Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MASTER THESIS

Fine-Tuning Pre-Trained Language Models for German Multi-Document Summarization

vorgelegt von

Timo Johner

MIN-Fakultät

Fachbereich Informatik

Studiengang: IT-Management und -Consulting

Matrikelnummer: 7213739

Abgabedatum: 18.12.2020

Erstgutachter: Prof. Dr. Chris Biemann

Zweitgutachter: Dr. Abhik Jana

Abstract

We are facing vast amounts of textual information such as news, social media or emails in our everyday lives. One way to comprehend and compress this flood of information can be through *Automatic Summarization*. Capturing salient details from multiple data sources to produce an abridged version, on the other side, is described as *Multi-Document Summarization* (MDS) and has increasingly become object of research in the field of *Natural Language Processing* (NLP). As with other tasks within NLP, Multi-Document Summarization has strongly benefited from rapid evolutions of neural networks, but also demands for large training corpora as well as computational resources.

Here, recent advances of training *Transformer* models on large datasets followed by fine-tuning the model on specific downstream NLP tasks such as summarization have shown great potential by achieving state-of-the-art results. However, those findings mostly focus on English language, leaving their adaptability and performance on other languages in the uncertain.

Therefore, in this thesis a method for fine-tuning pre-trained language models on the task of Multi-Document Summarization of German textual information is conducted, followed by an in-depth error analysis on the model to be considered. Based on the hypothesis that current pre-trained language models can also perform summarization on languages beyond English, the applied methods show the adaptability on German language in an exemplary fashion and provide a fine-tuned model that incorporates latest developments within the field and achieves state-of-the-art performance. To further gain insights an analysis is carried out that investigates potential shortcomings and errors of the utilized datasets and the applied models.

The fine-tuned model generates coherent and comprehensible summaries from the source documents. In addition, the model outperforms previous approaches to German Multi-Document Summarization. Nevertheless, the conducted error analysis shows erroneous patterns, for example factual errors, that were produced by the model and can be found when applied on English and German datasets.

Zusammenfassung

In unserem alltäglichen Leben sind wir umgeben von textuellen Informationen in Form von Nachrichten, Sozialen Medien oder E-Mails. Eine Möglichkeit, diese Informationsflut zu verarbeiten und zu komprimieren, ist beispielsweise mittels *Automatic Summarization*. Die Erfassung wichtiger Details aus mehreren Datenquellen zur Erstellung einer Zusammenfassung wird hingegen als *Multi-Document Summarization* (MDS) bezeichnet und ist zunehmend Forschungsgegenstand auf dem Gebiet der *Natürlichen Sprachverarbeitung* (engl.: *Natural Language Processing*, NLP). Wie auch in anderen Bereich der NLP hat MDS stark von den rapiden Entwicklungen neuronaler Netze profitiert. Gleichzeitig benötigen diese Modelle aber auch riesige Mengen an Trainingsdaten sowie Rechenkapazitäten.
Mögliche Ansätze von vortrainierten *Transformer*-Modellen und anschließender Anpassung (*fine-tuning*) auf spezifische NLP-Aufgaben wie beispielsweise Textzusammenfassung übertreffen hierbei alle bisherigen Ergebnisse. Allerdings fokussieren sich diese Ansätze meist lediglich auf die englische Sprache und lassen somit ihre Anwendbarkeit und Leistung in anderen Sprachen im Unklaren.
Folglich soll in dieser Thesis eine *fine-tuning*-Methode sowie eine Fehleranalyse dieser vortrainierten Sprachmodelle auf das Problem *Multi-Document Summarization* von deutschen textuellen Informationen durchgeführt werden. Ausgehend von der Hypothese, dass aktuelle vortrainierte Sprachmodelle auch in der Lage sind, Textzusammenfassungen in anderen Sprachen – über Englisch hinaus – zu generieren, sollen die hier angewandten Methoden die Anpassbarkeit der Modelle am Beispiel der deutschen Sprache zeigen. Außerdem soll ein fein abgestimmtes Modell, welches die neuesten Entwicklungen auf dem Gebiet miteinbezieht, bereitgestellt werden. Um abschließend Erkenntnisse über mögliche Mängel oder Fehler zu gewinnen, wird außerdem eine Fehleranalyse durchgeführt.
Das verfeinerte Modell erzeugt aus den Quelldokumenten kohärente und verständliche Zusammenfassungen und übertrifft bisherige Ergebnisse von Multi-Document Summarization in deutscher Sprache. Nichtsdestotrotz zeigt die durchgeführte Analyse Fehlerstrukturen, beispielsweise faktische Fehler, die das Modell, wenn es auf englische oder deutsche Datensätzen angewandt wird, erzeugt.

Acknowledgement

These are extraordinary times as I have never seen my supervisors in person. Nevertheless, I am really thankful for the support and collaboration I experienced throughout this work. I would like to thank Chris for giving me the opportunity to write this thesis and for his valuable feedback. I want to thank Abhik for his clear vision of what lies ahead and his inspiring thoughts. I really enjoyed working with you.

As this work also marks the end of my academic studies, I would like to thank my friends and flatmates for their support, new food for thoughts, controversial discussions and having a good time.

I also want to express my gratitude towards my mother, father and sister for their tremendous support beyond all measures, their belief in me and for always being there for me.

Lastly, I would like to put forward my gratitude and love towards my girlfriend Inga for sparking new ideas, motivating me when in doubt and always being by my side.

Contents

1. Introduction

Within Natural Language Processing (NLP) the task of *Multi-Document Summarization* can be described as the procedure of representing a set of multiple documents about the same topic in a short and concise version capturing relevant information while filtering out redundant and unnecessary information. This thesis provides a solution for MDS on German textual information and points out erroneous cases within the utilized datasets and applied model.

1.1. Motivation

You have been kindly asked to read this document. As you read through it, your brain makes notes, creates links between words, sentences – and even embeds it in a wider context. Time and space is limited so when someone would ask you about the document you would present him or her only the important information of this document – a summary.
The automation of this task of *Document Summarization* is extremely useful as we are facing vast amounts of information such as news, social media or emails in our everyday lives. This information is cognitively overwhelming and outgrows what we as humans are capable of processing. Complexity even rises with multiple documents, demanding to cope with different opinions and perspectives while aiming for concision, readability and completeness.
With the application of deep neural networks on NLP tasks there has been huge progress in summarizing documents. But obstacles still remain such as sparsity of large datasets or the demand for high computational resources. Consequences are poor performance due of overfitting to training data and failed generalization. This is especially the case for datasets within the domain of summarization and further increases on the subtask of Multi-Document Summarization as the creation of those datasets is of cognitive and time-consuming nature. The situation for summarization of German textual information is even more critical due to an intensified lack of summarization datasets.
Here, *Transfer Learning* is a means that allows to pretrain representations on large unlabelled text corpora and to transfer the learnings to a target task by using fewer

examples of annotated data. This approach has been adapted to the task of summarization showing state-of-the-art results and mitigated the need for large datasets. Nevertheless, until now, its adaptability to the task of Multi-Document Summarization on German textual information has not been explored.
For these reasons, this work seeks to explore and adapt the latest developments of Multi-Document Summarization on German language.

1.2. Research Questions

Research in the domain of Natural Language Processing focuses predominately on developing methods that work well for English, while their applicability on other languages is under-researched (Ruder, 2020). It would be valuable to know whether these models are also capable of being adapted to other languages such as German, how they perform and what their potential shortcomings are. Here the question is how to apply pretrained language models on summarization and fine-tune them on German datasets for multi-document summarization. The first goal of this thesis is to apply state-of-the-art language models for summarization and create a competitive fine-tuned model for German multi-document summarization that also improves the results of previous summarization models for German language.
Furthermore, there is little knowledge about potential gaps and errors of these models e.g. regarding the capture of semantics or long-term contexts, whether for English or for trans-lingual approaches, leaving progress of these models beyond common evaluation metrics in question. The second goal therefore is to explore potential errors and shortcomings of the applied model for English and German language.
Derived from these statements, the research questions for this master thesis are the following:

1. *Are pre-trained language models also applicable to other languages?*
2. *How are gaps and potential errors in language models structured and do they apply across languages?*
3. *What are general shortcomings derived from recognized erroneous patterns of language models?*

In this work, multi-document summarization is viewed as a task of merging multi-input to single-input followed by summarization. Given the fact that the model does not consider hierarchical structure within the source documents, this work might be also applicable for single-document summarization.

The scope mainly describes fine-tuning a model for multi-document summarization regarding the auto-hMDS dataset (Zopf, 2018), but arbitrary single- and multi-document summarization datasets might be utilized for comparison.

1.3. Overview

Chapter 2 embeds this thesis in a theoretical background that provides detailed information about the main concepts that will be applied. This includes key concepts of summarization in general, kinds of neural networks, language models as well as the here applied architecture patterns of Encoder-Decoder models and Transfer Learning. This is followed by related work in Chapter 3, that summarizes past and recent approaches and methods about summarization and their variations. The three datasets that were used throughout this thesis are introduced in Chapter 4. Chapter 5 describes the methodology for the following experiments. The conducted experiments on the three datasets as well as their evaluation and a comparison to previous models is outlined in Chapter 6. In order to further investigate the results regarding their quality, an analysis is carried out in Chapter 7. Lastly, Chapter 8 concludes with summarizing the findings and an outlook for future work.

2. Theoretical Background

This chapter describes the main concepts and theoretical foundation of this thesis. The first section determines the summarization task as well as its subcategories and how to quantitatively measure summarizaton systems with the ROUGE (Lin, 2004) score metric. Then the concept of neural networks and their evolution, namely *Feedforward Neural Networks* (FFNNs) and *Recurrent Neural Networks* (RNNs), is outlined as well as language models are introduced. Finally, the architectural components that are particularly relevant for this thesis are described – namely the *encoder-decoder* architecture and *transfer learning*.

2.1. Text Summarization

Text Summarization refers to the process of condensing long pieces of text into a shorter version. The intention is to create a coherent and fluent summary that outlines only key informational elements while preserving the meaning of the content. Text summarization is a task within the field of Natural Language Processing that tries to automate this process computationally. This becomes more and more important as the volume of information drastically increases and is already too large for humans to process manually. Text summarization can, for example, help to reduce reading time or enable faster skimming for relevant information. Possible applications for text summarization are for example news, scientifc literature, meetings, mails and books (Torres-Moreno, 2014, pp. 3-6).
Early work on summarization was based on *term-frequency* to automatically extract frequently occurring textual information, first from words and then from sentences, in order to use this information to generate abstracts (Luhn, 1958). Later, positional features of sentences and words within the text were incorporated (Edmundson, 1969). In recent years, machine learning methods have been adapted to summarization.
Approaches to text summarization within NLP can be based on statistical, graph-based, linguistic, machine learning and compression methods (Aries et al., 2019). Furthermore, the task of text summarization in NLP adapts several techniques from other NLP tasks such as text extraction, text classification or text generation.

The following steps are usually applied in the pipeline to perform text summarization on raw data: Read input document, select information from the document(s) that are of relevance, revision operations based on the input, improve the fluency of the generated text and output the final summary. The revision operation that is commonly used to transform articles into a summary can be described as following (Jing, 2002):

- Sentence reduction
- Sentence combination
- Syntactic transformation
- Lexical paraphrasing
- Generalization or specification
- Reordering

Contextual distinctions can be drawn between whether generic or query data is required to create a summary. *Query-focused* summarization systems take into account the input e.g. a question and produce a summary based on this input while *generic* summarization systems produce a general summarization of the source document. (Torres-Moreno, 2014, p. 12)
Concerning the output, two different approaches to generate summaries will be described in the following subsections. These approaches are also exemplified in Table 2.2, where the abstractive summary generates text with the same context based on the source text, while the extractive summary only copies fragments from the source text.

2.1.1. Extractive Summarization

Extractive Summarization systems identify the most important extracted fragments within a source and create the summary by concatenating these fragments. Extractive summarization is often defined as a sequence labeling task that generates a summary by deciding whether a sentence from the source should be included in the summary or not. The decision can be based on linguistic features that expose characteristics such as occurences of words, semantic relations or rhetorical structures of the text. Modern extractive systems include approaches that put the extractive fragments, such as sentences, phrases and words, in relation to fragments that have already been included and derive the meaning representation from the source to rank the fragments

in a coherent and fluent order. (Torres-Moreno, 2014, pp. 30-35)

2.1.2. Abstractive Summarization

Abstractive Summarization systems try to understand the semantics and context of the source input and generate the summary through text manipulation such as generation, substitution, reordering and deletion of text fragments to create novel sentences. Compared to extractive summarization, the task of abstractive summarization is often more complex and computational costs tend to be more expensive, because the generation of text that is coherent and grammatically correct requires a deep understanding of the domain language and the contextual information. On the other hand, abstractive summarization systems promise a better quality in terms of content and linguistic features as well as a more human-like summary (Torres-Moreno, 2014, pp. 219-220). Recent summarization models often combine extractive and abstractive summarization and use extractive summarization as an intermediate step followed by enriching the process with abstractive methods.

