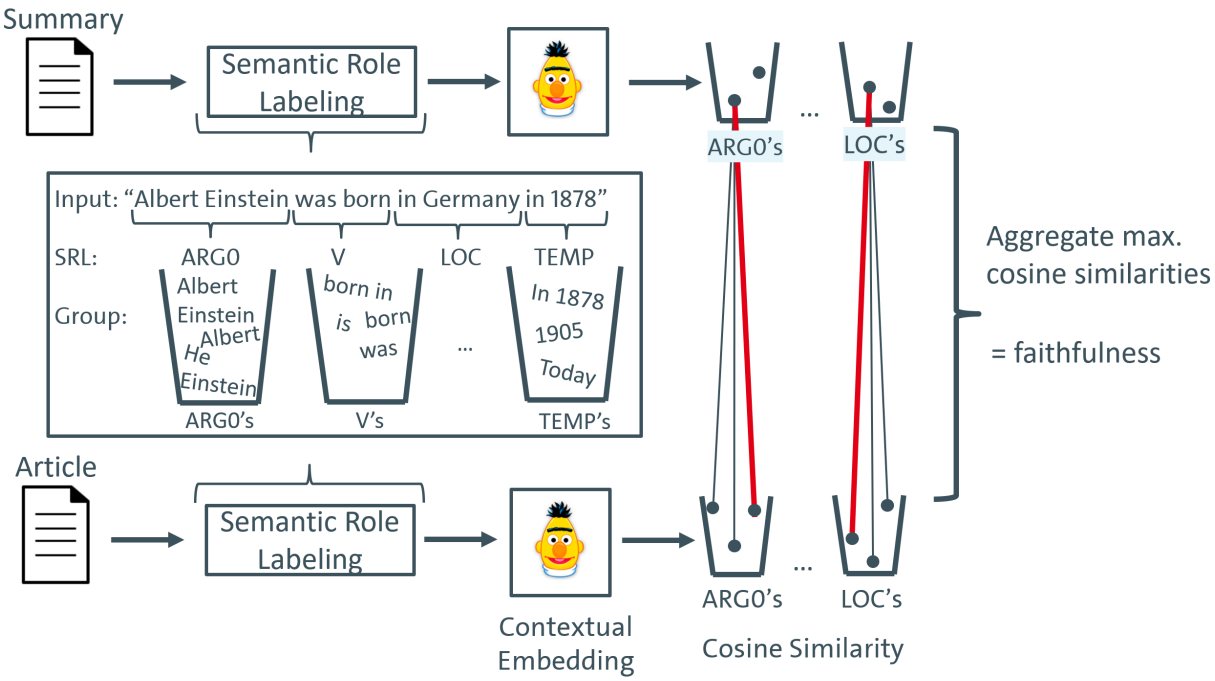

Figure 3.6: The computation process of the semantic role labeling faithfulness metric. Semantic roles are extracted with an of the shelf model, grouped per label and finally the faithfulness score is computed by aligning the phrases and aggregating the maximum similarities using a similarity metric.

where $metric \in \{precision, recall, F1\}$, L is the set of all semantic labels, sim is a similarity metric comparing two texts, $P_{D,l}$ and $P_{S,l}$ are sets of phrases with label $l \in L$ for source document D and summary S, respectively.

Implementation details: Two components are necessary to implement this approach: a semantic role labeling (SRL) model as well as a similarity metric that compares and aligns phrases. We use an off-the-shelf SRL model available in the AllenNLP (Gardner et al., 2018) toolkit which is based on SRL BERT (Shi and Lin, 2019). The SRL model is trained on the English OntoNotes 5 dataset (Hovy et al., 2006) for semantic role labeling.

As a result, the model is able to distinguish between many different labels. In our experiments, the SRL model outputs over 30 different labels. Similar to the YiSi authors Lo (2019), we merge the semantic role labels into more general role types (who, did, what, whom, when, where, why, how) for more robust performance. The mapping of the original OntoNotes labels to our reduced label set can be found in Appendix A.3.

As discussed above, we align the summary and source phrases using a similarity metric. We develop and test various text similarity measures which are explained in detail in the next section.

3.9 Similarity Metrics

The approaches to assess faithfulness of generated texts described in Sections 3.5 - 3.8 all extract very different information from the texts, but all methods eventually compare the extracted information with each other. Be it whole sentences, smaller phrases, named entities or Open

IE triples: the techniques to compare the structures are all the same as these structures are just text in the end. In this section, we describe various methods to compare two texts. While there are many more possibilities, we only focus on the following ones and evaluate them in the next Chapter 4.

In many cases it is necessary to align the texts first before comparing them. For example, the named entity recognition approach may output multiple named entities with the same label for both summary and source document. Let S be the set of summary texts and D denote the set of source texts to be compared. The goal is to align the elements $s \in S$ with the elements $d \in D$ and compute a similarity score of the aligned elements. Formally, this can be described as follows

$$alignment(s) = \arg\max_{d \in D} sim(s,d) \tag{3.21}$$

$$similarity(s) = \max_{d \in D} sim(s,d) = sim(s, alignment(s)) \tag{3.22}$$

where *sim* is a similarity metric that compares two texts.

When calculating the aggregated similarity between unaligned sets of summary and source texts, three different possibilities arise: calculating precision, recall or F1.

$$similarity_{precision}(S,D) = \frac{1}{|S|} \sum_{s \in S} \max_{d \in D} sim(s,d) = \frac{1}{|S|} \sum_{s \in S} similarity(s) \tag{3.23}$$

$$similarity_{recall}(S,D) = \frac{1}{|D|} \sum_{d \in D} \max_{s \in S} sim(s,d) \tag{3.24}$$

$$similarity_{F1}(S,D) = 2\frac{similarity_{precision} \times similarity_{recall}}{similarity_{precision} + similarity_{recall}} \tag{3.25}$$

Typically, we interpret the $similarity_{precision}(S,D)$ or $similarity_{F1}(S,D)$ as faithfulness score.

When the extracted information of source and summary are already aligned, it is sufficient to just compute the similarity between two texts. For example, in the Question Generation & Question Answering approach, the answers are already aligned and we only need to compute the similarity of these answers. In the next sections, we describe possible realizations of similarity metrics $sim(s,d)$.

Exact Match (EM)

Exact match is the simplest method to compare two texts, but also the most restrictive one being not able to match paraphrases. Exact match simply aligns identical text and can be formally denoted as:

$$EM(s,d) = \begin{cases} 1, & \text{if } normalize(s) = normalize(d) \\ 0, & \text{otherwise} \end{cases} \tag{3.26}$$

We apply additional heuristics $normalize(s)$ that include e.g. stopword filtering and lowercasing to both increase the number of matches and decrease false positives or noise (e.g. due to similar stopwords).

F1

In contrast to exact match, F1 surface similarity performs exact match on a token-by-token level. This similarity metric is also not able to match paraphrases, but it is at least not as strict as exact match allowing partial matches. E.g. "Obama" and "Barack Obama" have a similarity of 0.5. Formally, F1 similarity can be denoted as

$$precision(s,d) = \frac{1}{|s|} \sum_{t_s \in s} \max_{t_d \in d} EM(t_s, t_d) \tag{3.27}$$

$$recall(s,d) = \frac{1}{|d|} \sum_{t_d \in d} \max_{t_s \in s} EM(t_s, t_d) \tag{3.28}$$

$$F1(s,d) = 2\frac{precision(s,d) \times recall(s,d)}{precision(s,d) + recall(s,d)} \tag{3.29}$$

where $t_s \in s$ and $t_d \in d$ denote the tokens after whitespace tokenization of the summary text s and source text d. Again, we apply additional heuristics like stopword filtering and lowercasing before the tokenization for more robust performance.

BERTScore F1 (BS)

BERTScore is already explained in detail in Section 3.1. In short, BERTScore is computed very similarly to $F1$ with the only differences being:

1. Use a BERT wordpiece tokenizer instead of a whitespace tokenizer.
2. Represent tokens as vectors instead of the surface representation (plain text) obtained by embedding the text with BERT-based models.
3. Compare the tokens using cosine similarity instead of exact match.

Using word embeddings and especially contextual word embeddings (e.g. produced by BERT-based models) allows matching beyond exact match. This metric is able to compare and match paraphrases as well as capture distant dependencies between tokens. For example, the terms "Barack Obama" and "president of the US" have a BERTScore F1 similarity of 0.755 using RoBERTa-embeddings.

The cosine similarity between two vectors v and u is defined as follows:

$$cosine_{similarity}(u,v) = \frac{u^T \cdot v}{|u| \cdot |v|} \tag{3.30}$$

In this work, we calculate the BERTScore F1 similarity using the 8th layer of the RoBERTa-large model trained on the Multi-NLI dataset (Williams et al., 2018) to calculate the token embeddings.

Sentence similarity (SS)

The sentence similarity approach is already explained in detail in Section 3.5. To sum up, the sentence similarity approach is similar to BERTScore but instead of comparing token embeddings, embeddings of the whole text are compared using cosine similarity. Basically, this approach first computes an embedding representation of both texts and then calculates the cosine similarity of the two embeddings to assess the similarity of two texts. In this work, we utilize the pre-trained paraphrase-distilroberta-base-v1 model from the Sentence BERT library[5] to compute sentence embeddings.

Formally, the sentence similarity approach can be denoted as

$$ss(s,d) = cosine_{similarity}(embed(s), embed(d)) \tag{3.31}$$

where $embed(x)$ creates a vector representation of the text x using the sentence embedding model, s is a summary text and d is a source text.

Since the sentence transformer models also calculate contextual sentence embeddings, the sentence similarity and BERTScore F1 metric share the benefit of being able to match paraphrases. The paraphrase-distilroberta-base-v1 model is especially good for similarity and paraphrase matching tasks as it was trained on millions of paraphrase examples.

5. https://www.sbert.net/index.html

4 Evaluating faithfulness metrics

In this chapter, we answer the first research question: "Which faithfulness metric is the best?". In order to answer this question, we evaluate all faithfulness metrics described in Chapter 3 on a dataset that contains human faithfulness judgments and compute the correlation.

The first sections describes the dataset. Next, we show the final results in a leaderboard and, therefore, already answer the research question. The other sections explain various different experiments that describe how we tweak the faithfulness metrics to achieve the top scores.

4.1 Dataset

We use the dataset collected by Maynez et al. (2020) that contains human faithfulness judgements for document-summary pairs and treat the human faithfulness scores as ground-truths. The summaries of this dataset are generated by four different abstractive summarization models trained on the XSUM (Narayan et al., 2018) dataset: pointer-generator network (See et al., 2017), a transformer based model (Vaswani et al., 2017), BERT (Devlin et al., 2019) – a pre-trained transformer based model – and a topic-aware convolutional model (Narayan et al., 2018).

The XSUM dataset comprises about 200,000 British Broadcasting Corporation (BBC) articles paired with single-sentence summaries provided by the journalists that authored the article. A gold summary is either an introductory sentence that prefaces a news article or the first sentences of it. The rest of the news article is the source document. As a result, XSUM summaries are significantly more abstractive than e.g. summaries of the CNN/DM dataset and extractive summarization methods are not applicable. A Wang et al. (2020) observe that the summaries frequently contain facts that cannot be found in the source document, which makes this dataset especially challenging. Also, Durmus et al. (2020) find that models trained on XSUM are more abstractive. Abstractive summarization models generate summaries mostly by paraphrasing the input content and, thus, introduce novel words and novel n-grams. This increases the chance of generating unfaithful text making this dataset ideal to evaluate faithfulness metrics.

The XSUM hallucination dataset collected by Maynez et al. (2020) consists of 2000 document-summary pairs obtained by randomly sampling 500 articles from the XSUM test set and applying the previously mentioned four summarization models. Then, the authors trained annotators and prepared them by conducting two pilot studies to ensure a good annotation quality. Finally, three annotators per document-summary pair were given the task to identify text spans (hallucination spans) in the summary that are unfaithful to the article. Given the annotations of three annotators, the faithfulness score of a document-summary pair was calculated as follows: "We map the hallucination spans for each summary to word level. We assign a score of 1.0 to each word if it is not in one of the hallucination spans marked by an annotator. Finally we take the average over the number of annotations (3) and the number of words in the summary to get the final faithfulness score for each summary."[1]

1. copied from https://github.com/google-research-datasets/xsum_hallucination_annotations

Method	Pearson (r)	Spearman (p)
BERTScore	**0.501**	**0.486**
Entailment	0.366	0.422
Sentence similarity	0.392	0.389
Semantic role labeling	0.393	0.377
Named entity recognition	0.252	0.259
Question generation + answering	0.252	0.258
Open information extraction	0.169	0.185

Table 4.1: Pearson (r) and Spearman (p) correlation coefficients for faithfulness measured between human faithfulness judgements and different automatic methods.

4.2 Leaderboard

We compare the predictions of all faithfulness metrics described in Chapter 3 against the human faithfulness judgements of the XSUM hallucination dataset. First, we apply a faithfulness metric on all document-summary pairs. Then, we calculate Spearman correlation (p) and Pearson correlation (r) coefficients between human judgements and the predicted faithfulness scores. Results are reported in Table 4.1. Please note that this table reports only the best scores for every faithfulness method to give an overview.

BERTScore achieves the highest correlation with human judgements. The entailment, sentence similarity and semantic role labeling metrics perform very similarly. Interestingly, the named entity and the question generation & question answering (QGQA) approach have nearly identical scores. This suggests that the QGQA method is basically just an overly complicated way of comparing named entities. The method based on open information extraction is the last one in this ranking. The poor performance can be explained by the fact that the used models are relatively old. Furthermore, this approach needs a co-reference resolution component and both, co-reference resolution as well as open information extraction are difficult tasks with a lot of room for improvements. We exclude the FactCCX model from this ranking as the predictions have no correlation with the human judgements at all. The model is trained on the CNN/DailyMail dataset, but we use it to evaluate summaries from the XSUM dataset. We empirically find that the performance of FactCCX declines when it is fine-tuned on one dataset and used to evaluate models on another. Zhu et al. (2021) make similar observations when using FactCC for evaluation.

This experiment revealed that BERTScore is the best metric to evaluate the faithfulness of a summary, answering Research Question 1. In the next sections, we describe additional experiments explaining how we tweak the different metrics to achieve the best scores.

Embedding technique	Method	Pearson (r)	Spearman (p)
sentence-by-sentence	**bert_precision**	**0.501**	**0.486**
sentence-by-sentence	bert_recall	0.220	0.226
sentence-by-sentence	bert_F1	0.482	0.470
first 512 tokens	bert_precision	0.469	0.462
first 512 tokens	bert_recall	0.101	0.097
first 512 tokens	bert_F1	0.379	0.375
sliding windows	bert_precision	0.475	0.468
sliding windows	bert_recall	0.100	0.092
sliding windows	bert_F1	0.382	0.378

Table 4.2: Comparison of different embedding techniques to deal with the input token limit of 512 tokens of the RoBERTa model used to calculate BERTScore.

4.3 Different embedding techniques for BERTScore

The XSUM faithfulness dataset contains about 650 examples where the source document is longer than 512 tokens. This is problematic, as the RoBERTa model can only process up to 512 tokens at once. These long examples make up about 33% of the whole dataset and, therefore, have to be handled correctly.

