Karlsruhe Institute of Technology

Evaluating Factual Correctness of Abstractive Query-based long Dialogue and Meeting Summarization

Bachelor's Thesis of

Jannik Weiß

at the Department of Informatics
Institute for Anthropomatics and Robotics

Reviewer:	Prof. Dr. Alexander Waibel
Second reviewer:	Prof. Dr. Jan Niehues
Advisor:	M.Sc. Stefan Constantin

29. March 2022 – 28. July 2022

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 28th July 2022

. . . . . .

(Jannik Weiß)

Abstract

Abstractive long document summarization remains a difficult problem in natural language processing, yet the use cases - especially for long meeting summarization - are obvious. Too often do summaries contain factual inconsistencies. In order to build better summarization models for long document summarization it is integral to be able to measure the factual correctness of generated summaries, which itself is by no means a trivial task. In this thesis I take two promising metrics for factual correctness in summarization and adapt them specifically for long meeting summarization. As a result the metrics are able to process longer source documents which increases their performance. To show that reporting on factual correctness in summarization research is a meaningful addition to standard ROUGE scores I apply the two adapted metrics to DialogLED - a model specifically trained for long meeting summarization.

Zusammenfassung

Abstraktive Zusammenfassung von langen Dokumenten ist weiterhin ein schwieriges Problem in der Sprachverarbeitung. Gleichzeitig sind die Anwendungsfälle - besonders für die Zusammenfassung von langen Meetings - leicht zu sehen. Zu häufig enthalten generierte Zusammenfassungen faktische Fehler. Um bessere Modelle zur Zusammenfassung zu entwickeln ist es wichtig die faktische Korrektheit messen zu können, was an sich schon keine einfache Aufgabe ist. In dieser Thesis beleuchte ich zwei vielversprechende Metriken zur faktischen Korrektheit für Zusammenfassung und adaptiere diese speziell für die Zusammenfassung von langen Meetings. Dadurch können die Metriken längere Quelldokumente verarbeiten, was ihre Leistung verbessert. Um zu zeigen, dass es eine sinnvolle Ergänzung zu ROUGE Scores ist in Forschung zu Zusammenfassung auch über faktische Korrektheit zu berichten, wende ich die zwei Metriken auf das DialogLED Modell an, welches speziell für Zusammenfassung von langen Meetings trainiert ist.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

Good Summarization - whether human or machine generated - arguably is the most powerful method for high speed information gathering. Lots of people spend a lot of time summarizing content for others, including myself as I wrote the abstract of this thesis. Similar to other natural language processing tasks like translation or text classification, the practical applications of a good automatic summarization system are easy to see. Access to a short and concise summary of a larger body of text will save the reader time or allow them to get an overview of a greater variety of documents in the same amount of time. Combined with automatic speech recognition, automatic summarization of meeting transcripts can provide others easy access to the content of a missed meeting. Additionally, when training a model for query-based summarization, the summary can be individually tuned by prepending the input with a query like "What did person A say about topic b?". This way summaries can become even more specific to the user's needs. So what's holding back widespread use of query-based meeting summarization?

Long meeting summarization is difficult even for today's state of the art summarization systems. Research has shown that around 30 % of model-generated summaries contain factual errors [11]. But to create better summarization systems it is essential to be able to evaluate the goodness or factual correctness of a summary automatically.

In contrast to other natural language processing tasks summarization is very difficult to evaluate automatically, let alone manually. The evaluation of tasks like the classification of text into classes of sentiment for example is rather easy, because there is one ground truth label. In this respect the task of translation is more difficult, because there are a number of correct translations for a phrase. However translation has the advantage that it can be divided and conquered at a sentence level. The difficulty with summarization is that in order to summarize correctly a larger context of the document is needed and a summarization system also needs to determine what the important parts of a source text are. Therefore there is a plethora of correct solutions to any particular summarization problem, which calls for advanced automatic evaluation metrics that take factual correctness into account.

Another problem for summarization is that lots of current state of the art summarization models limit their input to 512 or 1024 tokens due to their transformer architecture [22]. This architecture makes heavy use of the attention mechanism whichs space requirement scales quadratically with the input length, thus prohibiting a practical use of input lengths above the mentioned limits.

To address the latter problem of input length a few methods have been proposed, among the most popular of which is the longformer [1] which substitutes the attention with a variant of it that only scales linearly with the input length.

Concerning the first problem, traditionally ROUGE scores [12] are used to evaluate and report the accuracy or correctness. However these fail to capture factual correctness well, because they only compare n-gram overlaps between a reference and a generated summary. Several methods have been proposed to better automatically capture the factual correctness of generated summaries. These methods often themselves make use of other natural language processing tasks and models. This has the effect that all of these methods can also only process inputs of up to 512 or 1024 tokens, since they report their results usually using standard transformer models like BERT.

1.2. Objective

The objective of this thesis is twofold:

1. Select promising factual correctness evaluation metrics and adapt them to be able to process long meeting transcript input. Evaluate the evaluation performance compared to the metric in its standard version.

2. Use the adapted metrics to gain more insight about the DialogLED model [25], which is a state-of-the-art long meeting and dialogue summarization model. This model has had special pretraining with transcript-style data, in order to better understand this type of text.

The remainder of this thesis is organized as follows: chapter 2 goes over fundamentals in summarization and evaluation metrics for summarization, chapter 3 highlights related work, chapter 4 and chapter 5 explain the adaptation of two factual correctness metrics for long meeting summarization and their application to two summarization models. Lastly, chapter 6 includes conclusion and further work sections.

2. Fundamentals

2.1. Summarization

Summarization is a sequence-to-sequence task, which means that the input is a sequence and the output is also a sequence. The sequence that serves as the input to a summarization model, however, is usually not the plain text, but a sequence of tokens. A token is an integer that represents a word, a punctuation mark, a subword, or in some cases also a single character. Therefore the first and last steps of an all encompassing summarization framework are usually tokenization steps in which a so called tokenizer encodes plain text to tokens or decodes tokens to plain text.

In Meeting summarization the process usually begins one step earlier with automatic speech recognition (ASR) to create a transcript. Like in summarization, ASR methods also make use of different types of neural networks, starting with TDNNs in 1989 [23].

2.1.1. Extractive and Abstractive Summarization

Summarization can be extractive or abstractive. Extractive summarization is a method which selects specific sentences from the source text and concatenates them to form a summary. This can formally be defined as follows. Let the source text T be a list of sentences: $T = (s_1, s_2, \ldots, s_N)$. Let $Score(T) = (v_1, v_2, \ldots, v_N)$ be the scores of the sentences according to the scoring function $Score$. The next step is to select the k sentences with the highest scores and concatenate them to form a summary. Potentially there might come additional clean up steps after the concatenation to resolve pronouns for example. The source text might also be split with a different granularity than sentence-wise. Extractive summarization has the advantage that the output contains grammatically correct sentences, with the reasonable assumption that the source text is grammatically correct. However it is not possible to introduce new words or to summarize some information in a shorter way than its shortest description in the source document. Moreover, this is not typically the way humans create summaries or expect summaries to look and read like. Especially in the meeting and dialogue domain extractive summarization is inadequate, because of the structure of transcript-style texts.