Original Text	
Alice and Bob went for a hike in the mountains. They enjoyed the diverse landscape and the waterfalls and even saw moose and one bear. Along the riverside they set up their tent and cooked some pasta. On the next day they got up early and climbed on top of the highest mountain where they had a spectacular view over the countryside. Fortunately the weather was very good without any clouds or rain.	
Extractive Summary	**Abstractive Summary**
Alice and Bob went for a hike. They enjoyed the landscape and saw moose. They set up their tent along the riverside and cooked pasta. Next day they climbed on the mountain where they had a spectacular view over the countryside. The weather was very good.	Alice and Bob went hiking in the mountains. They enjoyed it and even saw animals. They slept by the riverside and had pasta for dinner. On the second day they climbed the highest mountain and enjoyed the view over the countryside. Luckily the weather was very good.

Table 2.2.: Own example with the original text and exemplary extractive and abstractive summaries. The orange-marked texts highlights changes that were made in order to create a shorter and concise version.

2.2. Multi-Document Summarization (MDS)

Multi-Document Summarization is a subtask of text summarization and aims to produce an abbreviated version from multiple source documents, usually about the same topic. While a lot of the previous introduced methods are also relevant for multi-document summarization, there are new challenges that arise with summarizing multiple documents. The main challenge is particularly to create a summary which covers information about a topic comprehensively while avoiding redundant or even contradictory information in the results (Torres-Moreno, 2014, pp. 109-110). Additionally the MDS systems need to filter out the informational noise from the respective documents that do not provide any information about the topic in question. The balancing of what is relevant for a summary and what can be disregarded has strong impact on the structure and design of the summary.
Within MDS this can be done through *content selection*, *information ordering* and *sentence realization* (Jurafsky and Martin, 2009, pp. 810-815). Approaches to these challenges are for example redundancy detection through calculating the similarity between two sentences with Maximal Marginal Relevance (MMR) where sentences that are too familiar will be penalized (Carbonell and Goldstein, 1998). Another approach is to cluster related sentences and allow only one sentence from each cluster for the final summary (Witte and Bergler, 2007).

2.3. Summarization Evaluation

With *Automatic Text Summarization* there is also a need to quantify the quality of the machine producing output summaries. One method of measuring output summaries is to involve people who evaluate the generated summaries. If the inclusion of persons is not possible, the generated summary is often compared and evaluated with its human counterpart.
For evaluating the performance extrinsic and intrinsic methods can be applied (Mani and Maybury, 2001). Extrinsic evaluation determines the performance of the summary on other tasks. For example, how well does the summary provide information on a particular question? Intrinsic evaluation measures the informativeness of the generated summary by comparing it with the human-made summary. Early approaches to evaluating the informativeness are *Relative Utility* (Radev and Tam, 2003), *Factoid Score* (Teufel and van Halteren, 2004) and *Pyramid Method* (Nenkova and Passonneau, 2004) as well as ROUGE (Lin, 2004).
Recall-Oriented Understudy of Gisting Evaluation or ROUGE defines a set of met-

rics that "count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans" (Lin, 2004). ROUGE includes five measures: ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU. Most importantly here are ROUGE-N and ROUGE-L.
The recall-based ROUGE metric tries to evaluate the similarities between the reference summary and the generated summary with the assumption that the fluency can be measured by same word order. ROUGE-N measures the n-gram units that are in overlap between the generated summary and reference summary where n is defined as length of n-gram. For example ROUGE-1 refers to the overlap of uni-gram (each word) and ROUGE-2 refers to bi-grams. ROUGE-N is calculated as followed:

$$= \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \tag{2.3.1}$$

$gram_n$ and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occuring in a generated summary and reference summary.

ROUGE-L measures the Longest Common Subsequence (LCS) within two summaries with the assumption that a long common subsequence also determines the similarity between the two summaries. ROUGE-L is calculated as followed:

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{2.3.2}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{2.3.3}$$

$$\text{ROUGE-L} = F_{lcs} = \frac{(1+\beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{2.3.4}$$

Here X and Y represents the summaries with length m and n. $LCS(X, Y)$ stands for the longest common subsequence of the two summaries and $\beta = \frac{P_{lcs}}{R_{lcs}}$ when $\frac{F_{lcs}}{R_{lcs}} = \frac{F_{lcs}}{P_{lcs}}$. The advantage of $\mathrm{ROUGE\text{-}L}$ is that it already includes the longest in-sequence common n-grams. (Lin, 2004)

While the ROUGE method has gained large popularity within the domain of text summarization, there is also criticism regarding ROUGE as it does not consider synonymous concepts nor measure the content coverage from both the reference and generated summary (Ganesan, 2018). Despite this, ROUGE still remains the default evaluation metric regarding text summarization, since it provides an automatic and effective way to measure the performance (Fabbri et al., 2020).

2.4. Artificial Neural Networks

An *artificial neural network* (ANN), or often simply called *neural network* (NN), is a computational model whose structure and functions are inspired by the human brain. McCulloch and Pitts (1943) introduced first work of that kind, including models of neurological networks and recreating threshold switches based on neurons. The architecture of an ANN consists of neurons in different layers and connections between them which hold a certain weight. The information that is processed by the network changes its input and output and in that sense the network is adapting or "learning" while processing information. The upcoming sections describe the components of neural networks, followed by the concepts of feedforward and recurrent neural as well as the architecture of long short-term memory and a description of the learning procedure of artificial neural networks.

2.4.1. Basic Principles

Neuron

Neurons can be described as simple processing units within neural networks (McCulloch and Pitts, 1943). The neurons are intertwined over directed, weighted connections and can get activated through direct input or other neurons. The weighted connections provide an excitatory or inhibitory feature that can individually be adjusted and allows the network to "learn" as it evolves. The weights will be summed up to a weighted sum and passed through a non-linear function called activation function to generate the output.

Perceptron

The *Perceptron* is a binary classifier in supervised learning and was described by Rosenblatt in 1958. In his work, Rosenblatt initially defined the weighted sum and activation function as building blocks of the perceptron. Figure 2.1 depicts the structure of a single-layer perceptron as a simple neural network with three binary inputs x_1, x_2 and x_3, three weighted connections w_1, w_2 and w_3, and one binary output y. The binary output y is calculated through the activation function f whether the weighted sum $\sum$ is lower or higher than a defined threshold value. This indicates if the neuron becomes active or not. The bias b has a weight w_b itself, which will also be added to the weighted sum.
Here the perceptron's output can be calculated by $y = f((\sum w \cdot x) + b \cdot w_b) = f((w_1x_1 + w_2x_2 + w_3 + x_3) + b \cdot w_b)$.

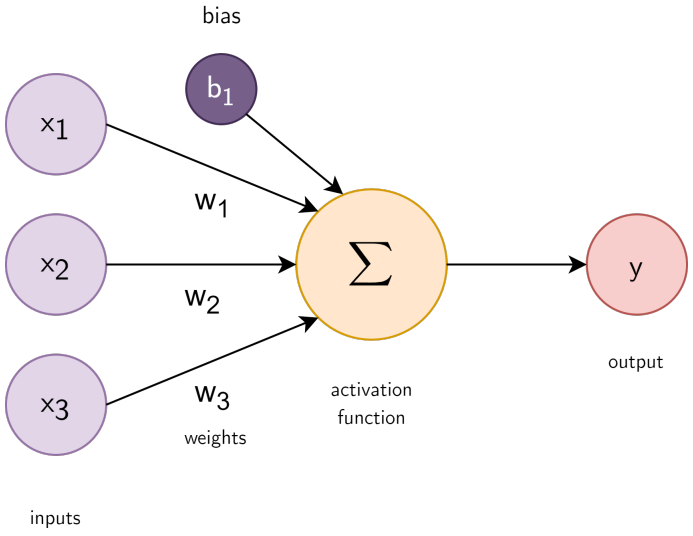

Figure 2.1.: Structure of a single-layer perceptron with three inputs, their according weights and a bias. Adopted from Minsky and Papert (1969).

Activation

The *activation* describes the switching status within a neuron. Close to the threshold value, the activation function of a neuron reacts particularly sensitive and determines the activation of a neuron dependent on the input and threshold value. Figure 2.2 shows the most common activation functions. The activation function that is used in the perceptron is also the simplest activation function called binary *threshold function* or *heaviside step function* (Equation 2.4.1). This function can only take two values, 0 or 1, but otherwise remains constant.
Another popular activation function is the *logistic function* or *sigmoid function* (Equation 2.4.2) which maps the range of values from 0 to 1 and the *hyberbolic tangent function* (Equation 2.4.3) which maps to values from -1 to 1. Compared to the binary threshold function the logistic function and the hyberbolic tangent are differentiable. Lastly the *Rectified Linear Unit* (ReLU) maps values from 0 to ∞ and is the most common activation function for neural networks and deep learning models today (Equation 2.4.4). (Jurafsky and Martin, 2019, Chapter 7)

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases} \tag{2.4.1}$$

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.4.2}$$

$$f(x) = tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.4.3}$$

$$f(x) = max(0, x) \tag{2.4.4}$$

2.4.2. Feedforward Neural Networks

The extension of the previously described *single-layer perceptron* is the *multi-layer perceptron* (MLP) as a class of *Feedforward Neural Networks* (FFNNs). MLPs define a neural network that consists of one input layer, one output layer and one or more processing layers invisible from the outside (hidden layers). Each neuron in one layer has

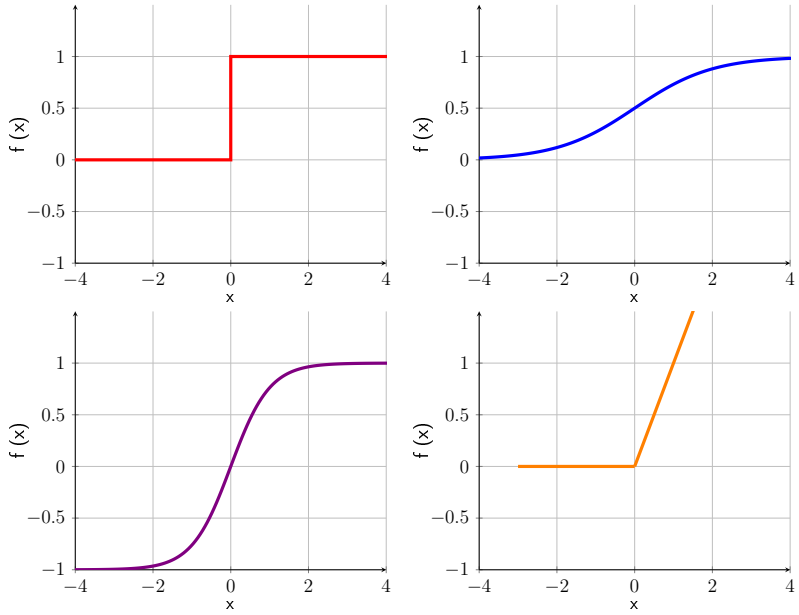

Figure 2.2.: Popular activation functions: binary threshold function (red), logistic function (blue), hyberbolic tangent (violet) and ReLU (orange).

weighted directed connections to the subsequent layer and can not be fed back. Compared to single-layer perceptrons, MLP are capable of learning to compute non-linear functions, which are essential for regression and classification in supervised learning. Figure 2.3 shows a feedforward neural network with two layers, three inputs and two outputs. (Jurafsky and Martin, 2019, Chapter 7)

2.4.3. Recurrent Neural Networks

Compared to FFNNs, *Recurrent Neural Networks* (RNNs) are capable of providing their neurons with a recurrent connection to the internal state (memory). After the output of a neuron is produced, the state is kept and sent back to the recurrent network. By this the recurrent network can consider the current input and the output that it has learned in order to make a decision. This makes it possible for RNNs to process input of variable length and makes them especially eligible for sequence processing. (Goodfellow et al., 2016, pp. 372-376)
Figure 2.4 shows a recurrent network with one hidden layer. Within the hidden layer the neurons have direct recurrences (self-recurrence) that start and end at the same neuron. The activation value of the hidden layer depends directly on the input as

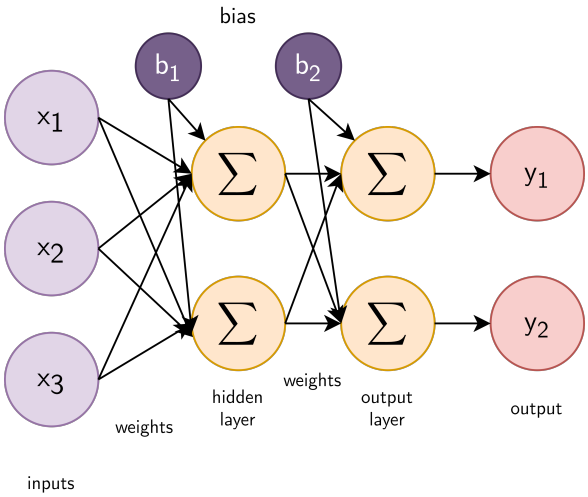

Figure 2.3.: Structure of a two-layer feedforward neural network with three inputs, two outputs, two biases and one hidden layer. Figure according to (Jurafsky and Martin, 2019, Chapter 7).

well as on the activation value of the hidden layer from the previous step. Between the neurons in one layer there are also lateral recurrences in which a neuron output is connected to other neurons within the layer.