We experiment with three different approaches to solve the token limitation problem.

sentence-by-sentence: Instead of embedding a whole document at once, we split the document into sentences and embed each sentence seperately. The dataset does not contain a single sentence that is longer than 512 tokens. We utilize the Spacy NLP toolkit to perform sentence splitting using the en_core_web_lg model.

first 512 tokens: As the name suggest, we only embed the first 512 tokens instead of embedding the whole document. The remaining tokens of the document are discarded.

sliding windows: We create multiple windows with a width of 512 tokens, where the first 128 and last 128 tokens overlap with the previous and next window, respectively. Embedding one window effectively only embeds the 256 tokens in the center. The overlapping tokens serve as additional context and their embeddings are discarded.

The results are shown in Table 4.2. The best correlation with human faithfulness judgements is achieved by the sentence-by-sentence approach. The other two approaches perform slightly worse.

IDF-Weighting	Method	Pearson (r)	Spearman (p)
no	**bert_precision**	**0.501**	**0.486**
no	bert_recall	0.220	0.226
no	bert_F1	0.482	0.470
yes	bert_precision	0.487	0.477
yes	bert_recall	0.243	0.253
yes	bert_F1	0.476	0.464

Table 4.3: Comparison of BERTScore correlation with human faithfulness judgements when applying IDF-weighting.

4.4 Weighting tokens by importance for BERTScore

BERTScore is calculated by aligning every token of one text (e.g. the summary) with the corresponding most-similar token of the other text (e.g. the source document) according to the cosine similarity. Then, the average of these maximum similarities is the final BERTScore. However, instead of simply taking the average, we can calculate a weighted average by weighting every token with its inverse-document frequency (idf).

We tokenize the whole dataset $D = d_1, d_2, ...d_N$ using the corresponding tokenizer of the RoBERTa model. Then, we calculate the idf for every token t as follows:

$$idf(t) = \log N - \log \sum_{i=1}^{N} \mathbb{1}[t \in d_i] \tag{4.1}$$

The results of applying IDF-weighting vs. not applying IDF-weighting are shown in Table 4.3. Unfortunately, IDF-weighting does not improve the performance when using BERTScore to assess the faithfulness.

4.5 Varying the input for textual entailment models

In Section 3.2, we explain three different techniques to utilize textual entailment models to assess faithfulness: sentence-to-sentences (s2s), document-to-sentence(d2s) and top-to-sentence (top2s). In this section, we shortly recap these techniques and then evaluate which technique works best for the given task.

sentence-to-sentence: An entailment model predicts the probability that sentence *b* is entailed by sentence *a*. Given that the source document and the summary consist of multiple sentences, we calculate the entailment probability between all pairwise sentence combinations. Next, we select the highest entailment probability for each summary sentence. We would average over all entailment probabilities to calculate the final entailment score, however, the dataset contains only one-sentence-summaries.

document-to-sentence: Instead of predicting the probability that sentence *b* is entailed by sentence *a*, we feed the entailment model with the whole source document and the summary sentence *b* as input. Thus, the model predicts the probability that sentence *b* is entailed by the

Sentence similarity model	Entailment model	Method	Pearson (r)	Spearman (p)
-	bart-large-mnli	s2s	0.152	0.190
-	bart-large-mnli	d2s	0.368	0.399
distilroberta-base	bart-large-mnli	top2s	0.251	0.302
-	roberta-large-mnli	s2s	0.178	0.157
-	roberta-large-mnli	d2s	0.168	0.116
distilroberta-base	roberta-large-mnli	top2s	0.301	0.279
-	bart-large-mnli_512	s2s	0.152	0.190
-	**bart-large-mnli_512**	**d2s**	**0.366**	**0.422**
distilroberta-base	bart-large-mnli_512	top2s	0.251	0.302

Table 4.4: Comparison of the correlation of various entailment models using different input techniques with human faithfulness judgements.

whole source document. Again, we would calculate the average entailment probability over all summary sentences, however, the dataset only contains one-sentence-summaries.

top-to-sentence: Given the summary sentence, we find the top k ($k = 3$) most similar sentences in the source document. We utilize the paraphrase-distilroberta-base-v1 pre-trained model from the Sentence BERT library[2] to obtain sentence embeddings for every source and summary sentence. Then, we calculate pairwise cosine similarities to find the k most similar sentences. Finally, we use the entailment model to predict the probability that the summary sentence is entailed by the top-k most similar source sentences.

In addition to experimenting with different input techniques, we try two different entailment models, RoBERTa-large and BART-large, which are both trained on the Multi-NLI dataset (MNLI, Williams et al., 2018). Since RoBERTa has a token limitation of 512 tokens and BART can deal with up to 1024 input tokens, we include BART512 where the input is also limited to 512 tokens for a fair comparison.

The results of this experiment are shown in Table 4.4. The BART models perform better when using the document-to-sentence approach, while the RoBERTa model achieves better performance using the top-to-sentence technique. Interestingly, the best performance is achieved by BART512 using the document-to-sentence approach. This suggests that models with a long context (> 512 tokens) are not necessary for this dataset.

4.6 Component analysis of the QGQA framework

The question generation & question answering framework (QGQA) utilizes three components to assess the faithfulness of a summary: question generation (QG), question answering (QA) and answer similarity. In this section, we compare different QG models, QA models and answer similarity metrics to find the best setup.

In the first experiment, we compare two question generation models while fixing the QA model to RoBERTa-base which is trained on the SQuAD2 dataset. Both QG models, T5-base and T5-small, are trained on the SQuAD2 dataset as well. However, the QG models are trained with

2. https://www.sbert.net/index.html

QG Model	Pearson (r)	Spearman (p)
t5-base	**0.219**	**0.229**
t5-small	0.183	0.198

Table 4.5: Comparison of question generation models of different quality used in the QGQA framework to judge the faithfulness of summaries. The table lists the correlation with human faithfulness judgements.

the reverse objective: instead of predicting the answers, the models are trained to predict the question given the context and the answer.

Results are shown in Table 4.5. Unsurprisingly, the T5-base model has a better performance than T5-small. However, the difference between these two models is noticeable suggesting that improving the question generation model can have a noticeable impact on the final performance of the QGQA framework.

In the second experiment, we compare three question answering models (RoBERTa-large, RoBERTa-base, BERT-large) that are trained on the SQuAD2 dataset while fixing the QG model to T5-base as we previously found out that T5-base is the best-performing QG model. Results are shown in Table 4.6. The best QA model is RoBERTa-large outperforming BERT-large by a small margin.

QA Model	Pearson (r)	Spearman (p)
roberta-large-squad2	**0.228**	**0.258**
roberta-base-squad2	0.219	0.229
bert-large-uncased-whole-word-masking-squad2	0.182	0.203

Table 4.6: Comparison of question answering models of different quality used in the QGQA framework to judge the faithfulness of summaries. The table lists the correlation with human faithfulness judgements.

For the last experiment, we use RoBERTa-large-squad2 as the question answering model, T5-base as the question generation model and vary the answer similarity metrics. We use and compare Exact Match (EM), BERTScore (bert), F1 and Sentence Similarity (SS) to determine the similarity of answers. The details of these similarity metrics are explained in Section 3.9.

Results are listed in Table 4.7. The best metric to compare the similarity of two answers is the precision variant of BERTScore. Interestingly, Exact Match performs quite good in comparison. F1 has almost the same performance as precision BERTScore while being way faster to compute as well as not being a model-based metric. We decide to use F1 score in our QGQA framework as it is already quite slow and adding a third model to the pipeline that barely improves the performance is not convincing.

We conclude these three experiments with the observation that the QGQA framework is fairly robust to the choice of the single components. Analyzing these scores alone, we believe that the largest performance increase can be achieved by improving the question generation model. In Sections 5.2 and 5.3, we perform a qualitative analysis of various faithfulness metrics including QGQA that grants further insights. Our final QGQA framework consists of the question generation model T5-base, the question answering model RoBERTa-large-squad2 and the answer similarity metric F1.

Answer similarity metric	Pearson (r)	Spearman (p)
F1	0.228	0.258
bert_recall	0.185	0.194
bert_F1	0.226	0.235
bert_precision	**0.252**	**0.258**
EM	0.200	0.226
SS	0.216	0.222

Table 4.7: Comparison of different answer similarity metrics used in the QGQA framework to judge the faithfulness of summaries. The table lists the correlation with human faithfulness judgements.

4.7 Comparing entities, sentences, semantic roles and fact triples

Many faithfulness methods introduced in Chapter 3 extract different kind of information from the text, but in the end, compare and align the extracted structures using a similarity metric. The named entity metric extracts entities, the sentence similarity metric operates on sentences, the open information extraction metric extracts fact triples and the semantic role labeling approach extracts phrases. Since all methods extract text, we can eventually compare the extracted information with the same metrics. In this section, we analyze which similarity metric works best for which faithfulness method.

We use and compare Exact Match (EM), BERTScore (bert), F1 and Sentence Similarity (SS) to align and calculate the similarity between the texts. The details of these similarity metrics and how they can be used to align texts are explained in Section 3.9.

The results are shown in Table 4.8. Please note that there are actually three variants of BERTScore and Sentence Similarity: precision, recall and F1. However, we observe that the precision variant always performs best. For this reason and for increased readability, we omit these scores from the table.

The F1 similarity metric performs best for all faithfulness methods but semantic role labeling. The precision variant of the model-based sentence similarity metric achieves the best scores for semantic role labeling. We are surprised to observe that the simple F1 similarity metric works so well for this task. Initially, we believed that being able to deal with and successfully compare paraphrases, synonyms etc. is a very important criterion for a similarity metric that assesses faithfulness. Especially when comparing simple metrics like Exact Match and F1 with very sophisticated model-based metrics like BERTScore that rely on contextualized embeddings, we thought their ability to align and score paraphrases would easily outperform these simple metrics. However, this experiment suggests that – more often than not – the model-based metrics assign high similarity scores to texts one would not consider similar in the context of faithfulness.

4.8 Grouping semantic role labels

The semantic role labeling (SRL) approach to faithfulness first extracts and groups phrases with the help of a SRL model. Then, it aligns and compares these phrases using a similarity metric. In this work, we utilize the SRL BERT (Shi and Lin, 2019) model for semantic role labeling and

Faithfulness method	Similarity metric	Pearson (r)	Spearman (p)
Named entity recognition	EM	0.251	0.255
Named entity recognition	**F1**	**0.252**	**0.259**
Named entity recognition	ss_precision	0.200	0.204
Named entity recognition	bert_precision	0.151	0.195
Semantic role labeling	EM	0.234	0.273
Semantic role labeling	F1	0.359	0.363
Semantic role labeling	**ss_precision**	**0.393**	**0.377**
Semantic role labeling	bert_precision	0.270	0.344
Sentence similarity	EM	-0.039	-0.039
Sentence similarity	**F1**	**0.392**	**0.389**
Sentence similarity	ss_precision	0.387	0.369
Sentence similarity	bert_precision	0.374	0.372
Open information extraction	EM	0.042	0.076
Open information extraction	**F1**	**0.169**	**0.185**
Open information extraction	ss_precision	0.134	0.186
Open information extraction	bert_precision	0.013	0.212

Table 4.8: Comparison of different answer similarity metrics used in various faithfulness metrics that judge the faithfulness of summaries. The table lists the correlation with human faithfulness judgements.

the paraphrase-distilroberta-base-v1 model of the Sentence BERT library to assess the similarity of phrases.

The SRL BERT model is trained on the OntoNotes 5 dataset (Hovy et al., 2006). This dataset contains many different semantic role labels. For instance, labeling all examples of the XSUM hallucination dataset with this model results in over 30 different labels.

We group the labels into more general role types (who, did, what, whom, when, where, why, how). In this experiment, we compare the standard variant of the SRL faithfulness metric with the variant that uses a reduced, grouped label set to quantify the performance gain. The mapping of the original label set to our reduced label set is listed in Appendix A.3.

The results are shown in Table 4.9. We can observe a small performance gain when using the reduced label set suggesting that the grouping is good and increases the robustness of the model.

Reduced label set	Method	Pearson (r)	Spearman (p)
no	ss_precision	0.371	0.354
no	ss_recall	0.199	0.193
no	ss_F1	0.281	0.272
yes	**ss_precision**	**0.393**	**0.377**
yes	ss_recall	0.246	0.230
yes	ss_F1	0.328	0.312

Table 4.9: Comparison of the standard variant of the SRL faithfulness metric that uses the whole label set and the improved variant that uses a reduced label set. The table lists the correlation with human faithfulness judgements.

4.9 Ensembling multiple faithfulness metrics

In this work, we develop and evaluate many different faithfulness metrics and find that BERTScore is the best-performing one. However, what happens if we create an ensemble of faithfulness metrics, i.e. combining multiple methods, to assess the factual correctness of a text? While this approach may be unfeasible in practice, e.g. due to very long evaluation time, it is still interesting to research. Therefore, we combine three faithfulness metrics into a linear ensemble in this experiment. We decide to use BERTScore, Entailment and the QGQA framework for this experiment as these three faithfulness metrics are already public and known in the research community.

We create a simple linear combination of the faithfulness metrics as follows:

$$faithfulness(s|d) = w \cdot BERTScore(s|d) + x \cdot Entailment(s|d) + y \cdot QGQA(s|d) + z \quad (4.2)$$

where s denotes a summary and d the corresponding source document.

We use simple linear regression to find the following coefficients:

$$w = 1.30 \quad x = 0.25 \quad y = 0.24 \quad z = -0.3$$

We also evaluate this ensemble model on the XSUM faithfulness dataset and achieve a Spearman correlation coefficient of 0.561 as well as a Pearson correlation coefficient of 0.556. These results are the best we have seen on this data set so far.

5 Qualitative analysis of faithfulness metrics

In this chapter, we provide answers to the second research question: "How do faithfulness metrics perform in practice?". The first section describes our annotation tool FaithAnno that we specifically develop to support us with the qualitative analyzes of this chapter. In the next section, we conduct an annotation experiment using FaithAnno with the goal to assess the potential of different faithfulness approaches. We investigate how much the performance would improve if a faithfulness metric had a component with human-like performance. In the final section, we perform an error analysis of several faithfulness metrics demonstrating problems and drawbacks while also suggesting some possible improvements.

Please note that we focus on BERTScore, Entailment and the Question Generation & Question Answering (QGQA) framework in this chapter. These faithfulness metrics have good correlation with human faithfulness judgements as shown in Chapter 4, are already public and, therefore, are known by the research community, in contrast to other faithfulness metrics developed in this work.

5.1 FaithAnno

FaithAnno is an annotation tool specifically designed to support qualitative analyzes of faithfulness metrics. The front-end web-application is implemented with React and Bootstrap. The back-end consists of a PostgreSQL database to store user data, documents as well as annotations. Furthermore, it consists of Keycloak, which is an open-source solution for managing users and access control, and Hasura, which automatically provides an easy-to-use GraphQL API to access and modify the data stored in the SQL database. The architecture is visualized in Figure 5.1.

We bundle all components in a Docker container to easily deploy the application. The source code is available on GitHub[1] and the docker container is available on Docker Hub[2].

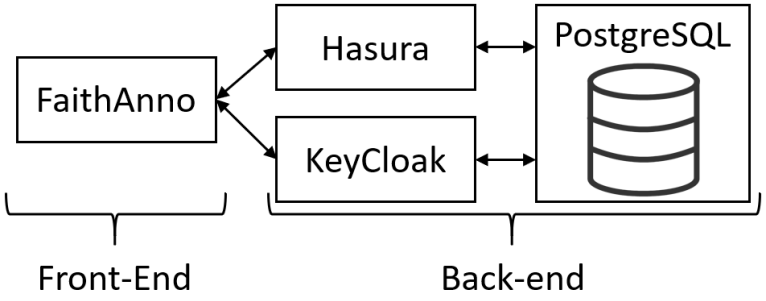

Figure 5.1: The architecture of FaithAnno. The front-end communicates with Hasura to access the data stored in the PostgreSQL database and it communicates with Keycloak to manage user access.