In abstractive summarization the generated summary can be any sequence of words and is produced by the summarization system and not by selecting parts of the source text. Examples of such summarization systems are described in the next section. This method for summarization opens up the opportunity for novel words and sentences in the generated summaries and for possibly more natural sounding paraphrases and summaries. However the guarantee of grammatical correctness is lost. But more problematic is the challenge of factual correctness. While it is possible to create pathological examples in

which even an extractive summary may contain factual errors, this problem is much more prevalent in abstractive summarization. This is one major reason which is holding back the widespread use of current state-of-the-art abstractive summarization systems [28]. Nevertheless abstractive summarization has the greater potential and has benefitted a lot from recent developments in natural language processing (NLP), which is why the remainder of this thesis only concerns itself with abstractive summarization.

2.1.2. Query-Based Summarization

The task of query-based summarization sits between summarization and question answering. An input text for summarization is prepended with a query that can be a question (e.g. "What was the group's decision concerning X?") or a request (e.g. "Summarize the presentation of person Y"). This is especially useful when summarizing long documents, because it offers the possibility to adjust the summary to specific requests from the reader.

2.2. Summarization Systems

Currently most state-of-the-art NLP models are based on the transformer architecture which was introduced in 2017 [22]. Nevertheless it is worth to take a look at the history and to see what its differences are to previous state-of-the-are models.

2.2.1. RNNs

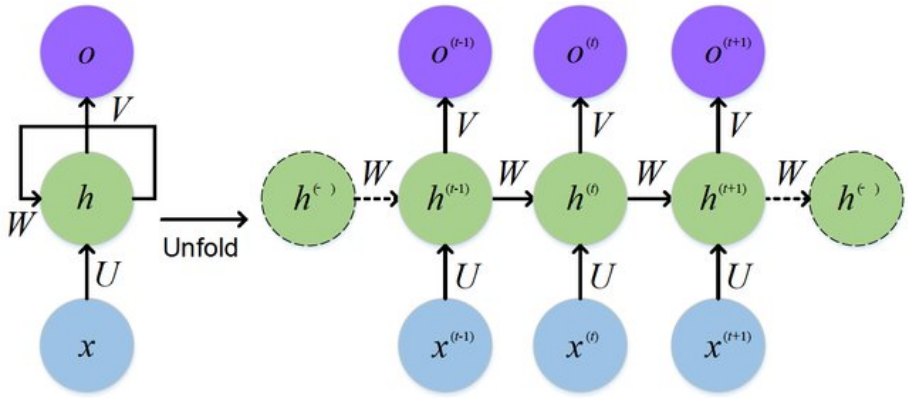

Figure 2.1.: Scheme of a Recurrent Neural Network [6]

A simple Multi-Layer-Perceptron (MLP) [17] is quite rigid in the sense that cannot really handle input or produce outputs of varying lengths. A Recurrent Neural Network (RNN) [3] builds on the MLP to be able to solve sequence-to-sequence problems. At each step the RNN takes one word from the input sequence and a so called hidden state and returns an output and an updated hidden state. This updated hidden state and the next word from the input sequence then serve as the input for the RNN in the next step. This can be described mathematically in the following way, where b and c are biases:

$$h^{t+1} = Ux^t + Wh^t + b \tag{2.1}$$

$$o^t = Vh^t + c \tag{2.2}$$

The central problem of RNNs is that of the vanishing gradient. Backpropagation of RNNs propagates the loss through several instances of the Network all the way back to the first word of the input sequence. Because of this the same derivative functions are applied multiple times, which can cause the effect of the words at the beginning of the sequence on the overall gradient to be very small. As a result a RNN is bad at learning connections of words which are far apart in a longer sequence.

2.2.1.1. LSTMs

LSTMs (Long Short Term Memory) [8] address this problem by introducing a more complex way to process the input and the hidden state with multiple gates which serve different purposes such as to forget or keep specific information. This makes it possible to put more focus on specific tokens in the hidden state or minimize the focus on insignificant tokens.

2.2.2. Transformers

The Transformer architecture is in large parts very different from the RNN architecture. It makes heavy use of the attention mechanism and processes the input sequence as a set rather than a sequence which makes it highly parallelizable. It consists mainly of encoder and decoder blocks as can be seen in Figure 2.2. Let's go through the architecture in detail.

Word and Embeddings and Positional Encodings

Even before embedding, a sequence of words needs to be converted to tokens. To create a useful input for the transformer each word or subword is mapped to an integer. Then for each token an embedding is created which is a learned vector representation of that token. These vectors already contain information about the similarity and relation of tokens with each other.

Since the transformer does not process tokens in a sequential manner, but rather all at the same time, their positional information (their order) is lost. To reintroduce this information a positional encoding is calculated for each token. The positional encoding is not dependent on the token itself, but is calculated from specific sine and cosine functions.

For each word its embedding and positional encoding are added to form the input vector to the first encoder of the transformer.

The Encoder

The encoder block consists of a Multi-Headed Self-Attention block, a feed forward neural network and two residual connections and normalization steps.

The attention mechanism is an integral part of the transformer. Its purpose is to determine which tokens of the input sequence to associate with which other tokens. Self

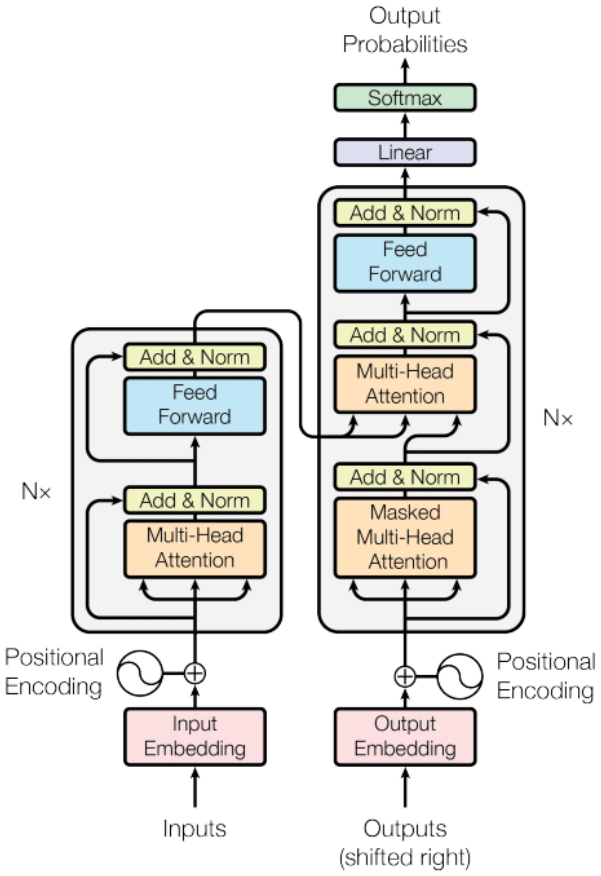

Figure 2.2.: Transformer Architecture. Left gray box: Encoder, right gray box: Decoder [22]

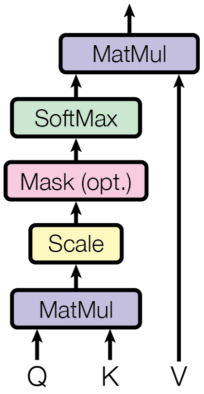

Figure 2.3.: Transformer Attention [22]

attention determines which tokens are most associated with which other tokens of the same sequence. This is done as follows. For each input vector of a token three vectors are generated, which are called the query, key and value vectors.