The classic feedforward neural network architecture consists of one input and one expected output. Several RNN architectures exist that differ on the number of inputs and outputs. The *one-to-many* architecture takes a fixed size input and creates an output that can be of variable length. An example for *one-to-many* architectures can be image captioning where the image represents a fixed size input and the created output can be of words or sentences. The *many-to-one* architecture represents the opposite where the input is of variable length and the output has a fixed length, e.g. sentiment classification where the input is a sentence and the output is a continuous value that represents the likelihood of having a negative or positive sentiment. Finally, the *many-to-many* architecture allows input and output of variable length and can be subdivided into two types wherein the first type, the input length equals the output length and in the second type, the input length is unequal to the output length. An example for the first type would be entity recognition. Per definition summarization systems process an input into a shorter concise output, which means that this type will be a necessary building block throughout the thesis. (Jurafsky and Martin, 2019, Chapter 7)

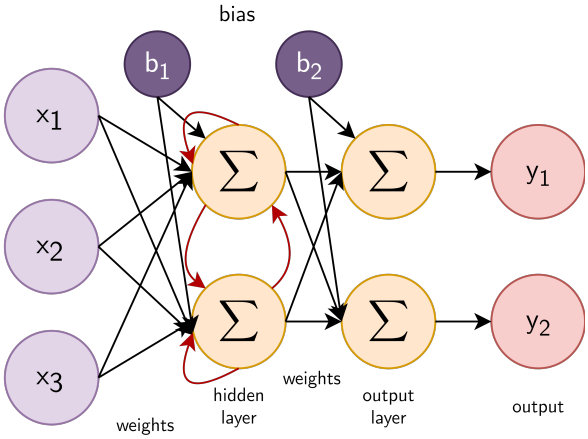

Figure 2.4.: Structure of a two-layer recurrent neural network with three inputs, two outputs, two biases and one hidden layer with self and lateral recurrent connections marked in red. Figure according to Rumelhart et al. (1985).

One shortcoming of the previously discussed RNN architectures is that they are only capable of capturing forward dependencies without the possibility of looking at previous states. Another problem when training large RNNs is that of *vanishing* or *exploding gradients* (Bengio et al., 1994). Due to the backward pass of training the hidden layers, they are subject to repeated multiplications, determined by the length of the sequence, where the gradient eventually vanishes or explodes if its below or above the value 1. Vanishing gradients describe the problem of exponentially decreasing gradients as propagating through the model because of small derivatives until the gradients fully vanish. In the case of exploding gradients the derivatives are large enough to grow the gradient exponentially until the gradients explode. This makes it more difficult to train RNNs for solving problems that require learning long-term temporal dependencies and results in ineffective learning or even the abortion of the training procedure, because weights and biases will not be updated properly.

2.4.4. Long Short-Term Memory

In order to solve the shortcomings of learning long-term dependencies, the *long-short term memory* (LSTM), introduced by (Hochreiter and Schmidhuber, 1997) and then further refined, was proposed. LSTMs change the design of previous RNNs by adding an additional cell state that remembers previous dependencies and can be manipulated

by structures called gates. The value of the cell state is always between 0 and 1 and therefore mitigates the previously described problem of vanishing or exploding gradients.
The LSTM architecture consists of three gates that control the propagation of the gradient and therefore the flow of information. The forget gate controls which information to throw away and which to keep. Information from the previous hidden state and the current input is passed through a sigmoid function with values between 0 and 1. The input gate passes the current input and the previous hidden state into a sigmoid function and into a tanh function. Afterwards the outputs from both functions get multiplied. This output and the output of the forget gate then form the cell state with a pointwise multiplication of the cell state with the forget vector and a pointwise addition of the output from the input gate that updates the new cell state. Lastly, the output gate defines the next hidden state by passing the previous state and current input into a sigmoid function and a tanh function. The result from both functions will then be multiplied and the hidden state will be determined. (Jurafsky and Martin, 2019, Section 9.4.1)
Variation of the LSTM are, for example, LSTMs with an additional *peephole connection* by (Gers and Schmidhuber, 2000) or the *Recurrent Units* (GRUs), introduced by (Cho et al., 2014). The peephole connection allows the gates of the LSTM to view the cell state in order to better measure time intervals. GRUs combine the input and forget gate to an update gate and remove the output gate therefore relying on fewer parameters which results in faster training and less needed data. (Greff et al., 2016)
LSTMs are successfully applied to a wide area of sequence processing tasks such as speech recognition (Fernández et al., 2007), language translation (Sutskever et al., 2014) or text summarization (Rush et al., 2015).

2.4.5. Training

Neural Networks are capable of familiarizing with problems through training and can derive paradigms as well as apply their learnings on previously unknown problems. This method is referred to as *generalization*. In theory, a neural network can learn through changing its connections, weights, thresholds or through adding or removing neurons. In practice, learning is mostly done through adapting weights.

Training FFNNs

The training of *Feedforward Neural Networks* includes choosing an optimizer, a cost function and the form of the output units. As FFNNs consist of hidden layers, an

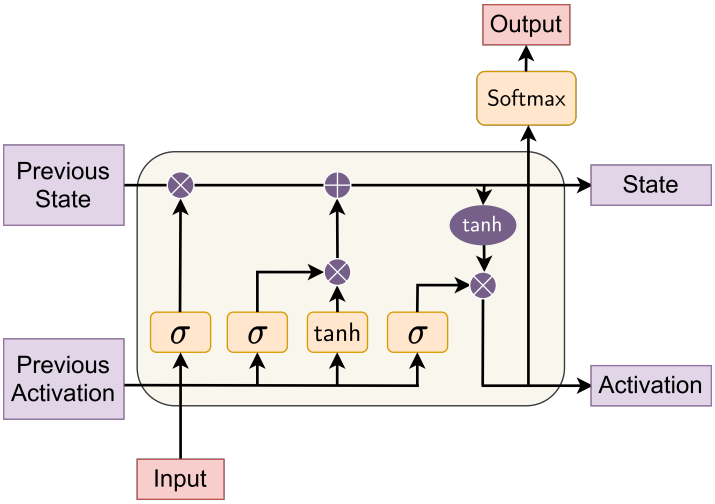

Figure 2.5.: Structure of a single LSTM cell according to (Hochreiter and Schmidhuber, 1997). The top line illustrates the cell state that is updated through the gates that are arranged between the top and bottom line.

activation function is required that computes the output values of these hidden layers. While for single-layer neural networks the derivative of the loss function can directly be computed, multi-layer neural networks with multi-dimensional weights require a more complex approach. The gradient can be computed via gradient-based learning and backpropagation (Rumelhart et al., 1986). Backpropagation provides the neural network with the ability to minimize the cost function by adjusting the weighted connections in the network and therefore minimizing the difference between the actual and the desired output vector. For this, the backpropagation algorithm makes use of the chain rule and propagates the error from the output back to the first hidden layer stepwise with respect to the previous input variables. (Goodfellow et al., 2016, pp.167-173).

Training RNNs

Similar to FFNNs the *Recurrent Neural Networks* take a training set, a loss function and the previously introduced backpropagation to obtain the gradients needed for changing the weights. Additionally, with RNNs the hidden layer to be considered needs to know the current output as well as the output of the followed layer to assess the error. This can be done with the *Backpropagation Through Time* (BTT) approach (Werbos (1974); Rumelhart et al. (1986); Werbos (1990)). In a first step, the hidden layer and

its output are computed at a certain time while accumulating the loss of each step in time and saving the value of the hidden layer at each step for the next step in time. At the second step, the sequence will be processed in reverse by computing and saving each error term in a hidden layer backwards. As this task is done sequentially and the computation graph requires all intermediate hidden states in order to learn the weights, this process can be very expensive and can further lead to *vanishing gradients* or *exploding gradients* as described in Subsection 2.4.3 (Bengio et al., 1994). (Jurafsky and Martin, 2019, Chapter 9)

2.5. Language Models

Language models aim to assign the probability of a sequence $P(w_1, ..., w_n)$ given a sequence of words with the length n. Language models can be categorized in N-gram language models and neural language models.

2.5.1. N-gram Language Models

N-gram models compute the probability $P(w|h)$ of a word w over some history h. n-grams are sequences of words where n defines the number of words (Jurafsky and Martin, 2019, Section 3.1). A bi-gram therefore determines a sequence consisting of two words e.g. "good day". Instead of taking all previous words into account, N-gram models approximate the history only using a defined number of previous words. Now to compute the joint probability of a sequence, the chain rule of probability uses the conditional probability of a word and approximates the history by using the last $k-1$ words:

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)...P(w_n|w_1^{n-1}) \approx \prod_{k=1}^{n} P(w_k|w_{k-1}) \qquad (2.5.1)$$

As the assumption is that the probability of a future unit can be predicted without considering not all but only some previous units, this approach is called a *Markov assumption*. This n-gram probability can be estimated with the so called *maximum likelihood estimation* (MLE) that counts the occurrence for a word in a given sequence by dividing the observed frequency of a particular sequence with the observed frequency of the prefix (Equation 2.4.1).

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})} \tag{2.5.2}$$

The following is an example for calculating bi-gram probabilities of a corpus that has two sentences:

<s> What a good day </s>
<s> What a good life </s>

Firstly a symbol is defined e.g. <s> that marks the beginning of each sentence and one symbol is defined for the end of each sentence </s>. Then, the calculation for each fragment is performed e.g. $P(\texttt{good|a}) = \frac{2}{2}$ or $P(\texttt{day|good}) = \frac{1}{2}$.

One shortcoming of n-gram language models is that in practice known words might appear in a context that was not included in the training data. If a word did not appear before in this sequence, the language model will assign zero probability. Therefore a more sophisticated way to estimate the probability of words in n-gram language models are smoothing algorithms such as *Laplace* or *Kneser-Ney*. The Laplace algorithm simply adds one to each count and thus gets rid of the potential zeros in n-gram probability. The interpolated Kneser-Ney algorithm (Kneser and Ney, 1995; Chen and Goodman, 1996) is most commonly used for n-gram smoothing and interpolates probabilities of n-grams e.g. if the language model estimates the probability for a tri-gram ('What', 'a', 'good') it will interpolate lower order grams e.g. the bi-gram ('What', 'a') to approximate the probability of an n-gram.

The major problem of n-gram models is that the number of parameters increase exponentially while there is no possibility to generalize from training to test set. These drawbacks can be solved with neural language models that can also take into account words with similar contexts that have similar representations.

2.5.2. Neural Language Models

The previous introduced foundation on neural networks and language models can be combined in neural language models that use feedforward neural networks or more commonly recurrent neural networks as recurrent neural language models. While N-gram language models and feedforward neural networks compute the probability of the next word in a sequence by previous words of a defined history, recurrent neural

language models use the current word and a hidden state that provides information on preceding words of the sequence through vector representations called word embeddings (described in Equation 2.5.3) (Mikolov et al., 2010).
As recurrent neural language models are capable of processing any length of input and can use information from many steps back, they are able to capture semantic similarities much better than ordinary language models. Forward inference in recurrent language models proceeds by retrieving word embeddings from the current word as an input and combine it with the hidden layer from the previous step. This representation is then passed through a softmax layer that generates a probability distribution over the entire vocabulary (Jurafsky and Martin, 2019, Chapter 9). Here again the chain rule of probability is applied to calculate the probability of an entire sequence with the output y (Equation 2.5.3).

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_1^{k-1}) = \prod_{k=1}^{n} y_k \tag{2.5.3}$$

In order to benefit from the feature of using preceding words, the neural language model needs training data in form of representative text. The network then uses cross-entropy as the loss function to predict the next word.
With a trained model, the network can create novel sequences by randomly sampling a word for the beginning of a sequence and then continue generating words using the hidden state from the previous step and the word embedding for the current word. This technique is described as autoregressive generation and is also key fundamental of state-of-the-art architecture for tasks such as machine translation, question answering and more interesting text summarization. Within text summarization, the neural language model is not constructing a sequence from scratch but uses a specified input to execute the subsequent autoregressive generation that constructs the sequence from the probability of every word of the vocabulary and a given context.