1. https://github.com/bigabig/faithanno
2. https://hub.docker.com/r/bigabig/faithanno

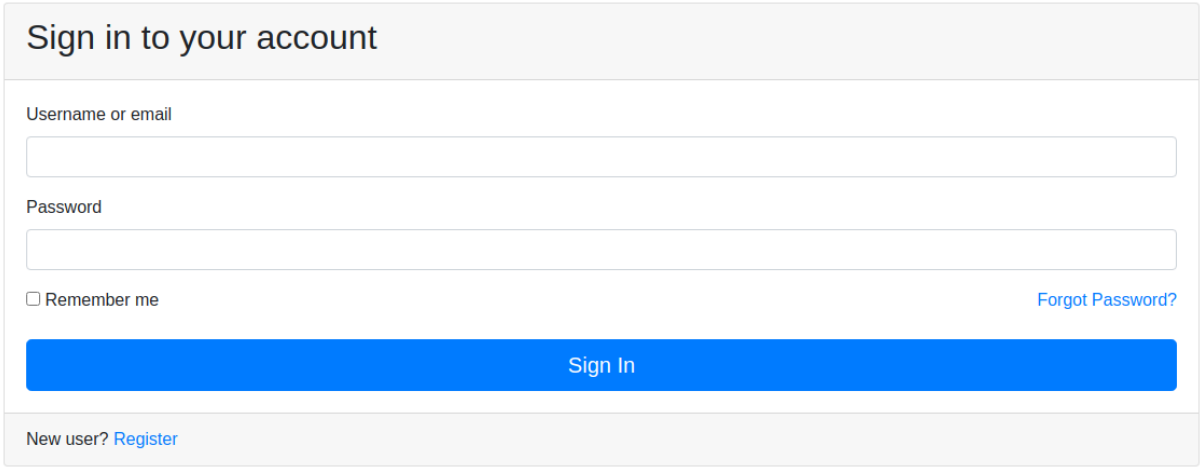

Figure 5.2: The login screen of FaithAnno. It has standard functionalities like "Forgot Password" and "Registration".

FaithAnno has three different views that are explained in the following: login, dashboard and annotation.

Before being able to access FaithAnno, it is necessary to login. The login screen is shown in Figure 5.2. It is a typical login similar to any other website with standard features like "Forgot Password" and "Registration". We decide to restrict the access to FaithAnno – even disabling the option to register – to be able to manually manage the annotators and assign specific documents to them.

After login, a user is greeted by the dashboard which is shown in Figure 5.3. The dashboard lists all documents (source-summary pairs) assigned to the user. Clicking on the "Continue annotation" button takes the user to the next un-annotated document, whereas clicking on the "Annotate" button in the list view redirects the user to the corresponding document. Both buttons open the annotation view. The dashboard also shows the number of remaining annotations.

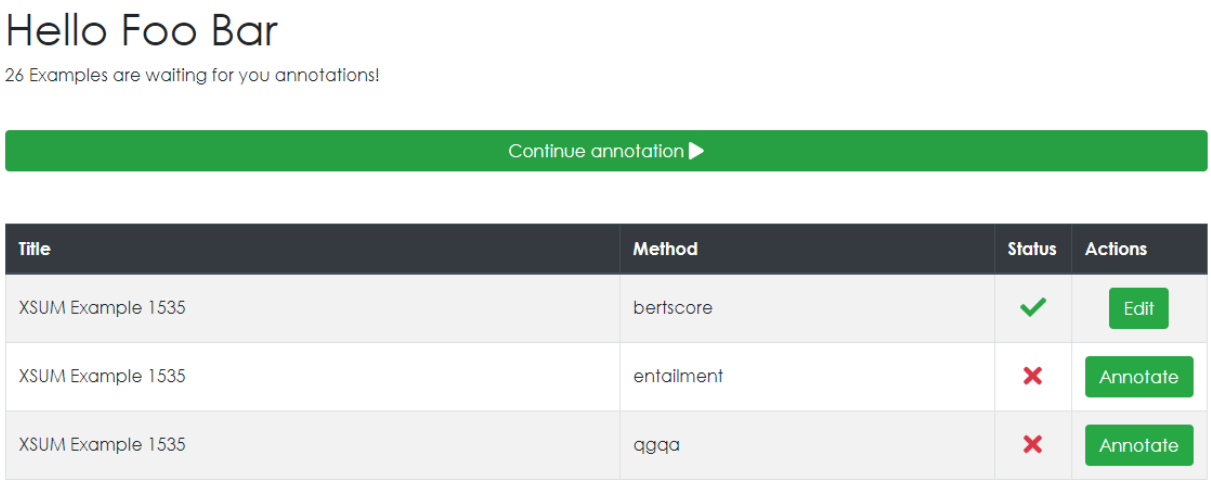

Figure 5.3: The dashboard of FaithAnno. It lists the status of the annotator's assigned documents, tracks the current progress and allows to select the next document to annotate.

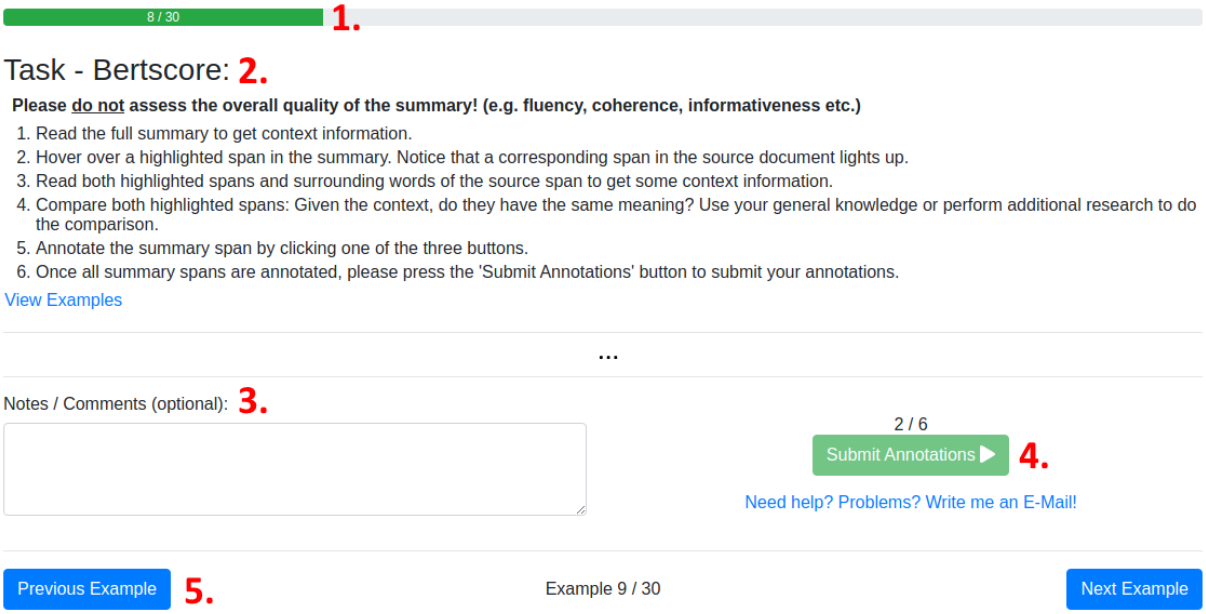

Figure 5.4: The common components of the annotation view of FaithAnno. It tracks the progress (1), shows a task description (2), allows to submit optional comments (3) as well as the annotations (4) and enables to navigate between the examples (5).

The annotation view follows the same structure for every faithfulness metric, however the main annotation component is different. Figure 5.4 shows the common elements of the annotation view. The progress bar at the top (1) visualizes the overall progress. In this particular case 8 out of 30 examples are already annotated. The task description (2) lists the steps an annotator should follow to perform the annotation. This guide is different for every faithfulness metric. Clicking on "View Examples" opens a modal window that demonstrates ten fully annotated examples. These examples explain some pitfalls and edge-cases as well as help annotators to get an idea of how to annotate. The "Notes / Comments" input field (3) allows annotators to provide extra information. The "Submit Annotations" button (4) is disabled initially, but gets automatically enabled as soon as all annotations are made. In this case, four additional annotations are required to enable the submission. Clicking this button stores the annotated data in the database. As soon as an annotator has done all annotations, the button gets enabled and has to be clicked to save the annotated data. The buttons "Previous Example" and "Next Example" (5) allow annotators to quickly navigate between the examples.

Annotation components

In this section, the main annotation component is explained for every faithfulness metric. We develop "explainability features" that give insights on how a certain faithfulness metric arrived at it's predicted faithfulness score. These explainability features are visualized in the annotation component and are very important for the annotation experiment explained in Section 5.2. The explainability features are also especially helpful for the error analysis in Section 5.3.

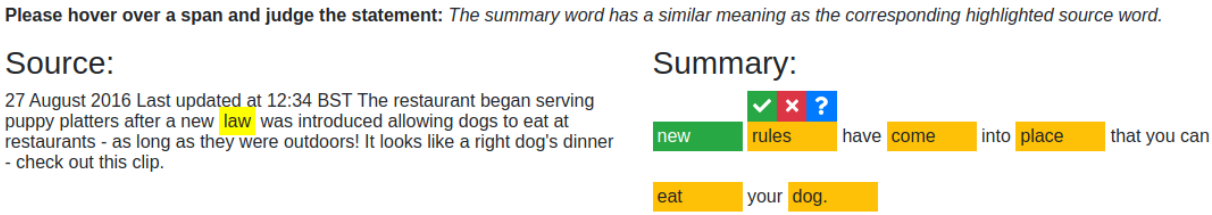

Figure 5.5: The BERTScore annotation component of FaithAnno. It highlights the corresponding source word when hovering over a summary word and enables per-word annotations.

BERTScore: Please recall that the precision variant of BERTScore matches every token of the summary to the most similar token in the source document. We apply stop-word filtering and highlight the remaining summary tokens with an orange background color. Hovering over such a summary word highlights the corresponding most similar word in the source document. This functionality is visualized in Figure 5.5. In addition to that, three annotation options are presented to the annotator: "Correct", "Incorrect" and "Don't know"; symbolized by a check, cross and question mark, respectively. This allows a quick and easy comparison of the aligned words. Furthermore, with the help of this functionality, it is possible to comprehend and analyze on which words the summarization model has paid attention to and, thus, it is possible to understand how the summary was generated.

Entailment: The top-to-sentence variant of the Entailment faithfulness metric first finds the top three most similar source sentences given the summary sentence and then calculates the entailment probability given both the top source sentences and the summary sentence. FaithAnno highlights the top three source sentences. This functionality is demonstrated in Figure 5.6. The feature enables to judge the faithfulness of a summary sentence without having to read through the whole source document which in some cases – depending on the length of the source document – can drastically reduce the time needed for the annotation. A scale ranging from "None" to "All" allows to annotate how many summary facts can be retrieved in the highlighted source sentences.

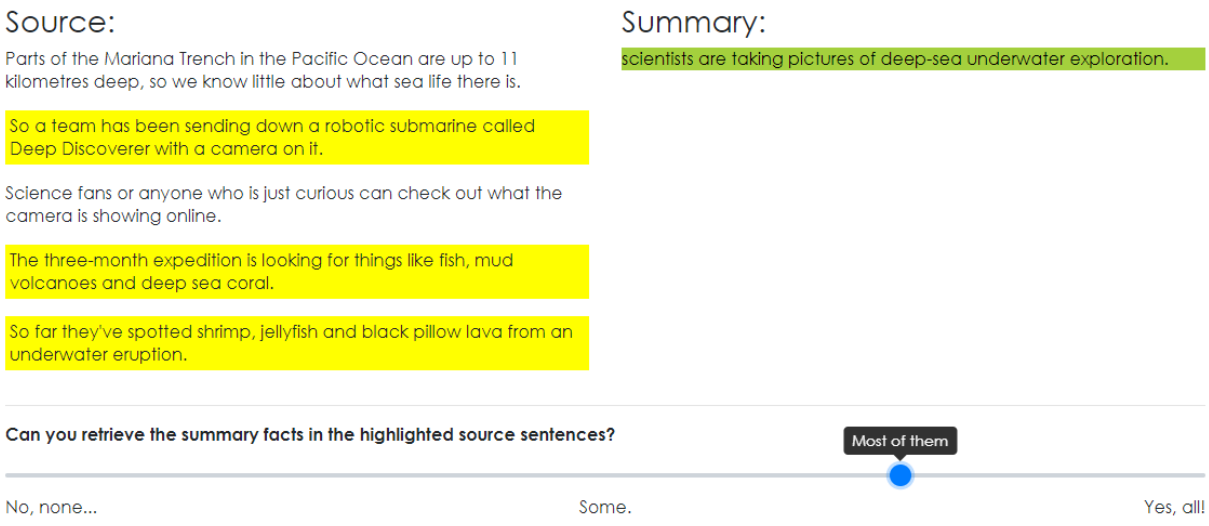

Figure 5.6: The Entailment annotation component of FaithAnno. It highlights the three most important source sentences and allows to annotate the proportion of summary facts that can be found in the source document.

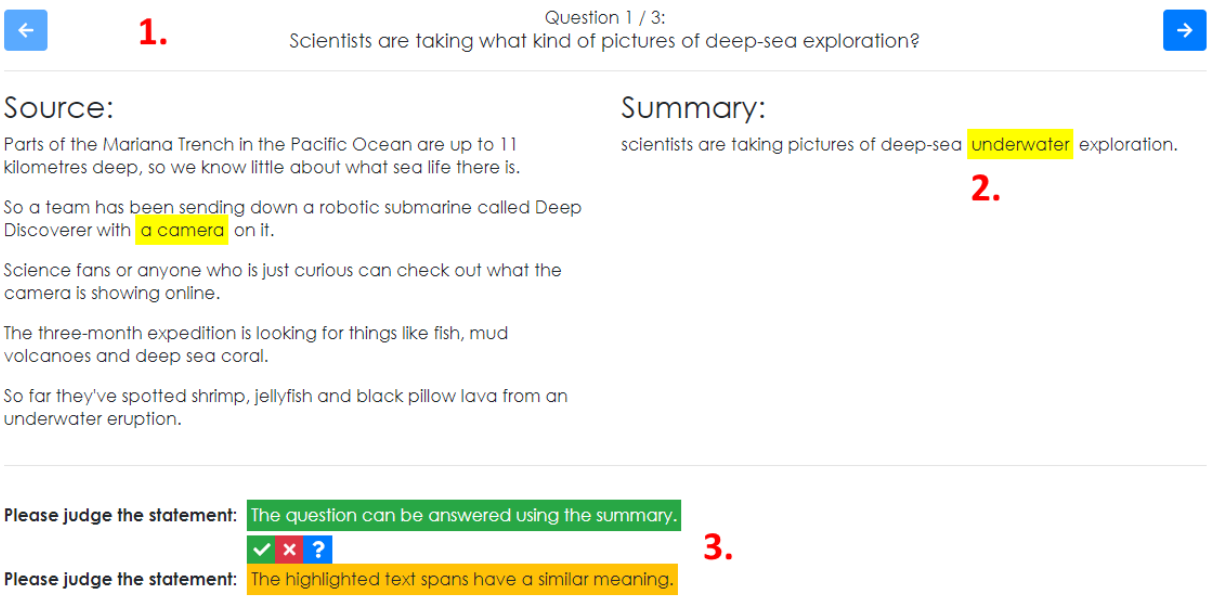

Figure 5.7: The QGQA annotation component of FaithAnno. It visualizes the generated question as well as the extracted answers to this question. The component also enables to judge statements e.g. about the question and answers.