The dot product of the query vector of token A and the key vector of token B denotes how important B is when considering A in the given sequence. Doing this for all combinations of tokens in the sequence can be formulated as a matrix multiplication of the keys matrix and the transposed queries matrix where each row in either matrix is the respective key or query vector. This results in a quadratic matrix with the dimension of the sequence length. The values of this score matrix are then scaled based on the dimension of the key and query vectors which allows for more stable gradients to alleviate an exploding gradient effect. Finally the score matrix is put through a softmax function. The score matrix is then used to select so to say the values of the tokens which should be attended to the most by multiplying it with the value matrix. This gives us the output vectors of the attention layer for each token. These output vectors contain encoded information on how each word in the sequence should attend to other words in the sequence. Formally, the attention mechanism can be described as follows, with d_k being the dimension of the key and query vectors:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Attention in the transformer is multi-headed. This means that there are in the order of 8 heads which all generate their own query, key and value vectors and perform attention as described above. Their outputs are concatenated and then projected to the correct output dimensions. The reason for this is that different heads can learn different associations.

The attention output vectors are added to the original input vectors and put through a normalization layer, before going through a feed forward neural network with another residual connection as can be seen in Figure 2.2. The output of one encoder block are context sensitive word embeddings.

Multiple of these can be stacked on top of each other, each taking the output of the previous block as their input. The output of the last encoder block serves as part of the input to the decoder blocks.

The Decoder

The decoder has similar components as the encoder. However the attention layers of the decoder have slight differences to the one used in the encoder.

The input to the decoder is the sequence of tokens which it has already generated in the previous time steps. These are also embedded using token embeddings and positional encodings.

During training where the output sequence is known it is necessary to prevent tokens to attend to future tokens. This is achieved by masking the corresponding triangular half of the query-key-matrix.

To incorporate the information from the encoder the encoder outputs are used to create the key and value vectors for the encoder-decoder attention, which is also called cross-attention. The resulting output vectors encode information about which words of the input sequence to attend to given the current state of the output sequence.

The rest of the decoder is analogical to the encoder.

Transformer Output

The transformer uses the output of the last decoder to generate the next token in the output sequence. This is done through a linear layer which maps the decoder output to the vocabulary. Lastly a softmax layer produces actual probabilities for each token in the vocabulary. The token with the highest probability is chosen as the output token.

Training

The most successful transformer models have been trained on large text corpuses. The training task usually is some kind of masking task, also called denoising. Single tokens from a text are masked and the model makes a prediction as to what the masked token is. Since this is unsupervised very large amounts of texts can be used for this pretraining. These pretrained models can then be finetuned with other datasets for specific downstream tasks like text classification or summarization.

Advantages and Limitations

The two major advantages that the transformer has over the RNN are its parallelizability and its theoretically infinite attention span. Because the transformer does not process words in sequence, but as a set it is highly parallelizable. The transformer also does not suffer from the vanishing gradient problem like the RNN does. The attention span is only limited by the practicality of the query-key-matrix multiplication in the attention layer. This matrix multiplication however also limits the length of the input sequence that a transformer can process. Lots of transformer models therefore currently limit their input length to 512 or 1024 tokens.

2.2.2.1. BERT

BERT is a pretrained transformer model that was released by Google in 2019 [5]. BERT was pretrained with the "masked language model" (MLM) task and a "next sentence prediction" (NSP) task and achieved new state-of-the-art results on multiple natural language processing datasets. The MLM task takes a text and masks out random tokens which the model predicts. The NSP task lets the model predict the next sentence after a text prompt out of two given choices.

2.3. Evaluation Metrics

To be able to evaluate the performance of sequence-to-sequence models, metrics which can be automatically calculated are needed. Human evaluation is very expensive, but because automatic metrics are not perfect, human judgements are nevertheless an important benchmark.

Let's take a look at two popular automatic evaluation metrics.

2.3.1. ROUGE

By far the most popular and most reported metric when it comes to summarization are ROUGE scores. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) were introduced in 2004 [12]. There are multiple ROUGE scores which all compare n-gram overlaps between a generated text and a reference text. As an example, consider the following reference text R and the generated text G:

$R =$ "*The team discussed their objective.*"

$G =$ "*The team talked about their plan.*"

Rouge-1

The ROUGE-1 score measures the matching unigrams and is calculated as follows:

$$ROUGE1\ Precision(R,G) = \frac{unigrams\ in\ G\ that\ also\ appear\ in\ R}{\#\ of\ words\ in\ G} = 3/6 = 0.5$$

$$ROUGE1\ Recall(R,G) = \frac{unigrams\ in\ R\ that\ also\ appear\ in\ G}{\#\ of\ words\ in\ R} = 3/5 = 0.6$$

$$ROUGE1\ F1\ Score(R,G) = \frac{2 * (precision * recall)}{precision + recall} = 0.55$$

Rouge-2

The ROUGE-2 score is calculated in the same way, but with bigrams. In the given example R has 4 bigrams, G has 5, and there is one overlapping bigram.

Rouge-L

The ROUGE-L score calculates the ration between the longest common (not necessarily consecutive) subsequence and the number of unigrams of two texts. In the given example the longest common subsequence is "The team their". Precision and Recall are calculated as follows:

$$ROUGE-L\ Precision(R,G) = \frac{3}{6} = 0.5$$

$$ROUGE-L\ Recall(R,G) = \frac{3}{5} = 0.6$$

In general, when using ROUGE scores, the ROUGE F-1 scores are calculated and reported.

2.3.2. BLEU

The BLEU (Bilingual Evaluation Understudy) score [16] also uses the n-gram overlap precision calculation like the ROUGE score, but goes a bit further.

First, overlapping n-grams are only counted as often as they appear in the reference text, preventing something like "The the the the the" to receive a perfect score in the running example.

Then the geometric mean is taken over the precision scores based on 1-grams, 2-grams, 3-grams and 4-grams to reward more correct word order.

Lastly a brevity penalty is introduced which punishes texts like "The team" which would otherwise receive a perfect score.

In contrast to ROUGE scores which are more recall oriented (Which n-grams of the reference text appear in the generated text?), the BLEU score is more precision oriented (Which n-grams of the generated text appear in the reference text?).

2.3.3. Advantages and Shortcomings

The big advantage of the ROUGE and BLEU scores is that they are easy and fast to calculate and do correlate with human judgments [24]. However they fail to capture synonyms and paraphrases well. Chapter 3.2 gives examples of more advanced metrics which better capture factual correctness.