2.5.3. Word Embeddings

Word embeddings are based on the semantic theory of *distributional hypothesis* that can be summarized under the assumption that words that appear in the same context tend to have similar meanings (Harris, 1954). Word embeddings map words or phrases to vector representations where words are represented as real-valued vectors in a defined vector space. Due to its numerical nature it is possible to perform mathematical

operations on these vector representations. Semantic similarities between words can be computed by calculating the distance between vector representations. The real-valued vectors can be learned in a way that resembles the structure of neural networks where distributed representations of words are learned based on their usage (Bengio et al., 2003). An efficient approach of mapping word embeddings from a text corpus with neural networks was introduced by (Mikolov et al., 2013a,b) with the *Word2Vec* algorithm that introduced the learning models Continuous Bag-of-Words (CBOW) and Skip-Gram in order to learn word embeddings that can differ in their complexity and size.
Another popular embedding model that learns in a similar but count-based way is the *GlobalVectors* (*GloVe*) algorithm (Pennington et al., 2014). Rather than predicting words, *GloVe* stores the occurrence of words in a matrix where the words are stored in rows while the columns define the context. This matrix is then factorized to represent a lower-dimensional matrix of features and words.
In 2016 *fasttext* was introduced (Bojanowski et al., 2017; Joulin et al., 2017), which relies on the method of skipgram from *Word2Vec* but improves its performance by using subword information, which can be defined as character-level n-grams, in order to construct embeddings for words as well as form representations of *out-of-vocabulary* words (OOV).
These previously described word embedding techniques are capable of mapping a word to a continuous vector space and creating an adequate representation. However, they are lacking the capability to capture the dependencies of words regarding their complex characteristics of use and how these uses vary across linguistic contexts when a word or phrase can have multiple meanings. Approaches to mitigate this problem were introduced with deep contextualised word representation models such as *Embeddings from Language Models* (ELMo) by (Peters et al., 2018) or *Bidiretional Encoder Representations from Transformers* (BERT) by (Devlin et al., 2019) based on the architecture of encoder-decoder models.

2.6. Encoder-Decoder

The Encoder-Decoder model based on recurrent neural networks was first introduced by Cho et al. (2014) and Sutskever et al. (2014). The architecture utilizes recurrent neural networks for sequence-to-sequence prediction with the benefit of one end-to-end model that can be trained on source and target sentences. The input and output is capable of handling sentences of variable length and the generated output is a contextualized representation of the input. The architecture is made of three components: an encoder,

a decoder and a context vector. The encoder can map an input sequence $(x_1, ..., x_n)$ to a sequence of continuous representations $z = (z_1, ..., z_n)$ of the context vector that will be utilized by the decoder to generate an output sequence $(y_1, ..., y_m)$ one element at a time.

Encoder

The encoder component accepts an input sentence of variable length and encodes this sequence to a corresponding sequence of contextualized representation that will be handed over to the context vector. Previously described modules such as RNNs, LSTMs and GRUs can all be applied as encoders.

Decoder

The decoder utilizes the previously discussed autoregressive generation to build up the output sequence from the context vector. As discussed before, the output sequence is constructed by looking at the previous hidden state and the output of that state. With this approach, the context vector is only used at the beginning of each sequence with decreasing influence as the sequence modelling continues. This can be solved with the context vector as a parameter for calculating each current hidden state. Another possibility is to condition the output to the generated hidden state as well as to the output generated at the previous state and the context vector. As with encoders, modules like RNNs, LSTMs and GRUs can be employed as decoders.

Context vector

The context vector can be defined as the function of a hidden state of an encoder. Due to the fact that this hidden state is dependent upon the input size, only the final hidden state is used, reducing the context vector to a fixed length. One shortcoming of this approach is that this reduced context vector provides contextual information that are more focused on previously added information than on the sequence as a whole. Approaches to mitigate these problems are Bi-RNNs as well as calculating the sum or average of all hidden states to produce the fixed length context vector. Thus, these methods remove insightful information on each individual state. This lack can be mitigated through attention models.

2.6.1. Attention

The attention mechanism was firstly presented by Bahdanau et al. (2015) with an enhanced context vector that includes annotations from the whole input supplemented with a focus on the surroundings of a current word. The proposed *alignment model* compares the input at position j with the previous output at position i and provides information about which parts of the source sentences the model should pay attention to. The similarity of relevance between the decoder hidden state and the encoder hidden state is calculated by using the dot product between vectors (Equation 2.6.1):

$$score(h^d_{i-1}, h^e_j) = h^d_{i-1} \cdot h^e_j \tag{2.6.1}$$

To further parametrize the score with its own set of weights, the variable W_s is introduced which provides the model with the ability to decide which information between the decoder and encoder states are interesting (Equation 2.6.2):

$$score(h^d_{i-1}, h^e_j) = h^d_{t-1} W_s h^e_j \tag{2.6.2}$$

This score will then be normalized with a softmax function to provide the relevance of the encoder hidden state j to the current decoder state i as the distribution α_{ij}. The weighted average over all encoder hidden states is then computed to the final fixed-length context vector (Equation 2.6.3):

$$c_i = \sum_j \alpha_{ij} h^e_j \tag{2.6.3}$$

While this attention mechanism is able to produce a separate context vector for every output step, this approach still requires sequential processing of the tasks, which demands a lot of computation.

2.6.2. Transformer

The *Transformer* model, introduced by Vaswani et al. (2017), follows the encoder-decoder architecture with the idea of replacing recursive or convolutional layers with *Self-Attention Layers* and allowing less sequential and more parallel for text generation.
The standard Transformer model consists of an encoder and decoder each with six identical layers and an input word embedding in the bottom-most encoder. Each encoder layer includes a multi-head self-attention mechanism and a feedforward network. The multi-head self-attention layer checks for other words in an input sequence while encoding a word and passes its output to the feedforward network. For this, it firstly creates weighted matrices from each input matrix named query matrix q_n, key matrix k_n and value matrix v_n. The values q and k will be used to calculate the softmax score which will then be multiplied with the value matrix v to the final resulting z matrices. These z matrices will then be concatenated and redirected to the feedforward network. Instead of sequentially working through the words, the feedforward network is able to execute words in parallel. Each feedforward network consists of two linear transformations with a ReLU activation in between. The decoder consists of the same layers but employs an additional attention layer as intermediary. This attention layer helps the decoder to pay attention to the input sentence while further processing. The encoder constructs the sequence by aggregating information from all other words, generating a new representation for each word using the entire context.
Popular models that make use of the transformer architecture are for example BERT (Devlin et al., 2019), $\mathrm{GPT\text{-}2}$ (Radford et al., 2019) or BART (Lewis et al., 2020). They have in common not only the use of the tranformer architecture, but also the utilization of transfer learning as a method to enhance *Natural Language Processing* models.

2.7. Transfer Learning

These previously discussed neural methods have shown great success in NLP and beyond. But they are also strongly dependent on large training data and so far lack the ability to generalize to conditions and problems beyond the ones the model encountered during training.
One approach to mitigate these problems is for example *Transfer Learning* where the objectives are to learn from a source domain D_s and its associated task T_s with transferring the gained knowledge to a target domain D_t with a target task T_t (Pan and Yang, 2010).

Transfer Learning can be classified into three dimensions: first, whether the source and target domain settings are different e.g. regarding their language; second, regarding the tasks and topics they are dealing with and third whether they are learning tasks sequentially or in a multi-task approach. Applying sequential transfer learning by providing pre-trained language models has increasingly improved performance of language modelling (Ramachandran et al., 2017). Thus, ULMFiT (Howard and Ruder, 2018) for example achieved similar performance compared to a non pre-trained model, but with much fewer examples. Other approaches like ERNIE (Sun et al., 2020), XLNet (Yang et al., 2019) or BERT (Devlin et al., 2019) have manifested this trend and shown that pre-trained models are also capable of scaling up when increasing the number of model parameters or the amount of pre-trained data.
Based on the theoretical foundations presented in this chapter, the following chapter will outline related work that has been done towards document summarization.

3. Related Work

In this chapter, the previous work on multi-document summarization and pre-trained language models in general will be outlined. Furthermore, the approaches taken so far and how these approaches relate to the research questions defined in this thesis will be discussed.

3.1. Multi-Document Summarization

The task of multi-document summarization (MDS) within the domain of natural language processing can be defined as a subtask of document summarization. As mentioned earlier (Section 2.2) initial work on summarization was based on statistical techniques to automatically extract textual information.

3.1.1. Extractive Approaches

McKeown et al. (1999) introduced first approaches to create extractive multi-document summarization systems with the objective to summarize multiple documents in any given domain. For this, McKeown et al. analysed similarities between different documents through word co-occurrence, matching noun phrases, synonyms or semantic verbs as well as through applying term frequency-inverse document frequency (TF-IDF) which measures the relevance of a word in a given document. Approaches to graph-based extractive summarization over multiple documents were introduced by Mani and Bloedorn (1999), who mapped terms and their relationships for each document within a graph in order to extract and rank salient textual information. Goldstein et al. (2000) did subsequent work, describing the challenges of MDS regarding redundancy, temporal dependencies, compression and co-referencing. They further introduced the concept of *Maximal Marginal Relevance*, which strives to maximize relevance and novelty of sequences in extractive summarizations by creating a ranked list and computing a diversity ranking among the ranked features. Further early work included Radev et al. (2000), where the technique of *centroid-based summarization* was introduced which

takes clusters of centroids created by a modified TF-IDF approach as an input and identifies which sentences are relevant for a defined topic through ranking.
Conroy et al. (2006) further contributed to extractive MDS with a topic-based approach where the system is fed with a topic specified by a text description. The system then evaluates each sentence in each document regarding its similarity to the predefined topic.
Haghighi and Vanderwende (2009) implemented a model based on hierarchical *Latent Dirichlet Allocation* (LDA) for extractive MDS which creates a vocabulary of stopwords, draws a content distribution for each set of documents as well as a document-specific vocabulary distribution of words that only appear in a single document followed by a distribution over topics.
Another similar approach by Shen and Li (2010) describes extractive MDS towards the dominating set graph algorithm where each node is a vectorized sentence and the relations between nodes are determined through cosine similarity. Depending on the type of summarization the system is able to choose particular sentences for building the graph.
Work on document summarization or multi-document summarization in German language is rare. Zopf (2018) introduced an English and German MDS corpus and showed its applicability for supervised machine learning through performing summarization with baseline models that achieve comparable results.

3.1.2. Abstractive Approaches

McKeown and Radev (1995) first introduced abstractive approaches to MDS with summaries formed through planning operators that combine and include related information from individual templates where each template was made out of news articles that cover a particular event. Further Radev and McKeown (1998) adopted the template approach and created a summarization system that creates a summary from fill-in information and collected phrases to generate an abstractive text that expresses this information.
Barzilay and McKeown (2005) introduced an approach to abstractive MDS based on sentence-fusion and sentence-compression which generates sentences using information common to most sentences that were acquired from a text corpus. The generation of sentences is grounded on syntactic and lexical information derived from the input documents and knowledge from the text corpus. Summaries are created through altering and reusing phrases from input sentences. The work by Filippova and Strube (2008) extended this work by using a graph-based approach where a dependency tree of sentences is constructed and pruned by removing its subtrees.

A more recent approach to abstractive MDS includes work by Bing et al. (2015) where sentences are constructed through the exploration of noun and verb phrases that receive a rank based on their redundancy. The phrases will then be selected and sentences will be generated through *Integer Linear Programming* (ILP) with the *simplex* algorithm (Dantzig and Thapa, 1997) considering the functions and constraints as linear problems.

3.1.3. Neural Approaches

With the broad adoption on several NLP tasks, neural networks have also been applied to the task of document summarization.
Relevant work includes Rush et al. (2015) who created a neural attention model for MDS that combined extractive and abstractive methods. For abstractive modelling the work included a encoder-decoder architecture based on CNNs and the attention mechanism with a beam-search included in the decoder. For capturing extractive word matches where necessary, for example proper nouns, additional features were tuned through modifying the scoring function to estimate the probability of a summary using a log-linear model. Here, the combination of extractive and abstractive methods outperforms the approach of using an extractive method solely.
The work by Cheng and Lapata (2016) builds upon the previous work of Rush et al. (2015) while further supplementing the architecture with RNNs based encoder-decoder where the attention mechanism is directly applied on sentences or words instead of on the representative vector. This approach is commonly described as *Pointer Networks* (Vinyals et al., 2015).
The work by Nallapati et al. (2016) continued the previous approaches with the combination of extractive and abstractive approaches to summarization but further introduced a *switching generator pointer model* that tries to balance between abstractiveness and extractiveness, especially in cases where the word to be considered is not part of a predefined vocabulary from the source documents. In this work, Nallapati et al. also created the CNN/Dailymail dataset (Hermann et al., 2015) suitable for the task of document summarization, which was originally a dataset for the task of question answering.
Yasunaga et al. (2017) have done further research towards graph-based neural multi-document summarization. The work proposes a system that incorporates sentence relation graphs through a *graph convolutional network* (GCN) (Kipf and Welling, 2016) that uses a GRU model as a RNN-based regression model. The advantage of this graph-based approach is that it does not need the complexity of an decoder architecture and further relies on granular salience estimation and sentence selection in a

greedy manner while also mimicking the attention mechanism.Tan et al. (2017) also used a graph-based approach to MDS but in combination with a hierarchical encoder-decoder framework in order to further investigate the competitiveness of abstractive methods regarding extractive methods.
Picking up the previous research from Rush et al. (2015) and Nallapati et al. (2016) the work by Liu et al. (2018) continued with the task of text summarization of articles from multiple documents and added the recently presented non-recurrent architecture of *Transformer* models (Vaswani et al., 2017), which is able to increase the performance when longer sequences are entered. They further introduced a new MDS dataset with over 2 million examples based on Wikipedia articles. In this dataset, the articles are forming the summary while the cited sources as well as web search results for the corresponding title form the source documents.
Fabbri et al. (2019) created an analogous MDS dataset that focuses on the news domain and applied a hierarchical model with a pointer-generator network that incorporates maximal marginal relevance (MMR) and achieves competitive results compared to transformer models published by then.