QGQA: The QGQA framework first generates questions based on the summary. Next, the generated questions are answered using either the source document or the summary and finally the answers are compared. FaithAnno visualizes the generated questions as well as the extracted answers. This is demonstrated in Figure 5.7. The current question is rendered at the top (1). The arrow buttons can be used to navigate between the different questions. Extracted answers are highlighted with a yellow background color (2). This feature enables to comprehend every step of the QGQA framework. Hovering over a statement (3) at the bottom enables to annotate it using three options: "Correct", "Incorrect" and "Don't know"; similar to the BERTScore annotation component.

5.2 Potential of faithfulness metrics

In this section, we conduct an annotation experiment using FaithAnno in order to asses how much potential the three different faithfulness approaches – BERTScore, Entailment, QGQA framework – have. We exchange, so to speak, a critical component of a metric with humans. The annotators simulate a component that works almost perfectly or, in other words, they simulate a component that has human-like performance. In the case of BERTScore and the Question Generation & Question Answering (QGQA) framework, the annotators mimic the similarity metric by comparing either words or answers. In the case of Entailment, the annotators mimic both fact extraction as well as fact comparison. We basically try to answer the following question: "How much does a metric improve if one component of it has human-like performance?". So, we want to investigate how much potential an approach has, or whether an approach is well suited for assessing faithfulness or not. Furthermore, we use the results of this annotation experiment to find cases that are helpful for the error analysis of the next Section 5.3. Especially cases where the annotators strongly disagree with each other are interesting for the error analysis.

Annotation tasks

We analyze three approaches to assess faithfulness in this experiment.

1. BERTScore matches every summary word with the most similar source word and the faithfulness is based on word similarities.
2. The QGQA framework asks questions addressing the information of the summary and compares the answers. As a result, the faithfulness is based on answer similarities.
3. Entailment finds three important source document sentences and calculates whether the facts of the selected source document sentences imply the facts expressed in the summary.

Consequently, we have three different annotation tasks, which are explained shortly in the following. The annotation tasks are performed using FaithAnno. The user interfaces corresponding to every task are explained in the previous Section 5.1.

BERTScore annotation task: Important words of the summary (no stop or filler words) as well as the most similar word in the source document are highlighted and presented to the annotators. Annotators are asked to judge whether the summary word has a similar meaning as the corresponding source word. The annotators can choose between three options: "Correct", "Incorrect", "Don't know".

QGQA annotation task: A question as well as the corresponding summary and source answer are presented to the annotators. A document-summary pair can have up to six questions to assess the faithfulness. The annotators are asked to judge whether the answer have a similar meaning. Again, they can choose between three options: "Correct", "Incorrect", "Don't know".

Entailment annotation task: The three most relevant source document sentences are as well as the summary are presented to the annotators. The annotators are asked to assess how many summary facts can be retrieved from the source sentences. The annotators can choose between five options: "None", "Few", "Some", "Most of them", "All".

Conduction of experiment

We apply the three faithfulness metrics on the XSUM hallucination dataset by Maynez et al. (2020) to obtain faithfulness scores and select 100 controversial samples for the annotation tasks. Controversial, in this case, means that the human faithfulness judgement as well as the judgements of BERTScore, Entailment and QGQA are fairly different. We also make sure that no source document has more than 15 sentences to make the annotation task a little bit easier.

Then, we give five annotators access to FaithAnno and let them annotate as many samples as they want to. We made sure that no annotator saw the same sample twice. In order to assure good annotation quality, we wrote annotation guidelines, prepared a video that explains how to use FaithAnno and included a short guide explaining the annotation task directly in FaithAnno as shown in Section 5.1. Furthermore, we gave the annotators the opportunity to contact us at any time to ask questions and to reduce misunderstandings.

We collect 900 annotations in total: we select 100 document-summary pairs, every sample includes 3 annotation tasks and we need 3 annotators per task. We use Fleiss' Kappa to evaluate the inter-annotator agreement. Fleiss' Kappa calculates the agreement for any number of annotators giving categorical ratings to a fixed number of items. It expresses the extent to which

Task	Fleiss' Kappa
BERTScore	0.754
QGQA	0.776
Entailment	0.207

Table 5.1: Inter-annotator agreement for all three annotation tasks.

the observed amount of agreement among annotators exceeds what would be expected if all annotators made their annotations completely randomly[3]. Fleiss' kappa also allows that different items may be rated by different individuals which makes this metric a perfect fit for our tasks.

Table 5.1 reports the inter-annotator agreement for every task. Both, the BERTScore and the QGQA annotation tasks have high Kappa values above 0.75, which can be interpreted as substantial inter-annotator agreement. However, the Entailment task has only slight inter-annotator agreement. One reason for this low inter-annotator agreement is that Fleiss' Kappa will be generally higher when there are fewer categories. While BERTScore and QGQA have two categories[4], annotators can choose between five different categories in the Entailment task. Another reason is that the Entailment annotation task is substantially more difficult than the other two tasks. Annotators only have to compare two short texts for the BERTScore and QGQA task: words in the case of BERTScore and answer spans in the case of QGQA. However, annotators have to read and understand most of the source document, read the summary, extract facts from both texts and compare the extracted facts to solve the Entailment task. We ask all annotators which annotation task is the most difficult and takes the most time. All annotators agreed on the Entailment task.

Qualitative error analysis

We manually investigate samples where annotators disagreed to comprehend the inter-annotator agreement and reveal some difficulties.

BERTScore has a high inter-annotator agreement. Only a very small portion of the disagreements was due to mistakes made by the annotators. Such mistakes include classifying two words as not similar, even though they are exactly the same, or classifying two words as similar, even though they express something quite different. We believe these errors are either because of carelessness or due to mis-clicking. Table 5.2 includes some of these mistakes.

Source Word	Summary Word	Similar?
county	town	no, no, yes
city	state	yes, no, no
dead	dead	no, yes, yes

Table 5.2: Examples from the BERTScore annotation tasks where annotators make simple mistakes like mis-clicking or carelessness.

3. adopted from https://en.wikipedia.org/wiki/Fleiss%27_kappa

4. Actually, the annotators can choose between "Correct", "Incorrect" and "Don't know". However, the third option was almost never used so that we converted all "Don't know" into "Incorrect".

Source Word	Summary Word	Similar?
gunman	man	yes, yes, no
Holyhead	Angleseay	no, no, yes
convicted	guilty	yes, no, yes
protestors	riot	no, yes, no
goal	1-0	yes, no, no
injury	suffering	no, yes, no
sentenced	jailed	yes, no, yes
airport	Heathrow	no, yes, no
76-year-old	age	no, yes, no
murder	death	yes, no, no

Table 5.3: Difficult examples from the BERTScore annotation task where annotators disagreed with each other.

The way larger part of the disagreements was because of difficult and ambiguous cases where it is not clear whether the two words should be considered as similar or not. In some cases, annotators have different background knowledge and, therefore, assess the similarity in a different way e.g. "airport" vs. "Heathrow" or "centre-back" vs. "defender". In most other cases, the differences are simply because of subjectivity e.g. "gunman" vs. "man" or "Mr." vs "man". Table 5.3 demonstrates some difficult examples.

QGQA also has a high inter-annotator agreement. Since this task is very similar to BERTScore, the problems and reasons for disagreements are very similar. We find that the number of plain mistakes made by the annotators is smaller, however, the number of difficult cases increases. In this task, one common reason for disagreements is that annotators used different contextual information to assess the similarity. For example, "Mr. Fidler" and "a farmer" is not similar when just comparing the strings. However, given the context of the source document, it becomes clear that "Mr. Fidler" is indeed a farmer. There are – similar to the BERTScore task – a lot of cases where the similarity is not obvious: answers like "antifreeze" vs. "chemical spill" or "three man" vs. "three youths" are difficult to judge. Table 5.4 lists more interesting examples.

Source Answer	Summary Answer	Question	Similar?
The victim	a 23-year-old man	Who died after a shooting outside a house in Sheffield?	yes, no, no
three man	three youths	Who attacked three people in a Kent restaurant?	yes, no, no
antifreeze	chemical spill	What caused the death of a cat in a Leicestershire street?	yes, no, yes
Mr. Werner	a motorist	Who was jailed for causing the death of a motorcyclist?	no, yes, yes

Table 5.4: Difficult examples from the QGQA annotation tasks where annotators disagreed with each other.

Entailment has a low inter-annotator agreement and is also considered the most difficult task. We find that the annotators disagree in 83 out of 100 samples. We analyze the degree of disagreement by comparing the difference between the minimum and maximum classification value. Consider the following example: two annotators say 25% of the facts can be found, one annotator says 50%. Then, the difference between minimum and maximum is 25.

17 samples have no difference, 49 samples have a difference of 25, 27 samples have a difference of 50 and 7 samples have a difference of 75 or more. We argue that the samples with a difference of 25 are not problematic. This small difference can be explained with subjectivity of the annotators and difficulty of the task. However, the other samples are problematic and interesting and we investigate them to find common problems. We find that all 7 samples where the difference is higher than 75 are due to annotator mistakes. In all of these examples, two annotators have exactly the same opinion, whereas one annotator makes a completely different annotation: e.g. two annotators annotate 0% and one annotator annotates 75%.

Table 5.5 demonstrates two representative examples where the annotators have a difference of 50. These example demonstrate the difficulty of this task. The first summary mentions a 24-year-old man who died after being shot in a certain street, whereas the source document mentions a 25-year-old man that was seriously injured after being shot in a certain district. The annotations range from 25% to 75% meaning that one annotator only retrieved 25% of the facts, while another annotator found 75% of the facts. Both annotations are understandable. The facts expressed in the summary are "24-year-old-man", "died", "being shot", "in Sheffield street". "Being shot" is the only unquestionable common fact of the summary and source document. One annotator might argue that "25-year-old man being shot" and "24-year-old man being shot" is a similar fact and, in addition, might know that "Sheffield street" is in the "Shiregreen area" and, therefore, deems these facts as similar resulting in an annotation of 75%, whereas another annotator might disagree with this which results in a score of 25. The second example of Table 5.5 can be explained in a similar way.

Source	Summary	Annotation
Officers found the man with a gunshot wound to his chest. Local residents said the man lives close to the scene of the shooting. On Monday, a 25-year-old man was seriously injured when he was shot in the street in the Shiregreen area.	A 24-year-old man has died after being shot in a Sheffield street.	50%, 25%, 75%
Police said there had been a fight with a group of youths who tried to gain entry, Che was stabbed in a later scuffle in an alleyway. A 16-year-old boy was also stabbed and taken to a north London hospital. Two boys aged 15 and 16 were also arrested on suspicion of murder and bailed until December.	A 17-year-old boy has been stabbed to death during a fight at a party in north-west London.	25%, 75%, 50%

Table 5.5: Difficult examples from the Entailment annotation tasks where annotators disagreed with each other.

Results

We finally utilize all annotations of a faithfulness metric to mimic a critical component of it that has human-like performance. The annotators mimic the similarity metric for the BERTScore and Question Generation & Question Answering method by comparing either words or answers, whereas the annotators imitate a fact extraction and fact comparison component for the Entailment approach. In this section, we answer the question "How much does a metric improve if one component of it has human-like performance?".

Since we have three annotators per sample, we have three different annotations per word (BERTScore), answer (QGQA) or summary (Entailment). We aggregate the annotations using majority vote (maj) and average vote (avg). Next, we calculate a faithfulness score using the aggregated annotations as described in the following.

BERTScore: Every important summary word is assigned either 1 if it is similar to the corresponding source word, or 0 otherwise. We average the similarity score over all important summary words to obtain the faithfulness score. For example, if the summary contains 3 important words which are labeled with $1, 1, 0$, the resulting faithfulness score is $2/3 = 0.66$.

QGQA: Every summary has up to six different questions and, consequently, up to six different answers. Every answer is assigned either 1 if the answer is similar to the source answer, or 0 otherwise. Again, we average the similarity score over all answers to obtain the faithfulness score. For example, if the summary contains four answers which are labeled with $0, 0, 1, 1$, the resulting faithfulness score is $2/4 = 0.5$.

Entailment: We interpret the aggregated annotations directly as faithfulness score. For example, if the annotators could retrieve 75% of the summary facts in the source document, the resulting faithfulness score is 0.75.

We compare the performance of human-enhanced faithfulness metrics[5] to the performance of the original faithfulness metrics in order to assess the potential of the metric. We argue that an approach has high potential, if the relative improvement due to the component with human-like performance is high. Therefore, we calculate the correlation of the faithfulness metrics with the human faithfulness judgements of the XSUM hallucination dataset.

Table 5.6 shows the results. Please note we find that BERTScore is the best-performing faithfulness metric on this dataset in Chapter 4, but it performs rather badly on these 100 selected examples. Both approaches, QGQA and Entailment, have a very high potential improving the correlation with human faithfulness judgements by more than 75%. In contrast to that, BERTScore only improves by 50%.

We conclude that two approaches are very promising to assess the faithfulness of summaries. The first approach is the QGQA framework which, in short, asks questions about the summary and compares the answers to assess the faithfulness. The other approach is Entailment, which extracts the facts from both summary and source document and judges whether the source facts imply the summary facts. We show that having a component with human-like performance leads to very good results for both approaches. Future research could try to improve these approaches in order to develop better faithfulness metrics.

5. We call them human-enhanced metrics as the final component that calculates the faithfulness score is replaced by humans, but the previous steps or components are the same as in the original faithfulness metrics.

Method	Pearson (r)	Spearman (p)	Relative improvement
BERTScore_original	0.236	0.273	-
BERTScore_enhanced_maj	0.353	0.385	49.9%
BERTScore_enhanced_avg	**0.356**	**0.397**	51.2%
QGQA_original	0.256	0.224	-
QGQA_enhanced_maj	0.444	0.461	73.4%
QGQA_enhanced_avg	**0.469**	**0.481**	**83.1%**
Entailment_original	0.360	0.294	-
Entailment_enhanced_maj	0.615	0.630	70.1%
Entailment_enhanced_avg	**0.632**	**0.637**	**75.6%**

Table 5.6: Correlation of the human enhanced faithfulness metrics with human judgements compared to the correlation of the original faithfulness metric with human judgements. Please note that the correlation is computed using only the selected 100 examples of the XSUM hallucination dataset. The human enhanced variant is always better than the original method.

5.3 Error analysis

In this section, we find and analyze common errors of three faithfulness metrics: BERTScore, Textual Entailment and the Question Generation & Question Answeringf (QGQA) framework. We apply the faithfulness metrics on the XSUM hallucination dataset by Maynez et al. (2020) and manually look through 100 random samples (one sample consists of a source-summary pair, a human faithfulness judgement and three faithfulness scores predicted by the three metrics). We also investigate 50 selected samples where the gap between human and predicted faithfulness score is especially large to understand the worst-cases even better.

BERTScore

We compare the faithfulness scores predicted by the BERTScore faithfulness metric with the human faithfulness judgements to get a first impression where potential errors happen. Figure 5.8 shows two plots. The first one visualizes the human and BERTScore predictions as point cloud, whereas the second one visualizes the difference between the human and BERTScore faithfulness scores computed on the whole XSUM hallucination dataset. The first plot reveals that BERTScore always assigns high faithfulness scores ranging from 0.66 to 0.99 where most of the data points are between 0.8 and 0.9. The second plot shows that BERTScore tends to overpredict rather than predicting lower faithfulness scores: 75% of the documents were rated with a higher faithfulness score than that of the humans.