3. Related Work

3.1. Long Dialogue and Meeting Summarization

The input length limit which transformer models impose due to the memory consumption of the attention mechanism begs the question: How can good summaries for long documents be produced? Additionally, with the obvious use cases in mind, the following questions pose themselves: How can good summaries of dialogue or meeting transcripts be produced? How well can standard summarization models handle the differently structured text? That is, a normal text has the structure "[sentence]. [sentence]." while a dialogue or meeting transcript has the structure "[speaker]: [utterance]. [speaker]: [utterance]."

3.1.1. Data

To be able to train and evaluate long texts or transcripts, appropriate datasets are needed. Like for short text summarization there are datasets which leverage existing text sources. While the popular CNN/DailyMail summarization dataset [15] takes short news articles and their highlights as summaries, the PubMed dataset [21] for example takes biomedical research papers which are rather long as source texts and their abstracts as summaries. PubMed is an example of a large long document summarization dataset. There are also some large short dialogue summarization datasets like SAMSum [7] and DialogSum [2].

The most interesting datasets for the objective of this thesis are long meeting or dialogue summarization datasets, of which there do not exist many.

QMSum

Datasets	# Meetings	# Turns	# Len. of Meet.	# Len. of Sum.	# Speakers	# Queries	# Pairs
AMI	137	535.6	6007.7	296.6	4.0	-	97 / 20 / 20
ICSI	59	819.0	13317.3	488.5	6.3	-	41 / 9 / 9
Product	137	535.6	6007.7	70.5	4.0	7.2	690 / 145 / 151
Academic	59	819.0	13317.3	53.7	6.3	6.3	259 / 54 / 56
Committee	36	207.7	13761.9	80.5	34.1	12.6	308 / 73 / 72
All	232	556.8	9069.8	69.6	9.2	7.8	1,257 / 272 / 279

Figure 3.1.: QMSum statistics. Top half: existing datasets, bottom half: QMSum. [26]

QMSum was published very recently and contains transcripts of different kinds of long meetings [26]. It includes the transcripts from the AMI [14] and ICSI [10] Datasets, and

from committee meetings in the Welsh and Canadian parliament. According to the authors it is the first meeting dataset that includes query-based summarization. For each meeting between 3 and 12 query-based summaries are given in addition to the general summary of the meeting. The advantages of QMSum are that the meetings are for the most part held in a rather private setting which makes people speak more naturally, and the summaries are human generated. The disadvantage is that the dataset is very small.

MediaSum

Statistics	NPR	CNN
Dialogues	49,420	414,176
Avg. words in dialogue	906.3	1,630.9
Avg. words in summary	40.2	11.3
Turns	24.2	30.7
Speakers	4.0	6.8
Novel summary words	33.6%	24.9%

Figure 3.2.: MediaSum statistics [29]

MediaSum is a collection of transcripts with abstractive summaries from interview segments on CNN and NPR [29]. The meetings are not as long as the ones in QMSum, but still on average definitely longer than 512 tokens. In contrast to QMSum this dataset is very large, which is an advantage. On the other hand, the summaries - while human generated - are very short and I believe they also need to serve the purpose of getting a reader or listener interested in the interview, rather than summarizing everything that is said.

3.1.2. Models

HMNet

HMNet was introduced my Microsoft Research in 2020 [27]. It is specifically designed for transcript summarization and specifically encodes speaker information. It is able to summarize long transcripts by employing a hierarchical approach which first encodes individual turns into one embedding and then processes these turn embeddings in a second step.

DialogLED

DialogLED [25] is a Longformer-Encoder-Decoder (short LED) model with additional pretraining. A LED is a transformer model with an adapted attention mechanism that scales only linearly with the input length if it exceeds 512 tokens (see section 4.1). The authors use a window-based denoising task on meeting transcript data for this pretraining. A specific window containing multiple turns of a transcript is selected and then different kinds of noise are introduced inside this window. Specific words or speakers might be

masked or turns are splitted or merged. Then the model is tasked with recovering the original content. The authors state that this pretraining lets the model better understand the structure of meeting or dialogue transcripts, while not being specifically trained for a downstream task. They verify their idea by reporting a small improvement of DialogLED over LED in summarization. The model also outperforms HMNet.

3.2. Factual Summary Evaluation Metrics

Popular evaluation metrics like ROUGE can not capture factual correctness of summaries very well. A variety of methods have been proposed to improve in this area, most of which themselves make use of transformer models.

3.2.1. Entailment

factCC was introduced in 2019 [11] and has been used as an evaluation metric in a number of other papers ([19], [28], [9]). The authors formulate factual consistency as a classification task. They use a BERT transformer model which takes the concatenation of a source text and a claim sentence and makes a binary classification of either CONSISTENT or INCONSISTENT. A good and factually correct summary will then receive high consistency probabilities for each sentence of the summary. To train the sequence classification model a dataset was created by applying a set of rule-based transformations on sentences of source documents. These were taken from the CNN/DailyMail dataset. The metric and data generation are explained in detail in chapter 4.

3.2.2. Question Generation and Question Answering

QuestEval [20] improves on previous work in factual correctness evaluation through question generation and question answering. The authors use T5 [18] transformer models to generate questions from the source document and the generated summary and measure the similarity between the answers when conditioned on either the source document or the generated summary. A higher similarity suggests a higher factual consistency. The authors conclude that their framework improves evaluation performance because it combines recall and precision oriented question generation and question answering applications.

3.2.3. Counterfactual Estimation

FactualCoCo [24] is a metric that determines factual correctness via counterfactual estimation. The idea is that a summary is more factually consistent if its words were generated more reliant on the source document than reliant on the language prior. For example a text might mention purple bananas (they exist!) and a summarization model might hallucinate about yellow bananas when generating the summary. This is probably because its general knowledge about bananas outweighed the specific case of purple bananas in this particular source text. In this case the words were generated more reliant on the language prior than on the source document. How this is quantified is explained later in subsection 4.2.2. The

authors also find that their evaluation metric correlates better with human judgments, especially when compared to ROUGE metrics, and it also performs better than QuestEval. Like all other previously mentioned evaluation metrics, however, the authors report and publish their scores and models only for short text summarization.

4. Methods and Models

In the DialogLED paper the authors only report ROUGE scores when comparing it with the standard longformer transformer model. But how do the two compare when looking at factual correctness? Possibly there are differences that the ROUGE scores do not capture. Applying factual correctness metrics here can give more insight into the effect of the meeting specific pretraining, which is supposed to set DialogLED apart.

At the same time it is worth to explore if it even brings a significant advantage to adapt these metrics to handle longer inputs.

4.1. The Longformer

The longformer [1] is a variant of the transformer with an attention mechanism which scales only linearly with the input length. This allows the longformer to process more input tokens. It is otherwise based on RoBERTa [13], which refines the pretraining from BERT and uses different pretraining and hyperparameters than BERT.