3.1.4. Transfer Learning Approaches

Most recently fine-tuning pre-trained neural language models has gained a lot of attention for NLP tasks. One such work that attempted to state the impact of transfer learning on different tasks such as summarization includes Raffel et al. (2020). The work outlines the abilities of general purpose language models and is looking at flexible ways that can perform well on changing surroundings regardless of the specified underlying problem, including question answering, document summarization or sentiment classification for example.
An example that applied transfer learning is the work by Liu and Lapata (2019) who used the *BERT* model (Devlin et al., 2019) to obtain sentence representations. Here, the BERT model was used as a basis and was supplemented with transformer layers for extractive summarization and with an pre-trained encoder and a decoder for abstractive summarization. Results of their models showed the performance on extractive and abstractive summarization as well as a method that combined both approaches.
One of the most recent works towards multi-document summarization includes Hokamp et al. (2020) who fine-tuned BART (Lewis et al., 2020), a language model that generalizes the concepts of bidirectional encoders and autoregressive decoders. In the paper Hokamp et al. tuned the pre-trained language model on the CNN/Dailymail dataset and then further fine-tuned the model on three MDS datasets. For applying the model

on the task of MDS, the authors used dynamic ensemble decoding that allows to combine multiple outputs into a single output through using a *reduce* function taking into account one to eight source documents to reduce the complexity of the task.
Based on the related work presented in this chapter, the following chapter will introduce the datasets that were used to investigate new approaches for fine-tuning pre-trained language models for German multi-document summarization.

4. Datasets

This chapter introduces the three datasets that were used in order to answer the proposed research questions.
For the experiments, two English and one German dataset were selected. The first two datasets are the CNN/Dailymail dataset (Hermann et al., 2015) and the Multi-News dataset (Fabbri et al., 2019), which are one of the most common datasets for the task of text summarization. Secondly, one German dataset was chosen. The *auto-hMDS* dataset (Zopf, 2018) is currently the largest German dataset for the task of multi-document summarization. There are only two other document summarization datasets in German language, such as the *SwissText 2019*[1] dataset for single-document summarization that contains of 100,000 documents with reference summaries from the German Wikipedia or the multi-document dataset *DBS corpus* that consists of 93 summaries from 293 source documents (Benikova et al., 2016).

4.1. CNN/Dailymail

Hermann et al. (2015) first introduced the English *CNN/Dailymail* dataset, which was originally used as a dataset for the task of question answering. Nallapati et al. (2016) enrichened the dataset and created abstractive multi-sentence summaries from the available information that can be used for *Single-Document Summarization*. The corpus consists of 311,971 news articles with 781 tokens on average and their corresponding summaries that consist of 3.75 sentences or 56 tokens on average (See et al., 2017). The dataset has an *anonymized* and a *non-anonymized* version. In the following experiments, only the non-anonymized version is used.

[1] https://www.swisstext.org/swisstext.org/2019/shared-task/german-text-summarization-challenge.html

4.2. Multi-News

The *Multi-News* dataset is an English dataset for MDS that was introduced by Fabbri et al. (2019). The dataset consists of over 250,000 news articles with an average length of $\sim$ 2100 words and 56,216 human written summaries with an average length of 260 words from the news website newser.com. The summaries are linked to between two and ten source documents and the source documents are retrieved from over 1,500 different news sites.

4.3. auto-hMDS

Zopf (2018) introduced the *auto-hMDS* dataset with the approach of selecting available summaries from Wikipedia and search for corresponding source documents on the Internet to create a more heterogeneous dataset. This reverse approach mitigates the effort for creating a multi-document summary by extracting the first section of a Wikipedia article, called the *lead* section, which features the most important information and then use this section as a summary for the topic. The corresponding source documents were retrieved by using the sentences from the lead section combined with the topic name as a query through *Google Custom Search Engine* (CSE)[2] to create a list of corresponding links followed by retrieving the page content. The content of the webpage (as a .html file) was then pruned by removing all mark-up language tags and structure the topics where each topic consists of input and reference files in .txt format. The input files include the sentences from the retrieved webpages as well as a file that provides the URL of the webpages where the sentences were retrieved. The reference files include the reference from a particular Wikipedia article and the segmented sentences that were used for the web query to search for corresponding articles. The dataset was provided by the authors and consists of 7,316 German and English summarization topics. This work only utilizes the German dataset which has a corpus size of 2,210 summarization topics and 10,454 source documents and is the largest multi-document summarization dataset in German language.
While the authors did not make the dataset publicly available due to copyright restriction, the corpus can be retrieved via the provided URLs through web scraping or may be provided on request.[3].

[2] `https://developers.google.com/custom-search`
[3] `https://github.com/AIPHES/auto-hMDS`

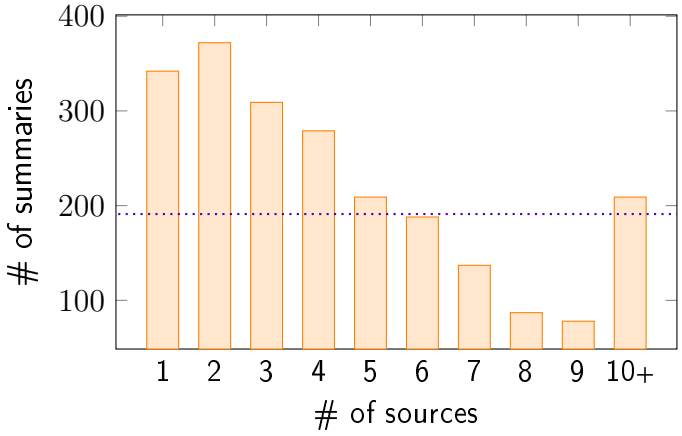

Figure 4.1.: Distribution of summaries to source documents. The dotted line marks the average number of summaries across all number of sources. "10+" means "10 or more source documents."

Data Exploration

With the goal to better understand the dataset, exploration steps will be applied to uncover patterns, characteristics and distribution of length and sources.
The 2,210 German summaries on average consist of 10.52 sentences and 182.15 words. A source document on average has a length of 160.21 sentences and 3,876.86 words. Figure 4.1 shows the distribution of number of source documents across the number of summaries. For example, the most common number of linked sources with 372 summaries is two sources, while 209 summaries have 10 or more sources. On average, a summary is linked to 4.73 source documents. 342 summaries are linked to only one source document.
Table 4.1 shows the distribution of summaries to the number of sources with the average number of sentences and words per summary and source. For the experiments the dataset will be split into training (80%), validation (10%) and test data (10%).
Compared to other multi-document datasets commonly used as benchmark datasets such as DUC 2004 (Paul and James, 2004) or TAC 2011 (Owczarzak and Dang, 2011), the auto-hMDS dataset is much larger and consists of more diverse topics from different genres such as books, animals, people or geographical sites (Zopf, 2018). Retrieved from the Internet, the summaries and the sources are also highly diverse in its origin and authors, as Wikipedia articles are commonly written by a wide range of different authors.

		source		summary	
summaries	**sources**	**sentences** (avg.)	**words** (avg.)	**sentences** (avg.)	**words** (avg.)
342	1	224.62	4596.07	136.76	6.95
372	2	212.04	4262.38	136.13	7.37
309	3	181.82	3754.35	143.60	8.04
279	4	177.62	3559.88	149.17	8.47
209	5	173.14	3580.46	177.30	10.11
188	6	163.71	3263.01	187.85	10.81
137	7	155.59	3153.34	191.93	11.41
87	8	137.18	2884.77	210.25	12.48
78	9	152.36	3121.72	233.86	13.58
209	10+	197.24	3296.34	254.67	16.01

Table 4.1.: Distribution of summaries to number of sources with average number of sentences and words per summary and source document.

With the proposed datasets in this chapter, the following chapter will present and describe the methods applied on these dataset.

5. Methodology

This chapter outlines the applied methods and the model that was utilized on the previously presented datasets in order to answer the initially declared research questions. Firstly, the following baseline models describe rudimentary methods on how to tackle the research questions in a heuristical way. Secondly the methodology, architecture and training of the chosen language model, the BART model (Lewis et al., 2020), will be discussed.

5.1. Baseline Methods

For the baseline approaches the Top-N sentences and *LexRank* (Erkan and Radev, 2004) were chosen. Both methods score sentences independently and generate the summary by selecting top-scored sentences.

5.1.1. Top-N Sentences

The Top-N sentences baseline method hypothesizes that the weighted occurrence frequency of a word in a given text gives an indication on the importance of a sentence. In a first step the text is tokenized in sentences and words, stored in separated variables. In order to remove stopwords that should be excluded from the word occurrence list, the method loads German stopwords from the Python library *NLTK*[4] and first checks if the word to be considered is listed as a stopword. If the word is not a stopword and encountered for the first time it is added to a vocabulary that holds each word with its corresponding frequency score. The weighted frequency will then be calculated by dividing the number of occurrence of all words by the frequency of the most occurring word.
Based on the weighted occurrence frequency of a word the sentence score for each sentence will then be calculated. To summarize a article, the top N sentences with the highest scores will be selected through the *heap sort* algorithm (Williams, 1964)

[4] https://www.nltk.org/

and concatenated to shape the final summary. Here, the summarization will be applied regarding the top-5 sentences.

5.1.2. LexRank

Erkan and Radev (2004) proposed the unsupervised graph-based *LexRank* algorithm with the hypothesis to evaluate a text regarding the lexical *centrality* of each sentence in a cluster followed by extracting the most important sentences and including them in a summary. Here, the centrality of a sentence can be defined by their similarity to other sentences in the same cluster. The similarity between two sentences is calculated through the cosine between two corresponding vectors and a cluster of documents is represented by a *cosine similarity matrix*. A graph is constructed with a predefined threshold for similarities and with each sentence representing a node that is connected with similar sentences. For the implementation of this algorithm the Python package *lexrank*[5] was used.

5.2. BART

With the condition of putting recent advances on pre-trained language models into account that achieve state-of-the-art results, the BART model (Lewis et al., 2020) was chosen. BART is implemented as a sequence-to-sequence model with bidirectional encoders and a left-to-right autoregressive decoder that utilizes the standard transformer architecture (Vaswani et al., 2017) whilst adapting the concepts of bidirectional encoders from BERT (Devlin et al., 2019) and autoregressive decoders from GPT-2 (Radford et al., 2019). The model consists of two versions; one base model that uses 6 layers in the encoder and decoder and one large model that uses 12 layers in each. The BART model is currently one of the best performing models for the task of text summarization with the premise of a significant performance increase through further fine-tuning (Lewis et al., 2020). Figure 5.2 depicts the architecture of BART with the bidirectional encoder and autoregressive decoder as its key components.
The model is trained with corrupted text through an arbitrary noising function and a sequence-to-sequence model that learns to reconstruct the original text. The encoder reads the sequential input e.g. a document to summarize while the decoder generates the outputs autoregressively. Both layers are connected by cross-attention where each decoder layer focuses on specific aspects over the final state of the encoder output creating sequences, closely connected to the initial input. The bidirectional encoder

[5] https://pypi.org/project/lexrank/

architecture takes all previous and subsequent tokens into account for predicting a masked token. In cases of text generation BERT without any modification loses its strength of bi-directionalism and becomes directional towards past words when following words are hidden. Here, BART adopts the architecture of GPT-2 to predict future words only by utilizing previous words. The advantage of BART therefore is the combination of contextual embeddings from BERT and text generation from GPT-2. Transformation as described in Lewis et al. (2020) can be implemented through token masking, token deletion, text infilling, sentence permutation or document rotation.
With its introduction, BART achieved state-of-the-art results in common NLP tasks such as question answering or summarization with the possibility of performing fine-tuning directly on these tasks. [6][7]

Training

The utilized implementation of BART (Lewis et al., 2020) includes generating summaries from the pre-trained BART large model that consists of 12 encoders and 12 decoder layers and was trained on 400 million parameters. Pre-training is done through corrupting documents and applying the model on reconstructing textual information between the decoder's output and the original document, as depicted in Figure 5.1.
The original document, here represented by $ABCDE$, will be processed by masking random tokens or by additionally inserting tokens before encoding. In this example the span $[C, D]$ is masked. The decoder then needs to reconstruct the original document by using the encoder's output and leaving the previous tokens uncorrupted.

For fine-tuning, the model was firstly preprocessed on every dataset with byte-level *byte pair encoding* (BPE) followed by binarizing the dataset. The BPE compression algorithm (Sennrich et al., 2016) allows to use representation of an open fixed-size vocabulary that consists of character sequences of variable length. The character sequences are merged to words where the most frequent pairs are replaced by a new symbol that represents a character n-gram. The result of this process is a vocabulary that is of equal size as the initial vocabulary with a segmentation of words in frequent characters. Other language models such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019a) also use this approach.
On each dataset fine-tuning tasks were applied by splitting the corpus into training (80%), validation (10%) and test (10%) and fine-tuning BART on the training part. The model was fine-tuned for 10 epochs with a batch size of 100 and an initial learning

[6] https://github.com/pytorch/fairseq/
[7] https://huggingface.co/transformers/

rate of $3e{-}05$. The training was conducted by using four GPUs type GeForce GTX1080 Ti with 11 GB RAM. The checkpoint with the best validation performance was then picked for applying the model on the remaining training data.
Based on the methods described in this chapter, the following chapter outlines the execution of these methods and their comparative evaluation.