Based on this observation, we look through the 100 random and 50 selected samples to find reasons why BERTScore assigns such high faithfulness scores even to very unfaithful summaries. Unfortunately, we do not have the time and capacities to quantify the results precisely. Instead, we report the findings in a descending order of importance / contribution to high faithfulness scores.

A major problem of BERTScore is that it is neither able to analyze the structure of the text nor able to comprehend the relations between entities. Even though this metric utilizes sophisticated

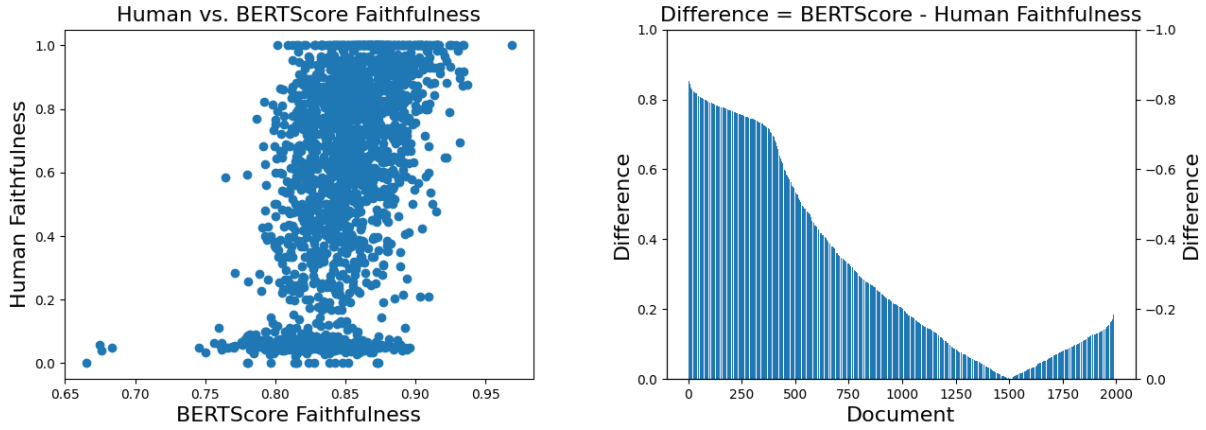

Figure 5.8: Two plots comparing the predictions of the BERTScore faithfulness metric with human judgements. The first (left) plot visualizes the predictions as point cloud. The second (right) plot visualizes the differences between BERTScore and human faithfulness.

contextualized embeddings, we find that most of the time this representation is not able to encode structure and relations well. In many cases, summaries consist of phrases (e.g. noun phrases or various types of entities) that appear identically in the source document but the phrases are used very differently, i.e. relations are very different or phrases are used in an entirely different context. In other words, many summaries are unfaithful even though their phrases match well with the source documents. BERTScore predicts very high faithfulness scores in these cases because this metric performs the comparison on token-level.

Another common issue for BERTScore is dealing with unfaithful compound nouns. Many summaries contain arbitrarily assembled compound nouns that sometimes seem to be generated by randomly selecting nouns of the source document and concatenating them to a phrase. Consider the following phrases to get an idea: “Macedonia’s Prime Minister Justin Riot” or “Sudan President Barack bin Laden”. Even though these phrases consist of completely incoherent words, BERTScore assigns high faithfulness scores since the metric compares token-by-token instead of operating on a phrase level and, therefore, is often able to retrieve every constituent in the source document.

Comparing numbers is also very problematic for BERTScore. Summaries that contain incorrect numeric values for e.g. amounts, counts, money, age, prices, dates, results of sport matches etc. are consistently rated as faithful by BERTScore. It does not matter to BERTScore whether someone achieved the second or first place in a race, 5 or 5000 people were killed, something costs 2000 or 2 million dollars. Table 5.7 shows examples that further underline this point. This type of error is mainly due to the tokenizer. Numbers like 5.000 are split into two tokens "5." and "000". This results in rather high similarities when comparing fairly different numbers like "5.000 dollar" and "5 dollar". In contrast to this, comparing singular and plural works relatively well e.g. "Two men" and "One man" have a similarity of 85%.

Furthermore, negations, opposites and contradictions are very problematic for BERTScore. Antonyms are often used in similar contexts and, as a result, BERTScore assigns very high similarity scores. For example "Smith is dead" and "Smith is alive" have a similarity of 93%. Negations are even more problematic as a single word is able to flip the meaning of a whole

Source	Summary	Faithfulness
2 people	2.000 people	93%
jailed for 4 years	jailed for 7 years	97%
they lost 4-0	they lost 3-2	94%
British number 4	British number 3	96%

Table 5.7: Examples where BERTScore struggles with numeric values. The metric predicts high faithfulness / similarity scores even though the source and summary express fairly different information.

sentence making up only a small percentage of it. For example the sentences “Tim does work in Hamburg” and “Tim does not work in Hamburg” are 97% similar according to BERTScore.

Lastly, BERTScore struggles with long source documents. The longer a source document, the more words it contains and the higher the chance that BERTScore finds a similar source word for a summary word. Despite being used in different contexts, summary words are assigned high faithfulness scores by BERTScore. This occurs especially often when common verbs like be, say, have, use etc. are used in the summary to connect unrelated entities or phrases.

Textual Entailment

To understand where a large portion of errors happen, we compare the faithfulness scores predicted by the Entailment faithfulness metric with human faithfulness judgements. Figure 5.9 shows two plots. The left one visualizes both human and Entailment faithfulness predictions as point cloud. The right one shows the difference between both predictions. Considering both plots reveals that Entailment tends to underpredict the faithfulness scores assigning mostly very small values ranging from 0.0 to 0.1. The right plot shows that about 88% of all documents are rated with a lower faithfulness score than that of the human judges.

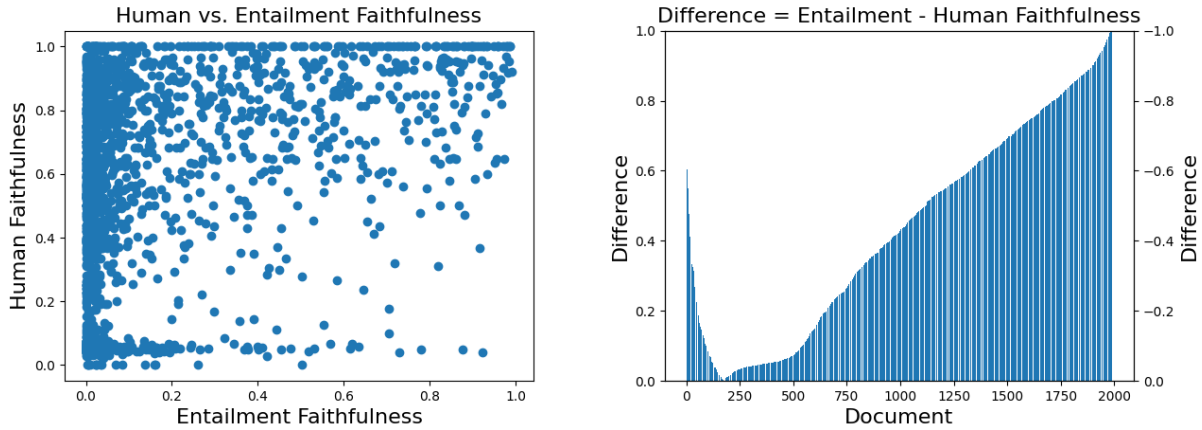

Figure 5.9: Two plots comparing the predictions of the Entailment metric with human judgements. The first (left) plot visualizes the predictions as point cloud. The second (right) plot visualizes the differences between Entailment and human faithfulness.

Based on this finding, we look through the 100 random and 50 selected samples to find reasons why Entailment assigns low faithfulness scores even to very faithful summaries. Again, we do not have the time and capacities to quantify the results precisely, but we report the findings in a descending order of contribution to low faithfulness scores.

We find that the biggest issue of the Entailment faithfulness metric is dealing with unfaithful verbs. In cases where the summary consists of multiple faithful entities or phrases which are connected by an unfaithful verb, the Entailment metric predicts very small entailment probabilities. Once we started investigating this, we found that the verb has the most impact on the predicted score. When interpreting verbs as relations, one could say that the Entailment metric pays most attention to the relations expressed in the texts. As soon as the relation of the first text does not imply the relation of the second text, the entailment probability decreases drastically. Please consider the examples listed in Table 5.8. While this property basically describes the way how entailment models should work, it is very disadvantageous for our dataset: when summaries consist of faithful entities connected with an unfaithful verb, the human annotators mark the verb as unfaithful and everything else as faithful resulting in a high faithfulness score. However, the metric predicts no entailment (= low faithfulness score) in such cases.

Another common problem is our introduced context limitation to $k = 3$ sentences. When we propose to calculate the entailment probability given the top three most similar sentences and the summary sentence, we assume that all information that is present in the summary is spread across at most three sentences of the source document. However, this assumption does not hold for all cases. Especially when the source document is long or when the summary is very abstractive and fuses the information of multiple sentences into one, this assumption is problematic. We observe many cases where additional information, i.e. increasing k, would lead to higher predicted faithfulness scores.

Source	Summary	Faithfulness
I like you.	I love you.	23%
Einstein was born in Germany.	Einstein died in Germany.	1%
A whale has been sighted off the coast of New South Wales.	A whale has gone missing off the coast of Australia.	1%
The minister said she has secured executive agreement to ask the assembly to pass a legislative consent motion to pardon convictions.	The assembly has approved a motion to pardon convictions.	0%
The men threatened a member of staff with a knife.	Two men have been threatened with a knife.	23%
Moscow imposed sanctions on Turkey.	Russia suspended all sanctions against Turkey.	0%

Table 5.8: Examples where Entailment struggles with unfaithful verbs. The metric predicts very low faithfulness scores even though most parts of the summary are considered faithful by human annotators.

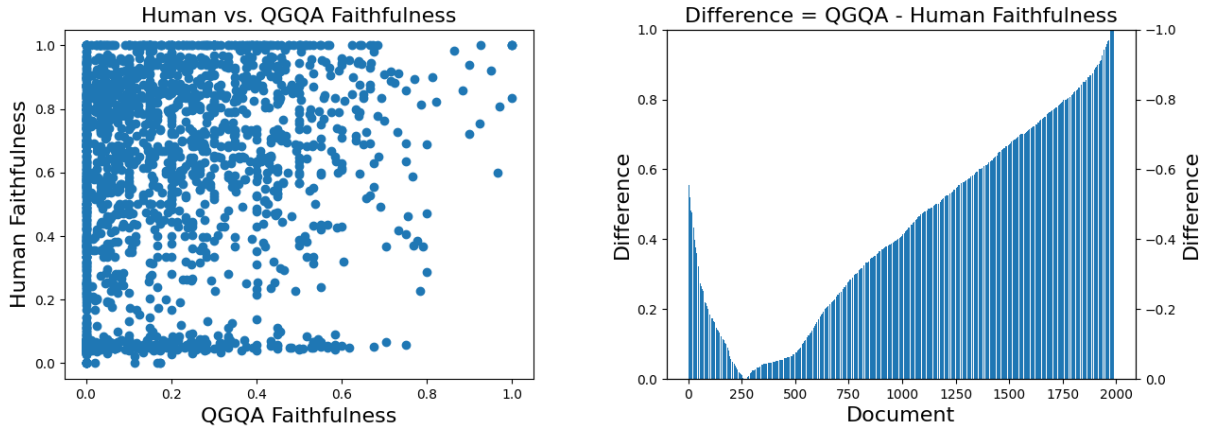

Figure 5.10: Two plots comparing the predictions of the QGQA faithfulness metric with human judgements. The first (left) plot visualizes the predictions as point cloud. The second (right) plot visualizes the differences between the QGQA and human faithfulness.

Similar to the last problem, we observe that the source sentence selection mechanism (calculating the sentence similarity between summary sentence and all source sentences, then selecting the top k most similar sentences) sometimes does not select the best possible sentences. Often, the selected sentences do not contain similar information but similar stop and filler words. In return, sentences that contain important information are not included. As a consequence, the summary faithfulness cannot be correctly determined. Here, focusing the sentence selection mechanism on important content rather than using plain similarity would definitely help. One simple solution could be to apply stopword filtering before computing the similarity.

Finally, we observe some cases where the summary suffers from grammatical errors or word repetitions. In these cases, the entailment model is not able to deal with the summary sentence and predicts no entailment. This is basically a robustness problem of the entailment model, but is not surprising since the training data of entailment models, e.g. the Multi-NLI dataset, does not cover such cases. Furthermore, we argue that summarization systems should first solve such simple problems like repetition and grammar before caring about faithfulness and, hence, we do not think that a faithfulness metric necessarily has to be able to deal with such errors.

Question Generation & Question Answering

We compare the faithfulness scores predicted by the QGQA framework with human faithfulness judgements to comprehend where most errors occur. Figure 5.10 shows two plots. The left one visualizes human and QGQA faithfulness predictions as point cloud while the right one visualizes the difference between both predictions. The first plot reveals that QGQA assigns faithfulness scores in the full range from 0.0 to 1.0. Most of the data points lie between 0.0 and 0.6. This plot suggest that QGQA tends to underpredict the faithfulness scores similar to the Entailment metric. However, the QGQA framework has the additional problem that almost no examples get assigned high faithfulness score: only about 10 examples have faithfulness scores in the range from 0.9 to 1.0. The right plot confirms that QGQA underpredicts faithfulness scores showing that about 85% of all documents are rated with lower faithfulness scores than that of the human judges.

Based on these finding, we look through the 100 random and 50 selected samples to find reasons why the QGQA framework assigns low faithfulness scores even to very faithful summaries and why the QGQA framework has problems with predicting high faithfulness scores. Again, we do not have the time and capacities to quantify the results precisely, but we report the findings in a descending order of importance.

Compared to BERTScore and Entailment, the QGQA framework consists of more components and generally correlates worse with human faithfulness judgements as shown in Chapter 4. Having three different components, the QGQA framework is way more prone to error propagation than BERTScore, which consists just of the embedding model or Entailment, which utilizes only an entailment model and a sentence selection mechanism.

The major problem of the QGQA framework – at least with our implementation – is the answer similarity metric. The answer similarity metric is responsible for comparing the found answers and calculating the faithfulness score. We decide to use the simple but effective F1 metric. However, performing exact match on a token-level comes with many disadvantages and we find several instances where a better answer similarity metric is needed. The answer similarity metric F1 has problems with the following:

1. Paraphrases with the same meaning e.g. matching "contraband goods" with "drugs"
2. Abbreviations e.g. matching "GB" with "Great Britain"
3. Singular and plural e.g. matching "men" with "man"
4. Generalizations e.g. matching "save 5€" with "save money"
5. Locations e.g. matching "London" with "England"
6. No background knowledge e.g. matching "John Deere" with "tractor" or matching "pharmaceutical firm" with "Accord Healthcare"

To our surprise, exchanging the answer similarity metric with a model-based one that is able to deal with all of the problems mentioned above (e.g BERTScore) does not help to increase the correlation with human judgements. In Section 4.6, we experiment with BERTScore as an answer similarity metric but find only very little improvements. Using BERTScore instead of F1 increases the correlation with human judgements from 0.23 to 0.25.