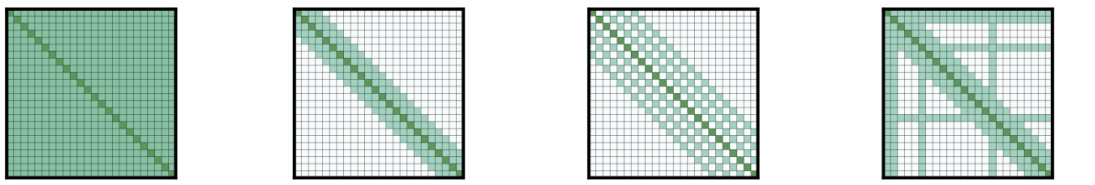

Figure 4.1.: Longformer Attention [1]

The longformer computes only a part of the query-key matrix in the attention layer using a sliding window approach. The window size however is usually 512 tokens, so the linear growth really only begins when the length of tokens is reached where standard transformers would start to truncate the input. Different attention heads can also attend to differently dilated windows to enlarge the context. Lastly global attention can be assigned to specific tokens and is symmetric. This is useful for example in query-based summarization where global attention would be set for the query tokens. Formally the memory requirements of the longformer attention can be written as follows with sequence length n, window size w and s tokens with global attention:

$$Memory = (n * w) + (2 * n * s)$$

4.2. Evaluation Metrics for long inputs

4.2.1. long FactCC

Model

The factCC metric classifies document-sentence pairs as consistent or inconsistent and therefore the model is given the entire source document together with the claim sentence as input. The BERT model which is published together with the paper truncates its input after 512 tokens. To build a long version of a factCC model I choose a longformer model. Since this is a classification task, it is an encoder only longformer model with a classification head at the top. This is a linear layer which maps the last encoder's output to the two classes CONSISTENT and INCONSISTENT.

Data Generation

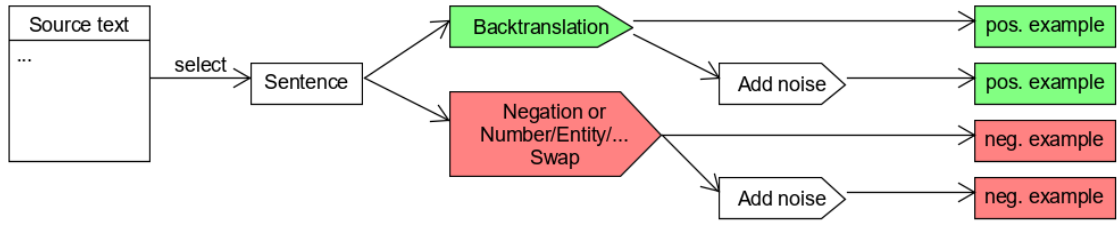

Figure 4.2.: factCC data generation process

The data used to train a factCC model is generated by applying rule-based transformations to sentences from the source text. These are:

- **Backtranslation:** A sentence is translated into other languages and then back to English using google translate. These are positive (i.e. consistent) examples.
- **Pronoun swap:** A random pronoun in the claim sentence is swapped. This - like all other swap operations - produces a negative (i.e. inconsistent) example.
- **Date swap:** A random date in the claim sentence is swapped with one that appears in the source text.
- **Number swap:** A random number in the claim sentence is swapped with one that appears in the source text.
- **Entity swap:** A random entity name in the claim sentence is swapped with one that appears in the source text.
- **Negation:** The meaning of the claim sentence is negated. This produces a negative example
- **Noise:** A random word in the claim sentence is either removed or duplicated. This is applied to both positive and negative examples.

Sentences are randomly selected from the source texts and transformations are applied if possible. Some transformations rely on named-entity recognition (NER) tagging, so that words are not swapped at random, but according to the specific category.

Provided Data

The authors of the factCC Paper provide a dataset with around 1.8 million examples created from the CNN and DailyMail stories using this method. They also provide a subset of this dataset containing around 1 million examples where the claim sentence appears within the first 512 tokens of the source text. They call this subset the clipped version of the entire dataset.

I use these datasets to reproduce a factCC BERT model as it was presented in the paper and to train two variants of a longformer with different input length limits, in order to be able to compare them.

Additionally, the authors provide a small dataset which they annotated themselves. It contains CNN/DailyMail stories as source texts and summary sentences created from several different models as claims. It includes 441 positive and 62 negative examples. I call this dataset MAT (Manually Annotated Testset).

Created Data

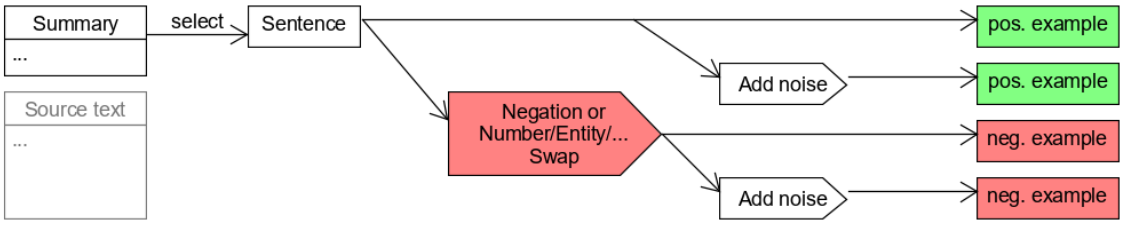

Figure 4.3.: factCC data generation process using summary sentences

I further create a similar dataset using the data generation methods from factCC on the basis of the MediaSum dataset. For this dataset I do not select sentences from the source text as claims, because the source text of MediaSum is an interview transcript. Instead I select sentences from the provided summaries as claims, because these are sentences which would potentially appear in a summary, as opposed to a "speaker: utterance" type of expression from the transcript. These sentences which I use as claims also are quite abstractive with regard to their source text, which is why in the data generation I do not need the backtranslation transformation.

I use the this generated dataset based on MediaSum to further train the factCC Model variants, in order to potentially improve their performance on transcript-style source texts.

GitHub Contributions

While using the resources provided in the factCC GitHub repository I spotted two typing errors for which I have submitted pull requests. One of them caused an error in a training

script and the other one was in the set of pronouns for the pronoun swap operation. It is thinkable that this might have cost the authors a very small percentage in the performance.

4.2.2. factualCoCo with a long scoring model

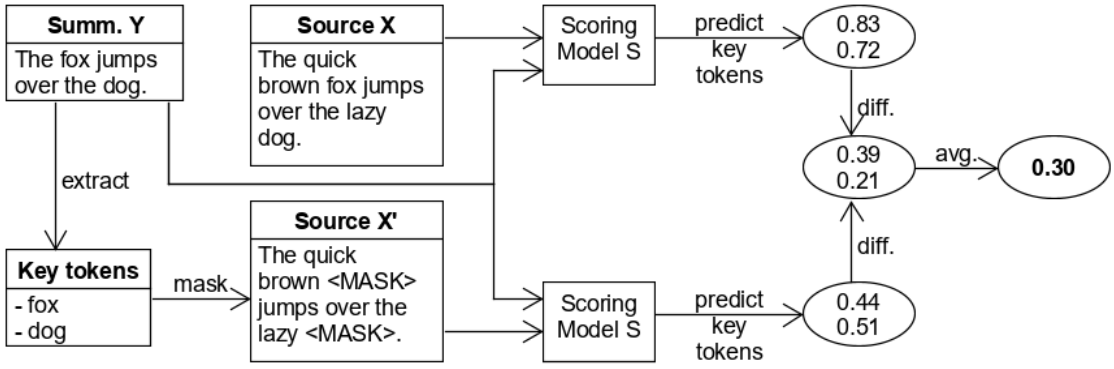

Figure 4.4.: Schematic description of the factualCoCo metric with an example

The factualCoCo metric quantifies how reliant on the source text a summary was generated as compared to reliant on the language prior. In order to determine this difference a few relevant key tokens from the generated summary are selected and are masked out in the source document. Then a standard summarization model is used to give the probabilities of the selected key tokens inside the summary when given the original document as the source text and when given the masked document as the source text. A good and factually consistent summary will result in higher differences of these probabilities. In order to be able to process longer input sequences I use different longformer models as the scoring model for factualCoCo.