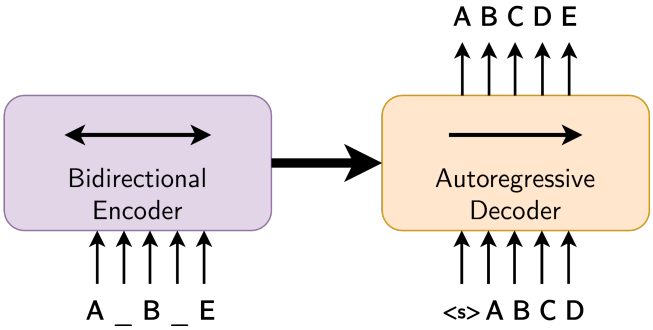

Figure 5.1.: Pre-training of BART with a bidirectional encoder that takes a masked input and passes the input to an autoregressive decoder that must reconstruct the original document, using the encoder's output and previous uncorrupted tokens. Figure according to Lewis et al. (2020).

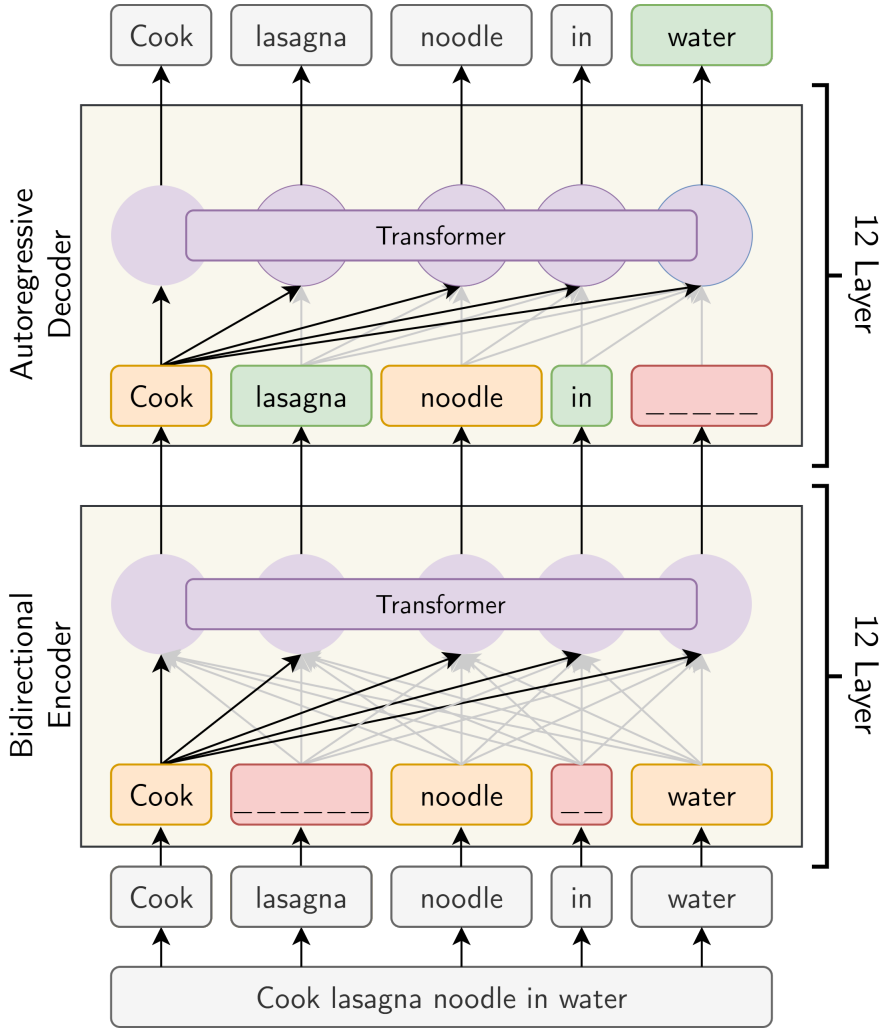

Figure 5.2.: Architecture of the BART sequence-to-sequence model with an 12-layer bidirectional encoder on the bottom and an 12-layer auto-regressive decoder on the top (Lewis et al., 2020).

6. Experiments

This chapter describes the summarization experiments and their results that were conducted in order to answer the research questions outlined in Section 1.2. Research question 1 seeks to investigate the potential of pre-trained language models when adapted to other languages. This will be measured by performing summarization firstly on the two English datasets to reproduce previous results followed by the auto-hMDS dataset. The results will be evaluated with the ROUGE evaluation metric. With research question 2, gaps and potential errors in language models and their applicability across languages will be investigated through experiments on the auto-hMDS dataset and a manual comparison of the results with previous results from the English datasets to point out erroneous patterns. Lastly, to answer research question 3 the experiments will be concluded with defining potential shortcomings of the applied language model.
First experiments were carried out towards single-document summarization of the CNN/Dailymail (Nallapati et al., 2016) dataset to explore the necessary method and settings of the BART (Lewis et al., 2020) model. These results will be compared to previous approaches applied on the dataset. Subsequently, the consequences of the findings will be applied on the Multi-News (Fabbri et al., 2019) dataset where similar actions will be taken but with the additional complexity of summarizing textual information from multiple documents. Here, the results will be compared to previous approaches. Also the possibility of reusing SDS models for MDS and achieving comparative results (as in Lebanoff et al. (2018)) is demonstrated. Finally, the model will be utilized for fine-tuning on the auto-hMDS dataset.
All results will be evaluated using the ROUGE evaluation metric, outlined in Section 2.3 and originally introduced by Lin (2004).

6.1. BART on CNN/Dailymail

In a first experiment, the results of BART on the SDS dataset CNN/Dailymail (Nallapati et al., 2016) will be reproduced that were part of the paper by Lewis et al. (2020) which initially introduced the BART model. The results are reproduced in order to find similar settings and explore common techniques before applying the model on a

new dataset, the task of MDS or a new language. Firstly the pre-trained version of the model was applied on the dataset to investigate the performance without fine-tuning. Afterwards the model was fine-tuned on the dataset using the model implementation of BART based on the *transformers* library (Wolf et al., 2019).
The dataset was split in 287,226 training pairs, 13,368 validation pairs and 11,490 pairs, following the approach of using the *anonymized* version of the data (See et al., 2017). Before fine-tuning, the data was preprocessed through tokenization using the BPE algorithm that splits words into more frequent subwords and builds a vocabulary from the dataset followed by transforming the textual information into binary representations. Lastly, the dataset was fine-tuned, using the same settings as in the paper by Lewis et al. (2020) with a beam search of size 4, allowing n-grams up to a size of 3, using the Adam optimizer (Kingma and Ba, 2015) default settings of $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a learning rate of $3e - 05$. Table 6.1 shows the results of the pre-trained and fine-tuned BART model on the CNN/Dailymail dataset as compared to previous models.
The ORACLE baseline method that chooses the best three sentences based on the ROUGE score achieved the highest results, which indicates the extractive characteristics of the dataset. Furthermore the table shows that the fine-tuning significantly improved the ROUGE scores as compared to the pre-trained model by more than 50%. The fine-tuned model achieves state-of-the-art results with scores that are comparable or outperform nearly all previous scores. Recently published work such as the model MATCHSUM by Zhong et al. (2020) outperforms the model on all three ROUGE scores, but only focuses on extractive summarization. Finally, the results of the fine-tuned model are very close to the results originally published by Lewis et al. (2020), which provides confidence that the environment and settings are correctly aligned for a further continuation of the experiments.

Model	R-1	R-2	R-L
ORACLE	52.59	31.24	48.87
LEAD-3	40.42	17.62	36.67
Extractive			
NEUSUM (Zhou et al., 2018)	41.59	19.01	37.98
SUMO (Liu et al., 2019b)	41.00	18.40	37.20
Abstractive			
BOTTOMUP (Gehrmann et al., 2018)	41.22	18.68	38.34
DCA (Celikyilmaz et al., 2018)	41.69	19.47	37.92
BERT			
BERTSUMABS (Liu and Lapata, 2019)	41.72	19.39	38.76
BERTSUMEXTABS (Liu and Lapata, 2019)	42.13	19.60	39.18
MATCHSUM BERT-base (Zhong et al., 2020)	**44.22**	**20.62**	**40.38**
BART			
BART-LARGE pre-trained	25.98	11.26	17.50
BART-LARGE fine-tuned	42.21	19.10	35.38

Table 6.1.: Comparative evaluation on CNN/Dailymail (Nallapati et al., 2016) dataset. Own results are marked in orange. Best results are in bold print.

6.2. BART on Multi-News

Proceeding with the previous settings, the model is applied on the MDS Multi-News (Fabbri et al., 2019) dataset. As the model expects single inputs, preprocessing is applied by concatenating multiple source documents to one source file that serves as the input file.

The dataset was split into 44,972 training pairs, 5,622 validation pairs and 5,622 test pairs. Again, the BART model was applied pre-trained on the dataset followed by fine-tuning the model on the data. For the fine-tuning, the previous approach was adopted with BPE for tokenization and the same fine-tuning settings. Table 6.2 shows the results of the pre-trained and fine-tuned BART model on the Multi-News dataset compared to the baseline approaches and previous introduced models. The table shows that the BART model scores are higher than the originally proposed HI-MAP (Fabbri et al., 2019) and is only surpassed by MATCHSUM (Zhong et al., 2020) and BERTEXT (Liu and Lapata, 2019) with around 5 points on ROUGE-1 scores, around 1 point on ROUGE-2 and nearly 20 points on ROUGE-L.

Model	R-1	R-2	R-L
ORACLE	43.08	14.27	38.97
LEAD	**49.06**	**21.54**	**44.27**
HI-MAP (Fabbri et al., 2019)	40.08	14.90	19.70
BERTEXT (Liu and Lapata, 2019)	45.80	16.42	41.53
MATCHSUM (Zhong et al., 2020)	46.20	16.51	41.89
BART-LARGE pre-trained	30.67	10.05	16.99
BART-LARGE fine-tuned	40.58	15.50	21.73

Table 6.2.: Comparative evaluation on Multi-News (Fabbri et al., 2019) dataset. Own results are marked in orange. Best results are in bold print.

The experiment is concluded with an manual analysis on around 50 summaries to check for further redundancy due to truncating the source documents to a single input. The investigation shows that the model is performing well with concatenating multiple documents to one input document. It also becomes apparent that the model is able to handle redundancy through e.g. n-gram blocking without repetitions or leaving out salient information. The randomly picked example in Table 6.4 depicts the fluency and coherence of the summaries. With these findings the experiment will be proceeded on the German auto-hMDS dataset.

Source 1
Adrianne Haslet-Davis, who lost her left leg below the knee in the Boston Marathon bombing, said she loves a good challenge. This April, she'll "take on a great one." Haslet-Davis announced Tuesday morning on the Hallmark Channel that she is training for this year's 120th Boston Marathon. Her team, "#AdrianneStrong," which she formed about five weeks ago, will raise funds...
Source 2
A woman who lost part of her leg in the boston marathon bombing two years ago returned to the site yesterday– but not to run. She returned to show that she could still... Dance. The elegant steps of a professional ballroom dancer were the only thing moving along Boylston Street this morning. Adrianne Haslet-Davis fox-trotted across the marathon finish line and trampled on the memory of...
Generated Summary R-1 = 54.28 R-2 = 30.77 R-L = 43.80
Adrianne Haslet-Davis, who lost her left leg below the knee in the Boston Marathon bombing two years ago, is training for this year's 120th Boston Marathon. "I made the decision a little while ago," she tells Boston.com. "I wanted to train and really see if I was ready this year both physically and mentally before announcing that I've been training." She's planning to run the April 18 race with her team, #AdrianneStrong, which will raise funds for the Oklahoma City-based Limbs for Life Foundation. "It's just a giant thank you to everyone," she says, per the Boston Globe.

Table 6.4.: Example of a generated summary of the BART (Lewis et al., 2020) model on the Multi-News dataset (Fabbri et al., 2019).

6.3. BART on auto-hMDS

Finally, the BART model was applied on the MDS *auto-hMDS* dataset, introduced by Zopf (2018). In contrast to previous approaches the models are computing a summary of predefined length of 100 or 200 words in order to make them comparable to the methods introduced in Zopf (2018). Here, it can be seen that a longer summary also implies decreasing ROUGE scores, regardless of the model to be considered. The ORACLE R-1 and ORACLE R-2 scores choose the best sentences in a greedy fashion according to the ROUGE score and indicate the best possible scores which can be achieved.
For initial investigations on the dataset, the summarization methods of LexRank (Erkan and Radev, 2004) and Top-5 Sentences were applied. Here, the Top-5 Sentences method fills the summary until 100 respectively 200 words are reached. These models already outperform the methods conducted by Zopf (2018), including LEAD and RANDOM, on all three ROUGE scores. The experiment with BART (Lewis et al., 2020) continues by adapting previous settings with firstly running the pre-trained version on the dataset, followed by fine-tuning the model on the dataset. The dataset consists of 2,210 pairs and was split for training purpose into training (80%, 1,768), validation (10%, 221) and test data (10%, 221). Again, the multiple source documents will be concatenated in order to create one single input file for each input.