The question generation component of the QGQA framework also has issues. One major problem occurs when the summary is rather short. In these cases, the question generation model often generates "questions" that are equal to the summary but the period is replaced with a question mark. This leads to unpredictable behaviour of the question answering model selecting random phrases as answers and ultimately decreases the faithfulness of the summary. In addition to that, we find many cases where the generated question is nonsensical and, thus, a good answer cannot be found and, consequently, the faithfulness of the summary decreases. The generation of unanswerable question happens especially often when the summary contains word repetitions. Table 5.9 lists summaries with word repetitions and automatically generated, unanswerable questions. Another issue that occurred rather frequently is that the generated questions target irrelevant information. Answers to such question almost always do not help to assess the faithfulness of the summary. This is directly connected to the answer candidate selection component of the QGQA framework that selects possible answer spans which are used by the question generation component to generate questions. Focusing the answer selection component on critical parts of the summary could definitely help here.

Summary	Question
A man has been found guilty of murdering his mother and his mother.	A man was found guilty of murdering his mother and who else?
The examination has concluded that a man died from measles and measles.	The examination has concluded that a man died from measles and what else?
A man and a woman have been arrested.	A man and what else have been arrested?
A motorcyclist has been arrested on suspicion of causing a crash between a lorry and a lorry.	A motorcyclist was arrested on suspicion of causing a crash between a lorry and what?

Table 5.9: Summaries with word repetition and questions generated by the QGQA framework that are not answerable using the source document. The generation of such unanswerable questions cause an artificial decrease of summaries' faithfulness.

The question answering component of the QGQA framework suffers from a fairly obvious problem: it is often not able to identify the correct answer in the source document. As a consequence, the faithfulness of the summary decreases, even though the summary is faithful. A more interesting problem can arise when the summary is very unfaithful. The QG model generates a question that starts with a question word like "When" or "Who" and continues with rather nonsensical phrases from the summary. Since the question word aims at a specific entity – e.g. date or person – the question answering model just selects the first fitting text span as an answer and, more often than not, the answer is the same for both summary and source document resulting in an increase of the faithfulness score. While this error sounds rather specific, it occurs surprisingly often in our sample set. Consider the following question: "When will Justin Riot step down as Prime Minister of Macedonia?". This question aims at a date, and even though "Justin Riot" is a non-existing person not mentioned in the source text, the source document is neither about "stepping down" nor about "Prime Ministers" the question answering model still finds the same answer in both summary and source document. Other exemplary questions like this are listed in Table 5.10. A possible solution to errors like this is using a question answering model that must not to find an answer but can predict "Don't know" or "unanswerable". Such models could detect that the question is mostly nonsense and consequently predict "unanswerable". Unanswerable questions could either be ignored completely or counted towards the unfaithfulness of the summary.

To conclude, basically every component of the QGQA framework has it's problems and needs to be improved in order to keep up with other faithfulness metrics like BERTScore or Entailment. However, this approach has a lot of potential as we demonstrate in Section 5.2.

Question	Source does not mention...	Target
Three potholes have been found in what county in Northumberland?	... any finding of potholes	County
A man has been found guilty of assaulting who after the death of a man in a row over a holiday park?	... that a man was found guilty	Person
How many youths were attacked in a Kent restaurant?	... that youths were attacked	Number
On what road did a lorry driver die?	... that a lorry driver died	Road
According to whom, a dog found on a beach in Dorset may have been poisoned?	... that a dog has been poisoned	Person

Table 5.10: Example questions where the question answering model finds the same answer in both summary and source even though the source document does not mention any of the information given in the question. The question answering model ignores the question's context and just focuses on the question word or question target.

6 Towards faithful summarization systems

In this chapter, we provide answers to the third research question: “How can faithfulness metrics be used to develop faithful summarization systems?”. The first section describes ideas how we could support human annotators that are widely used to perform manual faithfulness evaluations of newly proposed summarization systems. This approach only indirectly affects the development of faithful summarization systems. In the next section, we describe a pre-processing approach where we filter training data in order to improve the faithfulness of trained summarization systems. In the final section, we explain a re-ranking approach to increase the faithfulness of summarization systems with a post-processing step.

6.1 Predicting unfaithful text spans

The goal of this experiment is to test whether BERTScore could be used to assist human annotators that have the task to assess the faithfulness of summaries. In many recent papers about summarization (Gabriel et al., 2021; Cao et al., 2018; Huang et al., 2020; Y Zhang et al., 2020), humans are given the task to evaluate the factual correctness of newly proposed summarization system. This is of course a very challenging and time-consuming tasks as factual errors are often not easy to detect. Here, we check whether BERTScore can successfully highlight unfaithful text spans and, as a result, ease the annotation process. In other words, we evaluate the "helpfulness" of BERTScore to human annotators.

6.1.1 Dataset

For this experiment, we use the XSUM hallucination dataset again. It consists of about 2000 source-summary pairs that were annotated by three trained annotators. In contrast to the experiments of Chapter 4 where we used the human faithfulness judgements as ground-truths, we make use of the annotated spans in this experiment.

Consider the following exemplary summary: “Albert Einstein was born in England”. Imagine two annotators marked "England" as unfaithful and one annotator marked "in England" as unfaithful. We convert the summaries of the dataset into ground-truth prediction sequences as follows: for every character, we take the majority vote of whether it belongs to a faithful or an unfaithful text span. This results in a binary classification sequence. For this particular example, all characters of the sub-sequence "Albert Einstein was born in" are converted to False and all characters of the sub-sequence "England" are converted to True. After converting, the dataset is pretty balanced consisting of 98200 positive and 103003 negative characters.

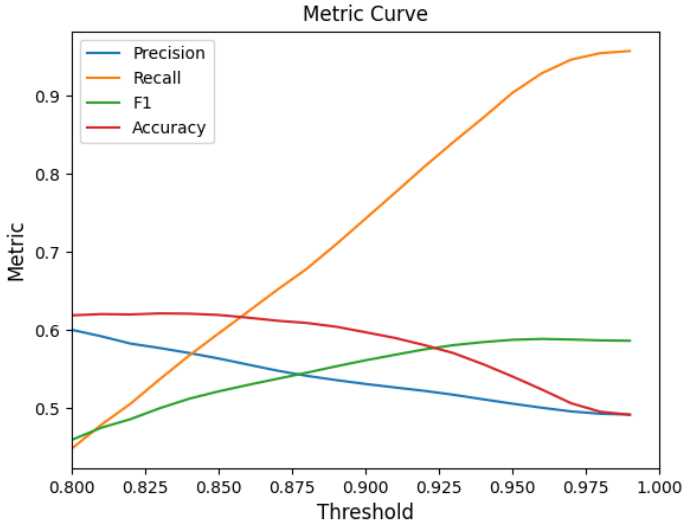

Figure 6.1: Evaluation of all thresholds for BERTScore's helpfulness to human annotators.

6.1.2 Results

We apply the BERTScore faithfulness metric on every source-summary pair of this dataset. As a result, every summary token gets assigned a similarity – or as we call it in this experiment – a faithfulness score. Next, we define a threshold. Every token which faithfulness score is less then the threshold is considered as unfaithful. Then, we convert the judged summary into a binary classification sequence as described above. Finally, we calculate precision, recall, F1 and accuracy for this binary classification task.

The results for selected thresholds are shown in Table 6.1. Figure 6.1 depicts precision, recall, F1 and accuracy for all thresholds in the range of 0.8 - 0.99. In Section 5.3, we found that BERTScore assigns faithfulness values between 0.8 and 0.9 to most documents. We choose the threshold values for this experiment accordingly. The subjectively best threshold for this dataset is 0.9, achieving an accuracy of 0.6 while still maintaining a good precision and recall.

Unfortunately, BERTScore's helpfulness is not that high. The annotators cannot rely on BERTScore's prediction alone. However, it could still be used in an annotation tool to guide the annotators and accelerate the annotation process. We plan to explore BERTScore's helpfulness to annotators in detail in future work. Developing a prototype application that features BERTScore as guide and evaluating the helpfulness manually is not in the scope of this work. Figure 6.2 visualizes a mock-up of such an annotation tool.

Threshold	Precision	Recall	F1	Accuracy
0.8	0.60	0.45	0.46	0.62
0.9	**0.53**	**0.74**	**0.56**	**0.60**
0.99	0.49	0.96	0.59	0.49

Table 6.1: Evaluation of BERTScore's helpfulness to human annotators. The table lists three representative faithfulness thresholds.

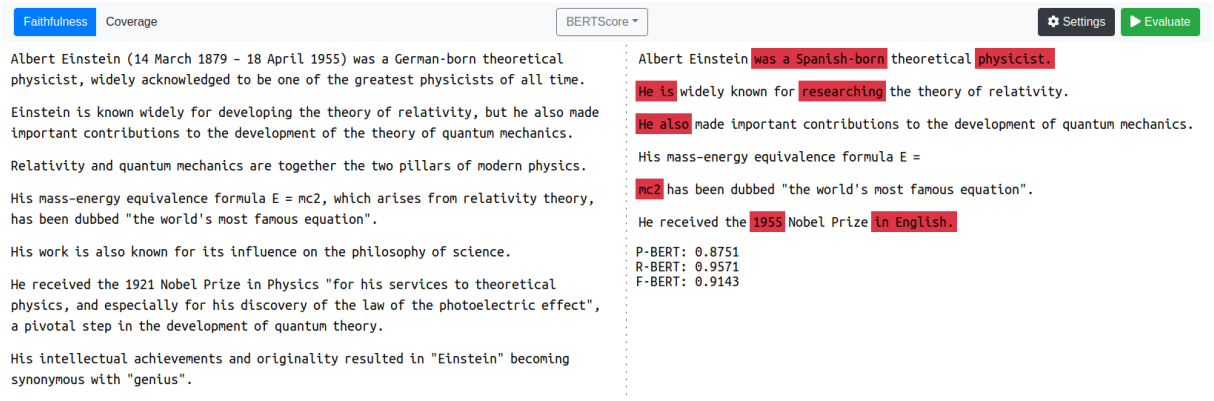

Figure 6.2: A mock-up of an annotation tool that uses BERTScore to highlight unfaithful text spans in the summary. The source document is displayed on the left side and the summary document is visualized on the right side.

6.2 Training data filtering

This experiment is based on the observation that many ground truth summaries of the XSUM dataset are unfaithful. We use BERTScore to evaluate the faithfulness of all ground truth summaries. Furthermore, we apply named entity recognition on both the ground truth summaries as well as the source documents. We find that the average BERTScore faithfulness is 0.85 and that about 50% of the named entities[1] that appear in the summary cannot be found in the source document.

We argue that the unfaithfulness of summarization models trained on the XSUM dataset is – among other factors – a consequence of the unfaithful training data. Therefore, we propose to filter the training data in a way that it consists of mostly faithful ground truth summaries.

We construct two new training data sets. The first one consists only of source-summary pairs where the faithfulness score calculated by BERTScore is greater than the threshold 0.85. The second training data set consist only of examples where every named entity mentioned in the summary can be found in the source document. Table 6.2 shows the number of examples before and after we applied the filtering.

Dataset	Number of training examples
XSUM	204,045
XSUM-BERTScore	146,745
XSUM-NER	101,844

Table 6.2: The number of training examples of the XSUM dataset and two filtered variants of it. The BERTScore variant is filtered using a threshold, while the NER variant is filtered by comparing the entities of the summary with the source document.

1. We only consider the tags PERSON (People), LOC (locations, mountains, water), ORG (institutions, companies, agencies), FAC (buildings, roads, bridges, airports), GPE (cities, states, countries), NORP (political groups, nationalities), EVENT (named sport events, wars, natural disaster). Other types of entities (e.g. dates, times, numbers) are ignored as it is difficult to align them with the source document since these entities can be represented in many different ways.

Model	BERTScore	ROUGE-1	ROUGE-2	ROUGE-L
t5-small-xsum	91.70	**34.23**	**12.05**	**26.83**
t5-small-xsum-bert	92.49	33.49	11.43	26.14
t5-small-xsum-ner	**92.54**	32.11	10.49	25.00
bart-base-xsum	89.67	**41.94**	**18.98**	**34.00**
bart-base-xsum-bert	**90.30**	40.82	17.94	33.02
bart-base-xsum-ner	**90.30**	39.44	16.79	31.84

Table 6.3: Evaluation of the summarization models T5 and BART trained on different variants of the XSUM dataset. XSUM is the original dataset, XSUM-BERT is a filtered variant where every ground-truth summary has a BERTScore larger than 0.85 and XSUM-NER is a filtered variant where every entity of the ground-truth summary can be found in the source document.

Next, we fine-tune two current transformer-based summarization models on the filtered datasets as well as on the original dataset. We use the huggingface transformers library (Wolf et al., 2020) to train T5-small and BART-base on the datasets. BART is a state-of-the-art summarization model that was pre-trained with a masking technique that is very similar to the summarization task, while T5 is pre-trained for multiple tasks including summarization and question answering. We limit the number of input tokens to 512 for both models.

We use the Adam optimizer with default parameters for training: a learning rate of 5e-05, no weight decay, an Adam beta1 of 0.9, beta2 of 0.999 and epsilon is set to 8e-08. We use a linear learning rate scheduler and no warm-up steps. Both models were trained until convergence with a batch size of 16 examples on a single V100 GPU. This took 5 and 3 epochs for the T5-small and BART-base model, respectively. Summaries are generated using beam-search with 4 beams.

We evaluate all six models (two different architectures trained on three datasets) using ROUGE to assess informativeness (ROUGE-1, ROUGE-2), fluency (ROUGE-L) and BERTScore to assess faithfulness since we found in Chapter 4 that BERTScore has the highest correlation with human judgements. The results are listed in Table 6.3.

Training the models on any of the filtered datasets leads to almost exactly the same faithfulness improvements. While the models T5-small and BART-base achieve a BERTScore of 91.70 and 89.67 when trained on XSUM, they achieve a BERTScore of 92.5 and 90.3 when trained on a filtered version of the dataset. Even though these are only small improvements of faithfulness, it is an indicator that this simple technique of filtering the training data is beneficial and that the training data has indeed influence on the faithfulness of the summarization system.

Naturally, the standard versions of the models trained on the whole XSUM dataset achieve the highest ROUGE scores as these models are trained on way more training data. Compared to the models trained on the NER variant of the dataset, the standard models saw about twice as many training examples. Interestingly, training on a reduced training data set only slightly affects the ROUGE evaluation scores loosing about 1 - 2 points.

Another interesting point that the results reveal is that the T5-small model actually achieves a higher faithfulness than the BART-base model. Considering the ROUGE scores, the BART-base model is clearly superior to T5-small. This is expected since BART's pre-training closely

resembles the summarization task and, furthermore, BART is already known to be a state-of-the-art summarization model for many datasets. There are definitely many possible explanations to this, but we noticed that summaries generated by BART are more abstractive when compared to summaries generated by T5 and more abstractive summaries – in contrast to more extractive summaries – leave more room to make hallucination errors. Finally, we want to note that our evaluation results indicate an inverse correlation of faithfulness and ROUGE, highlighting the need to evaluate summarization systems not only using ROUGE but also for examples using BERTScore as we do in this experiment.

6.3 Re-ranking

In this small series of experiments, we want to explore whether faithfulness metrics can be utilized to re-rank sentences or summaries in order to ultimately increase the faithfulness of the system. In other words, we apply faithfulness metrics in a post-processing step: after generating multiple texts, we utilize a faithfulness metric to select the best one.

6.3.1 Re-ranking sentences

For the first experiment, we use the dataset collected by Falke et al. (2019). This sentence ranking dataset contains 373 triples, each triple consists of a source sentence and two summary sentences. The source sentences are from the CNN/DailyMail dataset and the two summary sentences are generated by the summarization model from Chen and Bansal (2018). One summary sentence is faithful to the source sentences, whereas the other summary sentences is factually inconsistent with the source sentence. The goal of this experiment is to consistently rank the faithful sentence higher than the unfaithful sentence.