4.3. Summarization Models and Evaluation Application

I use the created variants of factCC to evaluate a DialogLED and a LED model on the query-based summarization task with the QMSum dataset. This serves two purposes:

1. Comparison of the evaluation metrics among each other

2. Comparison of DialogLED and LED

To these ends I fine-tune both a DialogLED and a LED model on QMSum and evaluate their outputs of the dataset's test split with the various evaluation models.

I define the factCC score of a summarization model as follows: Let $T = (T_1, \ldots, T_n)$ be the source texts and $S = (S_1, \ldots, S_n)$ the corresponding generated summaries. Each summary S_k is a list of sentences $s_k^1, \ldots, s_k^{l_k}$. The score of a summary is the average of the individual sentence scores and the score of the model is the average of the individual summary scores. *SM* denotes a factCC score model.

$$factCC\ Score(SM, T, S) = \frac{1}{n} \sum_{k=1}^{n} \frac{1}{l_k} \sum_{j=1}^{l_k} SM(T_k, s_k^j)$$

I define the CoCo score of a summarization model as well as the average of the scores of the individual summaries.

5. Experiments

5.1. Generated Datasets

5.1.1. MediaSum for factCC

The data generation based on the MediaSum dataset (see section 4.2.1) with summary sentences as claims yielded enough data to create a dataset of 600 000 examples, 100 000 of which contain noise, balanced between positive and negative examples.

5.2. FactCC Variants

5.2.1. Provided Checkpoint and Reproduction

There is an official factCC checkpoint which is reported to have a 74 % balanced accuracy on MAT (the manually annotated testset). This checkpoint was trained from an uncased base BERT model checkpoint on the provided clipped dataset for 10 epochs. I reevaluated it and got the accuracies shown in Table 5.1, which results in a balanced accuracy of 73 %. The balanced accuracy is balanced by class, which in all cases here is the mean of correctly identified positive and the correctly identified negative examples.

		MAT	
		pos	neg
official	pos	0.901	0.452
checkpoint	neg	0.099	0.548

Table 5.1.: Accuracy scores of the official factCC checkpoint on the manually annotated testset

Interestingly the official checkpoint performs much better at detecting positive examples than at detecting negative examples.

To further have a model which can better be compared to other factCC models I trained an uncased base BERT model checkpoint in the same way, but for only 3 epochs. This model is hereafter called BERT-512. The resulting accuracies are shown in Table 5.2.

The resulting balanced accuracy for the BERT-512 model is 67 % and it has the same imbalance in the capability to detect positive and negative examples. This difference is not mentioned in the factCC paper, and neither is the imbalance of positive (441) and negative (62) examples in the MAT.

		MAT	
		pos	neg
BERT-512	pos	0.918	0.581
	neg	0.082	0.419

Table 5.2.: Accuracy scores of BERT-512 on the manually annotated testset

5.2.2. FactCC Longformer Models

To create a factCC model that can process longer inputs I trained a base longformer model with an input limit of 2048 tokens for one epoch on the entire provided CNN/DailyMail-based dataset. Even though input length limits of up to 16384 would be possible, the compromise of 2048 was needed to avoid really long training times. In order to better understand the difference that the longer input length limit makes, I also trained a base longformer model with an input length limit of 512 (like the baseline BERT-512 model) on the clipped dataset for 3 epochs. The accuracies of both models on the MAT are shown in Table 5.3 and Table 5.4. Their balanced accuracies are 54 % (longformer-512) and 61 % (longformer-2048).

		MAT	
		pos	neg
longformer-512	pos	0.830	0.740
	neg	0.170	0.260

Table 5.3.: Accuracy scores of longformer-512 on the manually annotated testset

		MAT	
		pos	neg
longformer-2048	pos	0.952	0.740
	neg	0.048	0.260

Table 5.4.: Accuracy scores of longformer-2048 on the manually annotated testset

Both models perform very bad at detecting negative examples, compared to the BERT models. Interestingly the longformer-512 model has the same input length limit and was trained for the same time on the same dataset as the reproduced BERT-512 model. This warrants a closer look at the 62 negative examples of the manually annotated test set.

5.2.3. A closer look at negative examples

Even though the examples of the MAT were generated from models that performed abstractive summarization, an empirical analysis of the 62 negative examples shows that they are highly extractive. In most cases large parts of the claims appear exactly the same in the source text as well, with the exception of one or two words. A few examples are

provided in Table A.1. This might lead models to falsely assume that these claims are correct.

Another aspect to look at is the distribution of correctly identified negative examples. I find that three quarters of the negative examples correctly identified by the longformer models are also correctly identified by the official BERT checkpoint.

5.2.4. Transfer to MediaSum-based data

In the end the objective is to evaluate meeting summarization data which is not exactly the domain of the CNN/DailyMail-based training data of the previously mentioned models. Therefore it is relevant to explore how the models trained on the CNN/DailyMail-based data transfer their performance to a test set based on meeting transcript type data.

MediaSum-based factCC test data

I have created a test dataset based on MediaSum as described in section 4.2.1, which contains 5000 positive and 5000 negative examples, and also 1000 positive examples with noise and 1000 negative examples with noise. In contrast to the MAT this testset does not have claim sentences which were generated by summarization models. This would be the ideal case, however creating such a dataset takes a prohibitively large labeling effort. Still the claim sentences are very abstractive and the source texts are transcripts, which is the data domain of interest for this thesis.

To increase the meaningfulness of the results on the MediaSum-based test set, I created a shuffled version of it as described in Figure 5.1. A model which has truly learned to predict the factual consistency of a claim with the source document should receive an accuracy score around 50 % on this shuffled version. If the model does as well on the shuffled version as it does on the non-shuffled version, then it must have somehow learned whether a positive, negative or no transformation has been applied to the claim, which is not useful.