	100 words		200 words	
Method	**R-1**	**R-2**	**R-1**	**R-2**
Random (Zopf, 2018)	18.57	1.85	25.53	3.25
Lead (Zopf, 2018)	12.29	2.61	10.56	2.28
Oracle R-1 (Zopf, 2018)	43.02	21.61	47.69	21.17
Oracle R-2 (Zopf, 2018)	45.94	29.27	48.64	29.24
Top-5 Sentences	21.90	4.60	19.81	4.14
LexRank	29.91	6.67	24.19	5.51
BART pre-trained	28.48	8.79	20.84	6.02
BART fine-tuned	**38.43**	**12.93**	**30.24**	**9.09**

Table 6.5.: Comparative evaluation on auto-hMDS (Zopf, 2018) dataset. Own results are marked in orange. Best results are in bold print.

Table 6.5 shows the results of the BART model pre-trained and fine-tuned on the

dataset compared to previous approaches as well as to LexRank and Top-5 Sentences. The fine-tuned model outperforms all other methods on all ROUGE-scores. The example in Table 6.7 depicts a summary generated from the BART model on the auto-hMDS dataset (Zopf, 2018). Orange-marked text in the generated summary indicates text from Source 1 while violet-marked text is taken from Source 2. Red-colored text in the generated summary marks syntactical or grammatical mistakes. As this summary mainly consists of extractive fragments, it outlines that the model uses parts from the whole source, not only relying on the information provided at the beginning of the source document. It is also capable of re-ordering, replacing and combining these extractive fragments. Furthermore, the summary shows erroneous abstractive text generation, here color-marked in red, but also valid abstractive approaches such as "Die meisten Arten der Gattung werden als 'Nagetier' bezeichnet".
In this chapter, experiments were conducted in order to answer the research questions outlined in the beginning. To further analyse the results, the following chapter will explore the results regarding extractiveness, quality and shortcomings.

Source 1
Das Capybara (Hydrochoerus hydrochaeris) ist eine Säugetierart aus der Familie der Meerschweinchen (Caviidae). Es ist das größte heute lebende Nagetier und lebt in feuchten Regionen Südamerikas. Es wir auch als Wasserschwein bezeichnet [...]
Source 2
Das Wasserschwein gehört zur Familie der Meerschweinchen. [...] Dabei werden die Weibchen etwas größer als die Männchen. Somit gilt das Wasserschwein als größtes heute lebende Nagetier weltweit und hat trotz seines Namens nichts mit Schweinen zu tun. [...] Die Heimat reicht von Panama bis Nordostargentinien, östlich der Anden. Als Lebensraum dienen aufgelockerte Waldbestände mit deckungsreichem Unterwuchs oder auch ausgedehnte, hochgewachsene und versumpfte Grasfluren. [...]
Gold Summary
Das Capybara oder Wasserschwein (Hydrochoerus hydrochaeris) ist eine Säugetierart aus der Familie der Meerschweinchen (Caviidae). Es bildet gemeinsam mit dem Panama-Capybara (Hydrochoerus isthmius) die Gattung Hydrochoerus und ist das größte heute lebende Nagetier. Es bewohnt feuchte Regionen in Südamerika und ist vom Körperbau seiner semiaquatischen (teilweise im Wasser stattfindenden) Lebensweise ideal angepasst.
Generated Summary R-1 = 39.28 R-2 = 24.09
Das Capybara Aber (Hydrochoerus hydrochaeris) ist eine Säugetierart aus der Familie der Meerschweinchen (Caviidae). Es ist das größte heute lebende Nagetier und lebt in feuchten Regionen Südamerikas. Die Heimat reicht von Panama bis Nordostargentinien, östlich der Anden. Als Lebensraum dienen aufgelockerte Waldbestände mit deckungsreichem Unterführung oderuch aus verschiedenen Gründungsgebieten im Regierungsraum. Die Art ist vor allem im tropischen Gebiet verbreitet, wenn sie nur in sehr geringen Regionen lebend ist. Die meisten Arten der Gattung werden als „Nagetier“ bezeichnet.

Table 6.7.: Example of a generated summary of the BART (Lewis et al., 2020) model on the auto-hMDS dataset (Zopf, 2018). Orange-marked text in the generated summary indicates text from Source 1 while violet-marked text is taken from Source 2. Red-colored text in the generated summary marks syntactical or grammatical mistakes.

7. Analysis

In this chapter, the results obtained in the experiments described above are analysed in detail. Even though the applied model achieves state-of-the-art results when evaluated with the automatic ROUGE metric (Lin, 2004), this metric is only able to assess the models output regarding the N-gram overlaps and therefore only provides limited information. These limits are also discussed by Schluter (2017) and Maynez et al. (2020). In order to further investigate the quality of the model and its outputs the following sections analyse the results regarding the applied datasets and the model-generated outcome. More examples of the results can be found in the Appendix A.

7.1. Dataset Analysis

In order to further understand the discussed results, it is necessary to analyse the datasets regarding their extractiveness, quality and shortcomings to fully understand the outcome.

7.1.1. Extractive vs. Abstractive

In a first step, the extractiveness of the model will be evaluated and compared to the extractiveness of the provided summaries from each dataset. For this, the summaries generated by the fine-tuned BART model will be compared to the gold summaries on all three datasets (CNN/Dailymail, Multi-News, auto-hMDS). Even though, according to one of the recent studies (Lewis et al., 2020), the BART model output is "highly abstractive, with few phrases copied from the input", this could not be confirmed here, as we observed summaries are mainly built from extractive fragments or even whole paragraphs, as can be seen in Section 6.5. For further quantitative investigation, the extractiveness was measured by using the method of *extractive coverage* and *extractive density*, introduced by Grusky et al. (2018).
In this work, Grusky et al. (2018) described a method with an article A that consists of a sequence of tokens $(a_1, a_2, ..., a_n)$ and a corresponding article summary S that consists of a sequence of tokens $(s_1, s_2, ..., s_m)$. The set of shared extractive fragments

can be defined by $\mathcal{F}(A, S)$ and calculated through a greedy algorithm. The *extractive fragment coverage* quantifies the similarities between the summary and the text and is measured by:

$$COVERAGE(A, S) = \frac{1}{|S|} \sum_{f \in \mathcal{F}(A,S)} |f| \tag{7.1.1}$$

As a high coverage only quantifies that the words appear in the article and the summary disregarding its order, the *extractive fragment density* will also be calculated, which can be defined by $DENSITY(A, S)$ and measures the average length of the extractive fragment to which each word in the summary belongs. Additionally to the coverage, the square of the fragment length is used:

$$DENSITY(A, S) = \frac{1}{|S|} \sum_{f \in \mathcal{F}(A,S)} |f|^2 \tag{7.1.2}$$

The density and coverage are used to understand to what extent the distribution of the provided summaries across the three datasets and the distribution of the generated summaries from the model exhibit similarities.

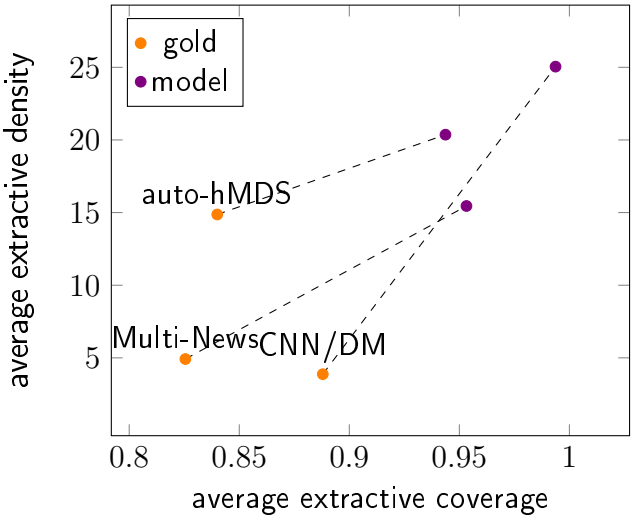

Figure 7.1.: Comparison of extractiveness of gold summaries and model-generated summaries with extractive coverage and extractive density.

Figure 7.1 shows the behavior of the model regarding extractiveness measured by the

average extractive density and average extractive coverage. Here, it becomes clear that the model-generated summaries from fine-tuned BART on all three datasets are much more extractive than their gold counterparts with an average extractive coverage over 94%. It becomes visible that the gold summaries are already much more extractive than the average human-generated summary, which tends to have a 79% match to sentences in the source document, according to Kupiec et al. (1995). The BART generated summaries on all three datasets further diverge to extractiveness and therefore create summaries that are unlike summaries generated by humans. This is especially the case for the CNN/Dailymail summaries, which nearly have an average extractive coverage of 100% and the highest average extractive density.

7.1.2. Quality of hMDS dataset

Random samples on 50 generated summaries of the auto-hMDS dataset (Zopf, 2018) clearly shows false and misleading results in at least 22 summaries (44%). When inspecting the source documents corresponding to these summaries it becomes visible that the source documents contain a lot of textual noise that do not provide any information for the summaries and occasionally cover other topics. This noise mainly comes from scraping the website that appears to be of high relevance according to the search engine but sometimes does not match the topic at all. Furthermore, textual noise is included from scraping content without profoundly cleansing the data.

Off-topic Information

The most severe flaws on the investigated generated summaries are coming from source documents that provide off-topic information. As the source documents were crawled by searching for corresponding websites that include fragments of the summary, some source documents mainly consist of information that are not relevant for the summary. The example in Table 7.2 about the aircraft A380 illustrates this issue where the off-topic information is color-marked in orange. The ROUGE score of this example is furthermore above average, which also confirms that the ROUGE metric alone is insufficient to evaluate the coherence or salience of a summary.

Website Information

The analysis also showed that the source documents include information about the webpage that do not provide any information for the correspondig topic. For example a lot of source documents include a sentence regarding the cookie consent which

Source 1
Pro-Kopf-Konsum von Schokolade in Europa nach Ländern [...] Container-Reedereien nach gesamter Anzahl der Schiffe 2017 Unternehmen in Deutschland nach Beschäftigtengrößenklassen (Stand Februar 2016) [...]
Generated Summary R-1 = 41.93, R-2 = 6.55, R-L = 25.80
Die Airbus-A380-Flugzeuge ist ein ausgelieferter Schokolade von Airbus. Sie ist das zweite ausgestorbene aus der Gruppe der Airbus Group und wurde im Jahr 2007 eröffnet. Die A380 ist die älteste A380-flugzeugart, die nach der Airbus A300-Ausstrahlung in den Jahren 2007 bis 2017 zu Beginn des Zweiten Weltkrieges zu sehen ist. Die Airbus A380 gehört zu den bedeutendsten Schiffe.

Table 7.2.: Example of a poor summary due to too much off-topic information in the source documents.

should be included in every webpage according to the directive 2009/136/EC of the European Union (EU). Here, a quantitative analysis discovers that more than 2,372 source documents consist of such a sentence. While these sentences do not only not provide any useful information for the summary, they further distract summarization systems from the relevant content and should therefore be removed. A similar error can also be found in 4,354 source documents that still contain a copyright note concering the website.
Also 3,061 source documents include login information for users of the website. This again is arbitrary noise information and should be removed.
Nevertheless, the inspection of the BART model shows that these information are mostly filtered out and not included in the summary.

7.2. Result Analysis

With the objective to further understand the results, the following section provides detailed information about the effect of fine-tuning, the created summaries and their compression as well as factual errors that were found through a manual analysis.

7.2.1. Effect of Fine-Tuning

Figure 7.2 visualizes the effect of fine-tuning regarding the ROUGE score (Lin, 2004) which was done in the experiment in Chapter 6. The results on the auto-hMDS show the variance of fine-tuning the dataset with summarization on 100 or 200 words. For each dataset the fine-tuning improved the results by a respectible margin, although

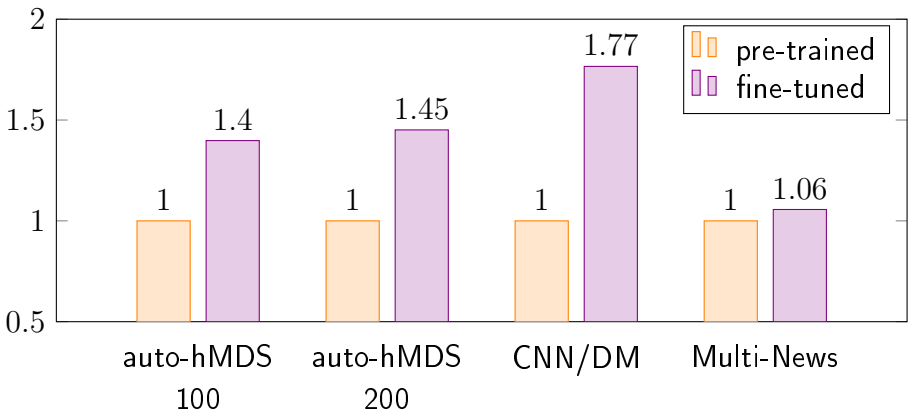

Figure 7.2.: Effect of fine-tuning compared to the pre-trained BART model results.

the Multi-News dataset only improved its performance by 6% regarding the pre-trained version. Moreover it becomes visible, that the auto-hMDS dataset is eligible to use for fine-tuning with the objective to improve the performance.