The CNN/DailyMail dataset was introduced by Hermann et al. (2015) and contains about 311.000 samples. The source documents are news articles and the summaries are acquired by concatenating extracted highlights written by an article's author. This dataset is widely considered as a highly extractive one consisting of rather disconnected sentences which, as a result, leads to extractive rather than abstractive summarization models. Consequently, summaries from models trained on this dataset do not suffer as much from hallucinations and factual inconsistencies as for example models trained on the XSUM dataset. Therefore, we consider the sentence ranking task of this experiment as rather easy and expect good results.

We compare the faithfulness metrics described in Chapter 3 against each other and we include some results of other researchers in our comparison as well. We test how often the metrics prefer the correct sentence i.e. giving the faithful sentence a higher score than the unfaithful sentence.

In mathematical terms, let N be any faithfulness metric and let d be the source sentence, x be the faithful summary sentence and y be the unfaithful summary sentence. Then, $N(d,z)$ is the faithfulness score of any sentence z given source sentence d calculated by N. We test

$$N(d,x) > N(d,y) \tag{6.1}$$

for each triple and calculate the accuracy over all triples.

Method	Correct predictions (in %)	Delta
Random	50.00	0
Named entity recognition	29.50	-20.5
Open information extraction	49.06	-0.94
ESIM (from Falke et al. (2019))	67.60	+17.6
Semantic role labeling	69.44	+19.44
Sentence similarity	69.71	+19.71
FactCC	70.00	+20
Question generation & answering	71.85	+21.85
BERTScore	77.48	+27.48
Entailment	**88.47**	**+38.47**
Human (from Falke et al. (2019))	83.9	+33.9

Table 6.4: Results on the sentence ranking experiment from Falke et al. (2019). Human performance was crowd-sourced. Ties are counted as incorrect predictions.

Results are shown in Table 6.4. Please note that this table just reports the best scores for every faithfulness method to give an overview. We conduct multiple experiments and test various different models to achieve these results. Please refer to Appendix A.2 where the experiments as well as the configurations that lead to the highest scores are described in more detail.

Entailment is the best faithfulness metric to distinguish between unfaithful and faithful sentences achieving 88.5% correct predictions outperforming even human performance. We use the RoBERTa-large model trained on the Multi-NLI (MNLI) dataset to achieve these scores. All other faithfulness metrics perform very similarly on this task ranking about 70% of the example sentences correctly. The only exception to this are the faithfulness metrics based on open information extraction and named entity recognition. Both metrics perform worse than randomly selecting the faithful sentence.

While we were expecting the bad performance of open information extraction, we were surprised to see that named entity recognition performs so badly. We already saw in Chapter 4 that open information extraction does not correlate well with human faithfulness judgements. However, named entity recognition actually performed rather well on this task. In order to understand and analyze the performance on this task, we look through some examples by hand. We find that, in almost every example, the entities mentioned in the summary sentences are also present in the source sentence. This explains the poor ranking performance of the NER faithfulness metric since this method judges the faithfulness based on the found entities. We count ties towards the incorrect predictions and if the entities of both summary sentences also appear in the source sentence, the metric assigns the same faithfulness scores to both sentences.

6.3.2 Re-ranking summaries

Equipped with the findings of the previous sentence re-ranking experiment, we want to exploit the re-ranking capabilities of the faithfulness metrics in this experiment in order to re-rank summaries in a post-processing step and hopefully improve the overall faithfulness of summarization systems.

Ranking method	Model	BERTScore	ROUGE-1	ROUGE-2	ROUGE-L
none	t5-small-xsum	91.70	**34.23**	**12.05**	**26.83**
Entailment	t5-small-xsum	91.68	34.08	11.92	26.59
BERTScore	**t5-small-xsum**	**92.60**	33.70	11.63	26.26
none	bart-base-xsum	89.67	**41.94**	**18.98**	**34.00**
Entailment	bart-base-xsum	89.54	41.48	18.69	33.68
BERTScore	bart-base-xsum	**91.10**	39.39	16.76	31.81

Table 6.5: Evaluation of the summary re-ranking approach to increase faithfulness. We compare the performance of T5-small and BART-base trained on the XSUM with and without applying summary re-ranking. Summaries are re-ranked with either the Entailment or BERTScore faithfulness metric.

In the previous experiment, we found that BERTScore as well as Entailment perform very good on the sentence re-ranking task. Here, we use these two metrics to re-rank summaries generated by our previously trained (see Section 6.2) summarization models: T5-small and BART-base.

First, we use the T5-small and BART-base model that we trained on XSUM to generate summaries on the test set of the XSUM dataset. However, instead of just generating one summary, we use beam-search with four beams to generate multiple summaries and select the $N = 10$ most probable summaries as candidates. Next, we apply both BERTScore and Entailment to assess the faithfulness of these candidate summaries and select the one with the highest faithfulness score. In other words, we re-rank the generated summaries based on faithfulness and select the top-ranked one as the final summary. The BART-large-mnli model is used to compute the Entailment faithfulness metric and the RoBERTa-large-mnli model is used to compute the BERTScore faithfulness metric. Finally, we evaluate the re-ranked summaries using ROUGE to assess informativeness (ROUGE-1, ROUGE-2), fluency (ROUGE-L) and BERTScore to assess faithfulness.

Results are shown in Table 6.5. Re-ranking the summaries with the BERTScore faithfulness metric successfully increases the faithfulness of the summarization model while only sacrificing 1 - 2 ROUGE points. Unfortunately, the Entailment faithfulness metric cannot improve the factual correctness. In the previous sentence re-ranking experiment, Entailment was able to outperform humans. However, it is not helpful for this task. One reason is that the sentence re-ranking experiment is constructed using the CNN/DailyMail dataset which is considered to be easier than XSUM. Another reason is that this experiment is about re-ranking summaries, whereas the other experiment was about re-ranking sentences.

In a last experiment towards more faithful summarization systems, we combine both training data filtering and summary re-ranking. Therefore, we use the T5-small and BART-base model that we trained on the filtered XSUM-NER dataset to generate ten summary candidates for every document of the XSUM test set. Then, we rank the candidate summaries and select the best one using BERTScore, as this method performed very well in the previous experiment. Finally, we compare these summarization models that use both pre- and post-processing steps with the standard summarization models that are only trained on the original XSUM dataset.

Results are shown in Table 6.6. Combining both training data filtering as well as summary re-ranking can successfully improve the faithfulness of the summary increasing the faithfulness from 91.7 to 93.51 and from 89.67 to 91.1 for T5 and BART, respectively. However, the

Model	Filtered?	Re-ranking?	BERTScore	ROUGE-1	ROUGE-2	ROUGE-L
T5-small	False	False	91.70	**34.23**	**12.05**	**26.83**
T5-small	True	False	92.54	32.11	10.49	25.00
T5-small	False	True	92.60	33.70	11.63	26.26
T5-small	True	True	**93.51**	31.35	10.04	24.36
BART-base	False	False	89.67	**41.94**	**18.98**	**34.00**
BART-base	True	False	90.32	39.44	16.79	31.84
BART-base	False	True	90.51	41.38	18.54	33.56
BART-base	True	True	**91.10**	39.39	16.76	31.81

Table 6.6: Evaluation of the summary re-ranking approach combined with the training data filtering approach to increase faithfulness. We compare the performance of T5-small and BART-base with and without applying these techniques. The training data was filtered using the faithfulness metric based on named entity recognition and the summary candidates were re-ranked with the BERTScore-based faithfulness metric.

ROUGE evaluation scores definitely suffer from this, loosing about 2 - 3 points. We argue that this trade-off is worth it as – even though ROUGE is a widely used metric – ROUGE scores are not as important as faithfulness scores since unfaithful summaries are basically useless in practice. Improving both ROUGE and faithfulness evaluation scores and, therefore, optimizing fluency, informativeness and factual correctness should be the goal of future research on summarization.

In this chapter, we demonstrated that simple pre-and post-processing techniques like training data filtering as well as summary re-ranking with the help of faithfulness metrics are able to slightly increase the faithfulness of summarization systems. One major advantages of approaches like this is that model architectures remain unchanged. However, to achieve more significant improvements, we believe that more advanced modeling techniques are necessary which incorporate the idea of faithfulness directly into the model.

7 Conclusion and future work

In this work, we answered three different research questions: (1) "Which faithfulness metric is the best?", (2) "How do faithfulness metrics perform in practice?" and (3) "How can faithfulness metrics be used to develop faithful summarization systems?".

We re-implemented and optimized three well-known faithfulness metrics. We also proposed several new approaches to assess the faithfulness of summaries. Next, we evaluated all faithfulness metrics by correlating them with human faithfulness judgements in order to find the best metric. In the second part of this work, we conducted two qualitative analyzes using our developed FaithAnno tool to comprehend how the metrics perform in practice and analyze which metric has the most potential. In the course of this, we also revealed common issues and error sources of these faithfulness metrics. Finally, we experimented with two simple techniques – training data filtering and summary re-ranking – to increase the faithfulness of summarization models.

We found that BERTScore is the best faithfulness metric correlating very well with human judgements even though it suffers from several problems. Our error analysis revealed that BERTScore's biggest problem is the inability to analyze structure and relations in texts. For example, if summaries mainly consist of phrases from the source document and if the phrases are linked by unfaithful verbs or wrong relations, they are rated as faithful. This is one of several reasons why this method is more inclined to label summaries as faithful.

The annotation experiment revealed that both approaches, Question Generation & Question Answering and Entailment, have high potential. Improving these methods are a promising direction to develop better faithfulness metrics. The annotation experiment as well as the error analysis also showed that especially a better answer similarity method can have a huge impact on the performance of faithfulness metrics. However, we need a fundamentally different way of embedding text to successfully compare unfaithful text spans. Current contextualized embeddings deem texts as similar that occur in the same context. Consequently, phrases like "24-year-old boy" and "30-year-old boy" or "born in" and "died in" or "5000 people" and "400 people" are very similar in this vector space. This is obviously very problematic to assess faithfulness as phrases like that must be considered dissimilar.

We proposed many alternative methods to automatically evaluate faithfulness. However, our experiments showed that these approaches alone cannot compete with an optimized BERTScore. The error analysis revealed that BERTScore has problems with analyzing the relations between entities and the structure of the text. A sophisticated combination of BERTScore with other methods such as Semantic Role Labeling, Named Entity Recognition or Open Information Extraction could lead to a better metric as these methods bring structure into the evaluation process. A better similarity metric, as described before, that is able to correctly compare unfaithful texts would definitely help these approaches as well.

While conducting the error analysis, we found that our implementation of BERTScore in FaithAnno is very helpful to explain the generated summary. This method aligns every summary word with a source word and, therefore, allows to trace back where the information came from. Even though this method does not directly visualize the summarization model's attention, it

definitely helps to understand how the summary was generated and which source words had the most influence. Furthermore, visualizing BERTScore is model-agnostic and can be used to analyze all kinds of automatic generated summaries.

In our last experiments, we found that the training data has an effect on the faithfulness of summarization models that is not to be neglected. If many gold summaries are unfaithful, the resulting model is also unfaithful. We also saw that simple techniques like data filtering and summary re-ranking have a positive impact on a system's faithfulness.

However, our results have some limitations. We only examined a single data set, namely the XSUM dataset for training summarization models and the XSUM hallucination dataset for evaluating faithfulness metrics. Therefore, we cannot say whether BERTScore works well beyond this one data set to evaluate faithfulness. We also assessed the potential of the different faithfulness metrics using only 100 annotated examples. It is questionable how well these results generalize. In addition, we do not have insights from other datasets and can not confirm that the pre- and post-processing steps improve the faithfulness of summarization models in general.

Consequently, we want to evaluate both techniques – filtering and re-ranking – on other data sets and find out whether they can lead to an improvement in faithfulness again. Furthermore, we aim to create a new annotated faithfulness data set that is not based on XSUM to check whether BERTScore correlates well with human faithfulness judgements on other data as well.

We obviously have not solved the faithfulness problem of summarization systems in this work. While we found that techniques like training data filtering and re-ranking of summaries can help with faithfulness, these are just small steps towards faithful summarization systems. In order to achieve more significant faithfulness improvements, more advanced techniques are required. One interesting approach is to incorporate faithfulness into the training objective of a summarization model e.g. not only maximizing the probability of the reference summary but also maximizing the faithfulness. Another approach is to first extract the facts of the source document and then generate the summary based on the extracted facts. This, however, requires advanced and good performing fact extraction pipelines. We confirm the findings of other researchers that current Open Information Extraction models are not sufficient for this task.

We conclude this work giving recommendations and reconsidering faithfulness in general. New summarization models should not only be evaluated with ROUGE for informativeness or fluency, but also for faithfulness. In this work we have shown that BERTScore is well suited for this task. BERTScore is available in the widely used huggingface transformers library and can easily be adopted to evaluate faithfulness. In addition, we recommend analyzing the gold summaries of the training data before training a summarization model, e.g. by using BERTScore, to get an impression of how faithful or unfaithful these gold summaries are. It may be worth filtering the training data in order to train a better, more faithful model.

In this work, we considered faithfulness as a continuous value ranging from 0, indicating absolutely unfaithful texts having basically nothing in common with the source document, to 1, depicting perfect agreement with the source document. However, what does it mean if researchers evaluate a summarization system and achieve a high faithfulness score of 90%? Are 90% of the summaries generated by this model completely faithful? Or does it mean all summaries have a faithfulness of 90%? Are all generated summaries actually unfaithful and is the system still useless in practice?

We should consider faithfulness as a binary property when it is used for model evaluation.

Bibliography

Amina Adadi and Mohammed Berrada. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6:52138–52160.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging Linguistic Structure For Open Domain Information Extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing,* 344–354. Beijing, China, July.

Satanjeev Banerjee and Alon Lavie. 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,* 65–72. Ann Arbor, Michigan, USA, June.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating Evaluation in Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing,* 9347–9359. Online, November.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356 (6334): 183–186.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the Original: Fact Aware Neural Abstractive Summarization. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence,* 4784–4791. New Orleans, Louisiana, USA.

Nofar Carmeli, Xiaolan Wang, Yoshihiko Suhara, Stefanos Angelidis, Yuliang Li, Jinfeng Li, and Wang-Chiew Tan. 2021. Constructing Explainable Opinion Graphs from Review. In *Proceedings of the 30th Web Conference.* Online, April.

Yen-Chun Chen and Mohit Bansal. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics,* 1:675–686. Melbourne, Australia, July.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of 27th Conference on Neural Information Processing Systems: Workshop on Deep Learning.* Montreal, Canada, December.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence Mover's Similarity: Automatic Evaluation for Multi-Sentence Texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* 2748–2760. Florence, Italy: Association for Computational Linguistics, July.

Kevin Clark and Christopher D. Manning. 2016. Deep Reinforcement Learning for Mention-Ranking Coreference Models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* 2256–2262. Austin, Texas, USA, November.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment,* 177–190. Southampton, UK, April.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A Survey of the State of Explainable AI for Natural Language Processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing,* 447–459. Suzhou, China, December.

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,* 4171–4186. Minneapolis, Minnesota, USA, June.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-Fact Correction in Abstractive Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing,* 9320–9331. Online, November.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 5055–5070. Online, July.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 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,* 2214–2220. Florence, Italy, July.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using Local Knowledge Graph Construction to Scale Seq2Seq Models to Multi-Document Inputs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing,* 4186–4196. Hong Kong, China, November.