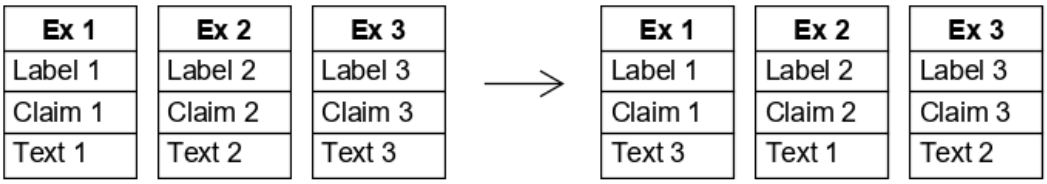

Figure 5.1.: MediaSum-based factCC data shuffle

Model Performance

Table 5.5, Table 5.6, Table 5.7, and Table 5.8 show the accuracies of the four previously mentioned models on the test dataset based on MediaSum. Their balanced accuracies are listed in Table 5.9.

The balanced accuracies show that the longformer models perform better than the BERT models. The major difference is that both BERT models perform very poorly at detecting

		MediaSum-based testset	
		pos	neg
official checkpoint	pos	0.314	0.024
	neg	0.686	0.976

Table 5.5.: Accuracy scores of the official factCC checkpoint on the MediaSum-based testset

		MediaSum-based testset	
		pos	neg
BERT-512	pos	0.366	0.041
	neg	0.634	0.959

Table 5.6.: Accuracy scores of BERT-512 on the MediaSum-based testset

		MediaSum-based testset	
		pos	neg
longformer-512	pos	0.792	0.398
	neg	0.208	0.602

Table 5.7.: Accuracy scores of longformer-512 on the MediaSum-based testset

		MediaSum-based testset	
		pos	neg
longformer-2048	pos	0.892	0.451
	neg	0.108	0.549

Table 5.8.: Accuracy scores of longformer-2048 on the MediaSum-based testset

Model	**MediaSum-based testset Balanced Accuracy**	**Shuffled**
official factCC checkpoint	64.5 %	55.4 %
BERT-512	66.2 %	55.3 %
longformer-512	69.7 %	66.0 %
longformer-2048	72.1 %	57.5 %

Table 5.9.: Balanced Accuracies of factCC Models on the MediaSum-based testset

positive examples, while the opposite was the case on the MAT. The longformer models on the other hand are better at detecting positive examples, although the difference is not as great as in the case of the BERT models. The likely reason for this is that because the claim sentences are so abstractive, the BERT models overwhelmingly predict them to be inconsistent with the source text. Consequently it appears that the longformer models are better at generalizing and transferring to other datasets. Among the longformer models the longformer-2048 outperforms the longformer-512, which is as expected, because it can process more input tokens of the source text. This advantage is also supported by the balanced accuracies of the shuffled dataset, which show that the longformer-512 improves

only by less than 4 % compared to the shuffled dataset while the longformer-2048 improves by almost 15 %.

5.2.5. Meeting-specific Models

To explore what advantages it might bring to specificly train factCC models on long meeting-style data I took the 1-epoch-checkpoints from the reproduced BERT-512 model and the longformer-2048 model and trained them for 2 epochs on the MediaSum-based training dataset. Their balanced accuracies are shown in Table 5.10. The "MS" suffix stands for MediaSum.

		Balanced Accuracies	
		annotated test set	MediaSum-based testset
Model	BERT-512-MS	59.8 %	90.8 % (81.5 % shuffled)
	longformer-2048-MS	60.6 %	89.7 % (69.6 % shuffled)

Table 5.10.: Balanced accuracy scores of BERT-512-MS and longformer-2048-MS

While both models perform similarly on both datasets, there is a 10 point difference in the improvement over the shuffled version of the MediaSum-based testset. This suggests that the BERT-512-MS model relies on the claim sentence itself to a greater extend than the longformer-2048-MS model does. This again supports the advantage of the longformer model of being able to process a longer input.

Additionally the longformer model's performance on the MAT did not worsen as it did in the case of the BERT model, which supports the better generalizability of the longformer.

5.3. FactualCoCo Variants

To build a long version of factualCoCo I use three different longformer models as scoring models. I use each with different input length limits for comparison. How they perform at evaluating summaries is described in the next section.

5.4. DialogLED vs. LED

5.4.1. Training

I trained a longformer-encoder-decoder (LED) model and a DialogLED model on the QMSum dataset for query-based summarization. Both models were initialized from pretrained checkpoints from hugginface. Because the dataset is not very large I can afford to set the input length limit for both models to 5120 tokens. The training lasts for 10 epochs and the best checkpoints are chosen from the validation dataset.

5.4.2. Evaluation

To gain a full picture I evaluate the summarization models' performance on the test set which contains 279 examples using all previously trained variants of factCC models. I also calculate the ROUGE scores. Since I do not have the capacity to manually label the generated summaries I cannot report the correlation of the factCC models with human judgments. The relevant aspect to look at then is the difference in scores between the two summarization models. The evaluation results are shown in Table 5.11. The scores are calculated as described in section 4.3.

Metric / factCC Model	Reference	LED	DialogLED
ROUGE-1	-	30.515	31.786
ROUGE-2	-	8.620	8.895
ROUGE-L	-	18.440	19.067
official factCC checkpoint	0.421	0.410	0.418
BERT-512	0.406	0.349	0.374
longformer-512	0.787	0.724	0.723
longformer-2048	0.825	0.740	0.747
longformer-2048-MS	0.825	0.758	0.755
BERT-512-MS	0.788	0.755	0.747

Table 5.11.: ROUGE and factCC scores of LED and DialogLED on QMSum

The ROUGE scores are in alignment with what was reported in the DialogLED paper in that they show a slight improvement of DialogLED over LED. The fact that all factCC models give a higher score to the reference summaries than to the ones generated from LED and DialogLED indicates that they do in fact work, because it is to be expected that both LED and DialogLED perform worse than the gold reference. However, none of the factCC models show a significant difference between LED and DialogLED as summarization models.

factualCoCo Model	Reference	LED	DialogLED
LED-512	0.027	-	-
DialogLED-512	0.147	-	-
LED-arxiv-512	0.052	-	-
LED-16384	0.094	0.218	0.170
DialogLED-16384	0.207	0.293	0.316
LED-arxiv-16384	0.075	0.144	0.140

Table 5.12.: factualCoCo scores of LED and DialogLED on QMSum

To evaluate LED and DialogLED with the factualCoCo metric three models were used as scoring models. The first thing to note in Table 5.12 is that all three models give higher scores to the reference summaries in their longer versions (bottom half) than in their shorter versions (top half). This shows that it is advantageous if the scoring model is able to process longer inputs if the inputs exceed 512 tokens.

The ideal scoring model would be a very similar one to the model that generated the summaries. However these two models should not be too similar. The first two factualCoCo scoring models are the LED and DialogLED models that were used for the summarization, just here with a different input length limit of 16384 (top two rows in the bottom half of Table 5.12). These two factualCoCo variants prefer "themselves" as summarization models. This kind of situation should be avoided, so the third factualCoCo variant has an LED model as a scoring model which was fine-tuned for long document summarization with the arxiv [4] dataset. This more neutral factualCoCo variant does again not find a significant difference between LED and DialogLED.

6. Discussion

6.1. Conclusion

I have taken two promising summarization metrics that focus on factual correctness, namely factCC and factualCoCo, and adapted them for long meeting summarization. Then I used these metrics to find out more about the way DialogLED differs from its baseline LED.