Saadia Gabriel, Antoine Bosselut, Jeff Da, Ari Holtzman, Jan Buys, Kyle Lo, Asli Celikyilmaz, and Yejin Choi. 2021. Discourse Understanding and Factual Consistency in Abstractive Summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics,* 435–447. Online, April.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software,* 1–6. Melbourne, Australia, July.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. Assessing The Factual Accuracy of Generated Text. In *Proceedings of the 25th International Conference on Knowledge Discovery + Data Mining,* 166–175. Anchorage, Alaska, USA.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys* 51 (5): 1–42.

Wang Haonan, Gao Yang, Bai Yu, Mirella Lapata, and Huang Heyan. 2020. Exploring Explainable Selection to Control Abstractive Summarization. arXiv: 2004.11779 [cs.CL].

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,* 28:1693–1701. Montreal, Canada.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9 (8): 1735–1780.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% Solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers,* 57–60. New York City, USA, June.

Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge Graph-Augmented Abstractive Summarization with Semantic-Driven Cloze Reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 5094–5107. Online, July.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. 2019. CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence,* 590–597. Honolulu, Hawaii, USA, July.

Dan Jurafsky and James H. Martin. 2009. Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition. Upper Saddle River, New Jersey, USA: Pearson Prentice Hall.

Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. 2019. Text summarization from legal documents: a survey. *Artificial Intelligence Review* 51 (3): 371–402.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2020. FFCI: A Framework for Interpretable Automatic Evaluation of Summarization. arXiv: 2011.13662 [cs.CL].

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural Text Summarization: A Critical Evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing,* 540–551. Hong Kong, China, November.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the Factual Consistency of Abstractive Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing,* 9332–9346. Online, November.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Analyzing Sentence Fusion in Abstractive Summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization,* 104–110. Hong Kong, China, November.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 7871–7880. Online, July.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep Text Classification Can be Fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence,* 4208–4215. Stockholm, Sweden, July.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out,* 74–81. Barcelona, Spain, July.

Chin-Yew Lin, Guihong Cao, Jianfeng Gao, and Jian-Yun Nie. 2006. An Information-Theoretic Approach to Automatic Evaluation of Summaries. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference,* 463–470. New York City, USA: Association for Computational Linguistics, June.

Hui Lin and Vincent Ng. 2019. Abstractive Summarization: A Survey of the State of the Art. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence,* 9815–9822. Honolulu, Hawaii, USA, July.

Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing,* 3730–3740. Hong Kong, China, November.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv: 1907.11692 [cs.CL].

Chi-kiu Lo. 2019. YiSi - a Unified Semantic MT Quality Evaluation and Estimation Metric for Languages with Different Levels of Available Resources. In *Proceedings of the Fourth Conference on Machine Translation,* 507–513. Florence, Italy, August.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* 3219–3232. Brussels, Belgium, October.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics,* 3036–3046. Minneapolis, Minnesota, USA, June.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 1906–1919. Online, July.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* 404–411. Barcelona, Spain, July.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. 2021. Entity-level Factual Consistency of Abstractive Text Summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics,* 2727–2733. Online, April.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* 1797–1807. Brussels, Belgium, October.

Juri Opitz and Anette Frank. 2021. Towards a Decomposable Metric for Explainable Evaluation of Text Generation from AMR. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics,* 1504–1518. Online, April.

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,* 311–318. Philadelphia, Pennsylvania, USA, July.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* 2227–2237. New Orleans, Louisiana, USA, June.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting Future N-gram for Sequence-to-SequencePre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing,* 2401–2410. Online, November.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report. OpenAI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21 (140): 1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics,* 2:784–789. Melbourne, Australia, July.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* 2383–2392. Austin, Texas, November.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing,* 3982–3992. Hong Kong, China, November.

Steffen Remus, Manuel Kaufmann, Kathrin Ballweg, Tatiana von Landesberger, and Chris Biemann. 2017. Storyfinder: Personalized Knowledge Base Construction and Management by Browsing the Web. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management,* 2519–2522. Singapore, Singapore.

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,* 1073–1083. Vancouver, Canada, July.

Peng Shi and Jimmy Lin. 2019. Simple BERT Models for Relation Extraction and Semantic Role Labeling. arXiv: 1904.05255 [cs.CL].

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised Open Information Extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics,* 885–895. New Orleans, Louisiana, USA, June.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP,* 191–200. Vancouver, Canada, August.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems,* 30:6000–6010. Long Beach, California, USA.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations.* Vancouver, Canada, April.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, Relation, and Event Extraction with Contextualized Span Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing,* 5784–5789. Hong Kong, China, November.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and Answering Questions to Evaluate the Factual Consistency of Summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 5008–5020. Online, July.

Qingyun Wang, Lifu Huang, Zhiying Jiang, Kevin Knight, Heng Ji, Mohit Bansal, and Yi Luan. 2019. PaperRobot: Incremental Draft Generation of Scientific Ideas. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* 1980–1991. Florence, Italy, July.

Xiaolan Wang, Yoshihiko Suhara, Natalie Nuno, Yuliang Li, Jinfeng Li, Nofar Carmeli, Stefanos Angelidis, Eser Kandogann, and Wang-Chiew Tan. 2020. ExtremeReader: An Interactive Explorer for Customizable and Explainable Review Summarization. In *Companion Proceedings of the Web Conference 2020,* 176–180. Taipei, Taiwan.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics,* 1112–1122. New Orleans, Louisiana, USA, June.

Thomas Wolf et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations,* 38–45. Online, October.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, C. Alberti, S. Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, L. Yang, and A. Ahmed. 2020. Big Bird: Transformers for Longer Sequences. In *Advances in Neural Information Processing Systems,* 33:17283–17297. Online, December.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning,* 11328–11339. Vienna, Austria, July.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *Proceedings of the 8th International Conference on Learning Representations.* Accepted as poster. Online, April.

Yuhao Zhang, Daisy Yi Ding, Tianpei Qian, Christopher D. Manning, and Curtis P. Langlotz. 2018. Learning to Summarize Radiology Findings. In *Proceedings of the 9th International Workshop on Health Text Mining and Information Analysis,* 204–213. Brussels, Belgium, October.

Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D. Manning, and Curtis Langlotz. 2020. Optimizing the Factual Correctness of a Summary: A Study of Summarizing Radiology Reports. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 5108–5120. Online, July.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing,* 563–578. Hong Kong, China, November.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing Factual Consistency of Abstractive Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics,* 718–733. Online, June.

A Appendices

A.1 Examples of unfaithful summaries

We found the following examples in the XSUM hallucination dataset by Maynez et al. (2020). Please consider them to get a feeling of the importance of faithfulness for automatically generated summaries.

Summary	Source
New rules have come into place that **you can eat your dog**.	The restaurant began serving puppy platters after a new law was introduced **allowing dogs to eat at restaurants** – as long as they were outdoors! It looks like a right dog's dinner – check out this clip.
An elephant has been hit by a stone at a zoo in western france after it was hit by a tree.	**The stone got past the elephant's fence** and a ditch separating the animal and visitors, the zoo said in a statement. **The girl** was taken to hospital and **died** within a few hours, the zoo added. The zoo statement said the enclosure met international standards and said "this kind of accident is rare, unpredictable and unusual". The statement went on (in French) to point out two other recent incidents in the US: Phyllis Lee, Scientific Director of the Amboseli Trust for Elephants, says that **targeted throwing of stones and branches by elephants is very unusual**. "It can happen when elephants are frustrated or bored. In my opinion, it's unlikely the elephant was directly targeting the girl – but exhibiting frustration. You can't predict what animals in captivity will do." The moments after **the girl was struck** at Rabat Zoo on Tuesday were filmed by a bystander and uploaded onto YouTube. The video shows the elephant waving its trunk behind a fence and swerves round to show a stone on the ground. Metres away people are gathered around the girl, holding her head and stroking her leg.

Table A.1: Exemplary unfaithful summaries found in the XSUM hallucination dataset by Maynez et al. (2020).

Summary	Source
A man who was convicted of **murdering his former girlfriend** has been **struck by a judge** in the us.	Officials believe Lawrence Phillips, 40, killed himself early on Wednesday. In the 1990s, he was a star player at the University of Nebraska, making him a sought-after rookie in the NFL. Phillips played for the St Louis Rams, Miami Dolphins and San Francisco 49ers until 1999, but he was plagued by trouble on and off the field. After his NFL career, Phillips continued to have problems with the law. Eventually, he was sent to prison after being convicted of assault for driving his car into a group of teenagers. In another case, he was convicted of **assaulting a former girlfriend**. Phillips, who was serving a term of more than 30 years, had been placed in a cell alone after he was accused of **killing his former cellmate**. On Tuesday, **a judge found that there was enough evidence** to try Phillips in the death of Damion Soward, 37. It is unclear how he took his own life. "He was an intelligent person and had some good qualities," Tom Osborne, his coach at Nebraska, told the Associated Press news agency. "Yet he had some anger issues and couldn't overcome the demons in his life."
Chinese police say they have seized more than **1, 000 tonnes of toxic powder** seized after at least **three people were allegedly smuggled into the city** last year.	More than **5,300 bottles of alcohol** were seized by the investigators in the southern city of Liuzhou. They also found **packets of a white powder** called Sildenafil, better known as the anti-impotence drug Viagra. Police in the Guangxi region are now investigating the two distillers. The Liuzhou Food and Drug Administration said (in Chinese) that the powder was added to three different types of 'baijiu' – a strong, clear spirit that is the most popular drink in China. They said the haul was worth up to 700,000 yuan. China continues to face widespread food safety problems. In June, police in cities across China seized more than **100,000 tonnes of smuggled meat**, some of which was more than 40 years old. The 2008 **tainted milk scandal** outraged the nation. Some **300,000 people were affected** and at least six babies died after consuming milk adulterated with melamine.

Table A.2: Exemplary unfaithful summaries found in the XSUM hallucination dataset by Maynez et al. (2020).

A.2 Sentence re-ranking experiments

The sentence ranking experiment by Falke et al. (2019) consists of 373 triples containing one source sentence, one faithful and one unfaithful summary sentence. The goal is to rank the faithful sentence higher than the unfaithful sentence. We experiment with all faithfulness metrics explained in Chapter 3. Here, we list the results for all metrics where multiple experiments were necessary to determine the best setup achieving the highest possible score.

Question Generation & Question Answering

QG Model	QA Model	Answer Similarity	Correct Predictions (in %)
bart-large-newsqa	bert-large-squad2	F1	55.23
t5-small	roberta-base-squad2	F1	65.95
t5-small	roberta-base-squad2	EM	66.76
t5-small	roberta-base-squad2	bert_precision	65.95
t5-small	roberta-base-squad2	ss_precision	67.83
t5-base	roberta-base-squad2	F1	65.95
t5-base	roberta-base-squad2	EM	62.73
t5-base	roberta-base-squad2	bert_precision	65.68
t5-base	roberta-base-squad2	ss_precision	68.36
t5-small	**roberta-large-squad2**	**F1**	**71.85**
t5-small	roberta-large-squad2	EM	69.71
t5-small	roberta-large-squad2	bert_precision	71.58
t5-small	roberta-large-squad2	ss_precision	69.71
t5-base	roberta-large-squad2	F1	68.36
t5-base	roberta-large-squad2	EM	67.29
t5-base	roberta-large-squad2	bert_precision	69.17
t5-base	roberta-large-squad2	ss_precision	69.71

Table A.3: Results of the question generation & question answering framework on the sentence ranking experiment from Falke et al. (2019). We experiment with two different question generation models, two different question answering models as well as all answer similarity metrics described in Section 3.9.

Entailment

Entailment Model	Method	Correct predictions (in %)
roberta-large-mnli	**s2s/d2s/top2s**	**88.47**
bart-large-mnli	s2s/d2s/top2s	77.21
bart-large-mnli-512	s2s/d2s/top2s	77.21

Table A.4: Results of the entailment faithfulness metric on the sentence ranking experiment from Falke et al. (2019). We experiment with various entailment models that are trained on the Multi-NLI (MNLI) dataset.

Sentence Similarity

Model	Similarity metric	Correct predictions (in %)
-	F1	56.03
-	EM	2.95
roberta-large-mnli	**bert_precision**	**69.71**
nli-bert-large	ss_precision	58.18
stsb-bert-large	ss_precision	65.42
stsb-roberta-large	ss_precision	65.15
paraphrase-distilroberta-base-v1	ss_precision	68.36

Table A.5: Results of the sentence similarity faithfulness metric on the sentence ranking experiment from Falke et al. (2019). We experiment with various models from the Sentence BERT library (https://www.sbert.net/index.html) as well as with all similarity metrics described in Section 3.9.

Open Information Extraction

Model	Similarity metric	Correct predictions (in %)
-	F1	46.11
-	EM	26.27
roberta-large-mnli	**bert_precision**	**49.06**
roberta-large-mnli	bert_recall	43.16
roberta-large-mnli	bert_F1	48.26
paraphrase-distilroberta-base-v1	ss_precision	47.99
paraphrase-distilroberta-base-v1	ss_recall	46.38
paraphrase-distilroberta-base-v1	ss_F1	47.45
-	RMR1	21.98
-	RMR2	26.27

Table A.6: Results of the open information extraction metric on the sentence ranking experiment from Falke et al. (2019). We experiment with the relation matching rate (RMR) and all similarity metrics described in Section 3.9.

Named Entity Recognition

Model	Similarity metric	Correct predictions (in %)
-	F1	18.50
-	EM	18.50
roberta-large-mnli	bert_precision	24.13
roberta-large-mnli	bert_recall	24.93
roberta-large-mnli	bert_F1	26.54
paraphrase-distilroberta-base-v1	ss_precision	28.69
paraphrase-distilroberta-base-v1	ss_recall	28.15
paraphrase-distilroberta-base-v1	**ss_F1**	**29.49**

Table A.7: Results of the named entity recognition metric on the sentence ranking experiment from Falke et al. (2019). We experiment with all similarity metrics described in Section 3.9.

Semantic Role Labeling

Model	Similarity metric	Correct predictions (in %)
-	F1	66.76
-	EM	50.67
roberta-large-mnli	bert_precision	66.49
roberta-large-mnli	bert_recall	64.61
roberta-large-mnli	bert_F1	67.83
paraphrase-distilroberta-base-v1	ss_precision	68.10
paraphrase-distilroberta-base-v1	ss_recall	67.56
paraphrase-distilroberta-base-v1	**ss_F1**	**69.44**

Table A.8: Results of the semantic role labeling faithfulness metric on the sentence ranking experiment from Falke et al. (2019). We experiment with all similarity metrics described in Section 3.9.

A.3 SRL labelset

We group the labels of the OntoNotes labelset to increase the robustness of the semantic role labeling (SRL) model and to ultimately increase the performance of the faithfulness metric based on SRL.

Our reduced lables	OntoNotes labels
Subject	ARG0, ARGM-COM, ARGA
Verb	V
Object	ARG1
Argument	ARG2, ARG3, ARG4, ARGM-PRD
Negation	ARGM-NEG
Location	ARGM-LOC
Direction	ARGM-DIR
Reasons	ARGM-CAU, ARGM-PRP, ARGM-PNC, ARGM-GOL
How	ARGM-MNR, ARGM-EXT, ARGM-ADV, ARGM-ADJ
When	ARGM-TMP

Table A.9: Mapping from the large OntoNotes labelset to our reduced variant. Please note that the OntoNotes labelset includes many more labels that either do not appear in our dataset or are filtered out completely as these labels do not appear often or do not have an important meaning w.r.t faithfulness.