In the case of factCC it turns out that using a longformer as the classification model instead of BERT improves the metric performance on the MediaSum-based data, which is meeting-style and more abstractive. On the other hand the BERT version of factCC performs better for more extractive claims. As abstractive summarization hopefully becomes less extractive this is a step in the right direction. The reason for this difference between the two models likely lies in the difference between BERT and RoBERTa, because the longformer is based on RoBERTa.

Both metrics factCC and factualCoCo showed improved performance when being able to process longer input. Therefore it is advantageous to make use of the Longformer in these metrics, when the input sequence would otherwise be truncated earlier.

With factualCoCo it is important to make sure the scoring model is not biased for or against a summarization model that it is evaluating.

Neither factCC nor factualCoCo showed a significant difference in the performance of LED versus DialogLED, although ROUGE scores suggest a slight advantage for DialogLED. This indicates that when looking at factual correctness DialogLED does not improve over LED, while the improvement picked up by the ROUGE scores may only be in fluency.

Although there are currently lots of different options for factual correctness metrics and there is no clear standard metric like ROUGE has been until today, future research into summarization should report results with this kind of metric more often to give a greater picture of the results.

6.2. Further Work

More different datasets could be used to explore which classification models for factCC perform better at which types of data. This thesis only looks at the MediaSum dataset and meeting summarization.

In order to evaluate the preformance of factual correctness metrics better, the creation of a labeled dataset of long source texts with abstractive model generated summaries would be very helpful.

Another approach for a long version of factCC would be to use a standard transformer loke BERT and to classify a claim segmentwise with potentially overlapping segments of

the source text. These classifications could then be combined per an OR operation to get the final classification.

An improvement to the training of summarization models could maybe be made by finding a way to incorporate metrics such as the ones used here into the training process or models themselves.

Bibliography

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer". In: *CoRR* abs/2004.05150 (2020). arXiv: 2004.05150. URL: https://arxiv.org/abs/2004.05150.

[2] Yulong Chen et al. *DialogSum: A Real-Life Scenario Dialogue Summarization Dataset*. 2021. arXiv: 2105.06762 [cs.CL].

[3] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *CoRR* abs/1406.1078 (2014). arXiv: 1406.1078. URL: http://arxiv.org/abs/1406.1078.

[4] Arman Cohan et al. *A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents*. 2018. arXiv: 1804.05685 [cs.CL].

[5] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[6] Weijiang Feng et al. "Audio visual speech recognition with multimodal recurrent neural networks". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 681–688. DOI: 10.1109/IJCNN.2017.7965918.

[7] Bogdan Gliwa et al. "SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization". In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/d19-5409. URL: https://doi.org/10.18653%2Fv1%2Fd19-5409.

[8] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[9] Luyang Huang et al. "Efficient Attentions for Long Document Summarization". In: *CoRR* abs/2104.02112 (2021). arXiv: 2104.02112. URL: https://arxiv.org/abs/2104.02112.

[10] A. Janin et al. "The ICSI meeting corpus". In: May 2003, pp. I–364. ISBN: 0-7803-7663-3. DOI: 10.1109/ICASSP.2003.1198793.

[11] Wojciech Kryscinski et al. "Evaluating the Factual Consistency of Abstractive Text Summarization". In: *CoRR* abs/1910.12840 (2019). arXiv: 1910.12840. URL: http://arxiv.org/abs/1910.12840.

[12] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[13] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[14] Iain Mccowan et al. "The AMI meeting corpus". In: *Int'l. Conf. on Methods and Techniques in Behavioral Research* (Jan. 2005).

[15] Ramesh Nallapati et al. "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 280–290. DOI: 10.18653/v1/K16-1028. URL: https://aclanthology.org/K16-1028.

[16] Kishore Papineni et al. "BLEU: a Method for Automatic Evaluation of Machine Translation". In: (Oct. 2002). DOI: 10.3115/1073083.1073135.

[17] Marius-Constantin Popescu et al. "Multilayer perceptron and neural networks". In: *WSEAS Transactions on Circuits and Systems* 8 (July 2009).

[18] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *CoRR* abs/1910.10683 (2019). arXiv: 1910.10683. URL: http://arxiv.org/abs/1910.10683.

[19] Nima Sadri, Bohan Zhang, and Bihan Liu. "MeetSum: Transforming Meeting Transcript Summarization using Transformers!" In: *CoRR* abs/2108.06310 (2021). arXiv: 2108.06310. URL: https://arxiv.org/abs/2108.06310.

[20] Thomas Scialom et al. "SAFEval: Summarization Asks for Fact-based Evaluation". In: *CoRR* abs/2103.12693 (2021). arXiv: 2103.12693. URL: https://arxiv.org/abs/2103.12693.

[21] Prithviraj Sen et al. "Collective Classification in Network Data". In: *AI Magazine* 29.3 (Sept. 2008), p. 93. DOI: 10.1609/aimag.v29i3.2157. URL: https://ojs.aaai.org/index.php/aimagazine/article/view/2157.

[22] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[23] A. Waibel et al. "Phoneme recognition using time-delay neural networks". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.3 (1989), pp. 328–339. DOI: 10.1109/29.21701.

[24] Yuexiang Xie et al. "Factual Consistency Evaluation for Text Summarization via Counterfactual Estimation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Nov. 2021, pp. 100–110. URL: https://aclanthology.org/2021.findings-emnlp.10.

[25] Ming Zhong et al. "DialogLM: Pre-trained Model for Long Dialogue Understanding and Summarization". In: *CoRR* abs/2109.02492 (2021). arXiv: 2109.02492. URL: https://arxiv.org/abs/2109.02492.

[26] Ming Zhong et al. "QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization". In: *CoRR* abs/2104.05938 (2021). arXiv: 2104.05938. URL: https://arxiv.org/abs/2104.05938.

[27] Chenguang Zhu et al. “End-to-End Abstractive Summarization for Meetings”. In: *CoRR* abs/2004.02016 (2020). arXiv: 2004.02016. URL: https://arxiv.org/abs/2004.02016.

[28] Chenguang Zhu et al. “Enhancing Factual Consistency of Abstractive Summarization”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 718–733. DOI: 10.18653/v1/2021.naacl-main.58. URL: https://aclanthology.org/2021.naacl-main.58.

[29] Chenguang Zhu et al. “MediaSum: A Large-scale Media Interview Dataset for Dialogue Summarization”. In: *CoRR* abs/2103.06410 (2021). arXiv: 2103.06410. URL: https://arxiv.org/abs/2103.06410.

A. Appendix

A.1. Data examples

Claim	Relevant part in source text
"hundreds of migrants on board may have capsized"	"The boat that sank in the Mediterranean over the weekend with hundreds of migrants on board may have capsized after being touched or swamped by a cargo ship that came to its aid"
"fda recommends anyone who has consumed a listeria-laden food should let their physician know."	"Dr. Swartzberg recommends anyone who has consumed a listeria-laden food should let their physician know."

Table A.1.: Negative examples from the factCC manually annotated testset (MAT)