7.2.2. Result Comparison

In order to compare the generated summaries from the introduced models, the following Table 7.4 shows example output summaries on the auto-hMDS dataset with the ROUGE-1, ROUGE-2 and ROUGE-L score compared to the Top-N Sentence model and LexRank. It becomes visible, that the Top-5 Sentences summary is missing out facts that might be of interest for a summary. The LexRank algorithm creates more stringent summaries but also contains sentence repetition and off-topic information. The summary generated by the BART model, although quite extractive, covers nearly all important information and is also fluent and coherent.

7.2.3. Compression

Another objective of summarization is to compress necessary information into a shorter concise version (Jing, 2002). To quantify this, the compression ratio defines the word ratio between the source document d and the summary s and can be calculated as follows (Equation 7.2.1) (Grusky et al., 2018):

$$COMPRESSION(d, s) = \frac{|D|}{|S|} \tag{7.2.1}$$

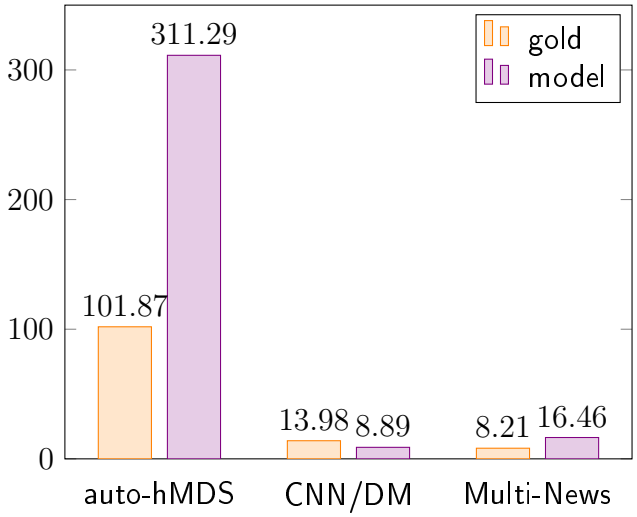

Figure 7.3.: Compression rate on all three datasets on the generated summaries from the BART model and the provided summaries of the datasets.

Figure 7.3 depicts the compression rate of the generated summaries as compared to the gold summaries on all three datasets. It becomes clear that the German *auto-hMDS* dataset compresses a lot more information in order to create the summary. The model-generated summary compresses three times more information than the provided gold summary. More interesting is that the compression rate is much higher than on the two English datasets with an model-generated summary compression rate of 8.89 (CNN/Dailymail) and 16.64 (Multi-News). From this it can be concluded that the German dataset does not only contain a lot of noise information, as outlined before, but also that models such as BART have to compress much more textual data in order to generate a summary.

7.2.4. Factual Errors

In a second step of the analysis, we manually examined randomly selected generated summaries from the BART model. For this, 50 summaries from each dataset (Fabbri et al., 2019) were selected and reviewed for factual mistakes or errors. At least 11 out of these 150 summaries, amounting to 7%, contained severe factual errors. As these errors exhibit different characteristics, the errors will be outlined in the following.

Table 7.6 shows such a factual mistake. The source document as well as the human-generated summary in this case describe a physical attack of a woman against a man. The model-generated summary (in Table 7.6) though twists the fact and produces a summary that accuses the man of a physical assault against the woman. This summary is not only factual incorrect but furthermore could indicate gender biasing that nurtures societal stereotypes where physical attacks mostly are carried out by men.

Another factual error in the Multi-News dataset was observed in a model-generated summary, which can be found in Table 7.8. Here it was erroneously stated that the singer Bob Dylan died in 2013. This fact was not included in the source documents nor in the human-generated summary and was therefore produced by the model. Such an error could also be found in the model results on the German auto-hMDS (Zopf, 2018) dataset, as depicted in Table 7.10. In this example the date and place of death were factually incorrect, while all other information in the text is correctly derived from the source documents.
The investigation on these errors are time-consuming and cognitively challenging, because the errors are – as shown – often embedded in correct context, which makes it especially hard to uncover the occurrence of such mistakes.

Gold Summary
Die Gruppe Trio war eine Band, die während der Neuen Deutschen Welle 1982 bekannt wurde. Prägnant an Trios Musik war in den Anfangsjahren der Minimalismus ihrer Liedtexte als auch ihre Beschränkung auf nur zwei Instrumente (Schlagzeug und Gitarre). Ihr größter Hit war Da da da, mit dem Trio auch international bekannt wurde.
Top-5 Sentences R-1 = 19.23, R-2 = 1.94, R-L = 9.61
"Da Da Da, ich lieb dich nicht, du liebst mich nicht" wurde 1982 zu einem internationalen Megahit und verhalf der Band zum Durchbruch. 1984 legten Trio eine kleine Pause ein, nach der sie ihr letztes Album "Whats the Password" veröffentlichten, das allerdings floppte. 1981 erschien das erste Album der Band, "Trio", beim Label Phonogram. Der Songtext war nicht so ausgefeilt, aber vielleicht gerade deshalb landete die Band "Trio" mit "Da da da" einen Welthit. Er erreichte in Deutschland Platz 2 der Charts (Platz 1 war Nicoles "Ein bisschen Frieden"), verkaufte sich weltweit 13 Millionen Mal und wurde in 30 Ländern veröffentlicht.
LexRank R-1 = 23.40, R-2 = 2.15, R-L = 12.76
1981 erschien das erste Album der Band, "Trio", beim Label Phonogram. 1981 erschien das erste Album der Band, "Trio", beim Label Phonogram. Der Songtext war nicht so ausgefeilt, aber vielleicht gerade deshalb landete die Band "Trio" mit "Da da da" einen Welthit. Der Songtext war nicht so ausgefeilt, aber vielleicht gerade deshalb landete die Band "Trio" mit "Da da da" einen Welthit. Trio waren auch für ihre unterhaltsamen Auftritte bekannt: Hier liest Drummer Peter Behrens eine Zeitung beim Spielen Mit ihrer Musik schafften es Trio, über die deutschen Landesgrenzen hinaus bekannt zu werden.
BART R-1 = 70.49, R-2 = 60.00, R-L = 70.49
Die Gruppe Trio war eine Band, die während der Neuen Deutschen Welle 1982 bekannt wurde, sich aber von jener distanzierte. Prägnant an Trios Musik war in den Anfangsjahren der Minimalismus, der sich sowohl textlich als auch in der Reduzierung auf nur zwei Instrumente (Schlagzeug und Gitarre) ausdrückte. Ihr größter Hit war „Da da da“, mit dem Trio auch international bekannzt wurden..

Table 7.4.: Output of the three models Top-5 sentences, LexRank and BART compared to the gold summary.

Generated Summary R-1 = 66.67, R-2 = 21.84, R-L = 28.33
Scott Baio is one of Donald Trump's strongest celebrity supporters in Hollywood, and now he's accused of physically assaulting the wife of a Red Hot Chili Peppers drummer. TMZ reports the actor filed a police report after Nancy Mack, wife of Chad Smith, confronted him at an event in Thousand Oaks, Calif., over his support of the president-elect. Mack's husband has been vocal about being anti-Donald Trump, citing him as a racist, and Baio says Mack confronted him for supporting Trump at a function with their elementary school kids. Baio told police that he asked Nancy Mack to keep it down because kids were present, but she didn't back down. [...]

Table 7.6.: Example of a generated summary of the BART model (Lewis et al., 2020) on the Multi-News dataset (Fabbri et al., 2019) that exhibits gender biasing.

Generated Summary R-1 = 67.99, R-2 = 29.91, R-L = 31.41
Pierce Brosnan's Bob Dylan painting sold for more than $1 million, and TMZ reports the new owner is none other than the woman who plunked down $17.8 million on Kim and Kanye's Bel-Air mansion. Ukrainian billionaire Marina Acton snapped up 007's painting of Zimmy earlier this week at the 25th annual amFAR Cannes charity gala. The former James Bond star, 65, who was trained as a commercial artist and worked as an illustrator, just auctioned off one of his paintings for $1.4 million, depicting the singer, who died in 2013.[...]

Table 7.8.: Example of a generated summary of the BART (Lewis et al., 2020) model on the Multi-News (Fabbri et al., 2019) which exhibits factual mistakes.

Generated Summary R-1 = 55.88, R-2 = 11.94, R-L = 30.88
Andrew Johnson (* 29. Dezember 1808 in Raleigh (North Carolina, USA; † 15. April 1865 in Greeneville, Tennessee) war der dritte Vizepräsident der Vereinigten Staaten, der durch den Tod seines Vorgängers ins Amt kam und der erste nach einem Attentat. Als Hauptaufgabe seiner Präsidentschaft galt die sogenannte Reconstruction, der Wiederaufbau der Südstaaten nach dem Krieg und ihre Wiedereingliederung in die Union.[...]

Table 7.10.: Example of a generated summary of the BART model (Lewis et al., 2020) on the auto-hMDS dataset (Zopf, 2018) which exhibits factual mistakes.

8. Summary, Conclusion & Future Work

8.1. Summary

In this thesis, the research questions towards the adaptability of pre-trained language models on other languages and potential errors were answered by performing the task of fine-tuning the pre-trained model towards multi-document summarization and analyse the results on three datasets. Throughout this thesis, it has been demonstrated that pre-trained language models are capable of multilingual applicability, exemplified by German multi-document summarization. Throughout the analysis, several erroneous patterns and gaps were found in the language model and datasets. The introduced experiments and analysis showed that the model is capable of adapting not only to German textual information but can also be fine-tuned with a small dataset following a similar approach to further improve results by a reasonable margin. Because the model was not adjusted towards handling multi-document structures, the findings might also be applicable on single document summarization.
The applied methods firstly included experiments with the largest multi-document summarization datasets in German language which were put into relation to two other common summarization datasets. These experiments included the use of one of the recent models for summarization, namely the BART model, and common baseline methods for summarization. The results show state-of-the-art performance on multi-document summarization with the BART model and achieved best performance for multi-document summarization on German language. Additionally, the analysis of the introduced datasets has been included to further investigate the acceptance and behavior of the model towards pre-training and fine-tuning. The analysis revealed that the model is able to further improve its performance when fine-tuned on German textual information and that it is able to produce coherent and fluent summaries from multiple sources. Furthermore we investigated the extractiveness of the BART model and uncovered factual errors across languages and quality losses due to dataset shortcomings.

8.2. Conclusion

This thesis showed the possibility of using transfer learning and pre-trained language models and fine-tune the model on German textual information. The experiments demonstrate that the fine-tuning process can be applied in a similar fashion using known settings and approaches and that even a small dataset is able to significantly improve the performance of a pre-trained language model. The evaluation metrics combined with practical examples and an in-depth analysis regarding the extractiveness and compression provide a strong indication for answering the first research question. The findings from the conducted experiments furthermore outlined gaps and potential errors when performing summarization. These errors were partly due to qualitative shortcomings within the datasets as well as due to the model's behavior. The manually conducted investigation regarding these errors answers the research question that shortcomings and gaps are indeed structured. The errors include factual errors that were found in the generated summaries, regardless of the datasets, while qualitative issues within the data were mainly found in the German dataset.
Lastly, to answer the third research question, it becomes visible that contemporary language models, here exemplified with BART (Lewis et al., 2020), still comprise shortcomings such as factual correctness as drawn in Section 7.2.4 or syntactical and grammatical mistakes as described in Table 6.7.
Concludingly, this work provides insightful information. Firstly, it demonstrates the process of fine-tuning pre-trained language models in general towards multi-document summarization and, in particular, their adaptability to languages beyond English. Secondly, it outlines potential errors and shortcomings that can be put into use for future improvement or development of summarization systems.

8.3. Future Work

While this work examines the applicability of pre-trained language models towards multi-document summarization of German textual information, future work in this area can be manifold.
A first approach could be to proceed with the task of multi-document summarization and study how a model such as BART would behave when fed with an input that was preprocessed e.g. through classifying multiple inputs. Hokamp et al. (2020) provides an initial approach in this area by applying a decoding method to ensemble the output of multiple instances of the same model to different inputs. A similar direction could investigate the performance of summarizing documents in a cluster to extract salient

information before feeding the model with this information. Considering that this work outlined the MDS approach, future work could also include experiments towards SDS with the BART model. Concerning multi-linguality, future work could further be extended towards the applicability of BART on the summarization task in a multilingual approach. Similar work was conducted with the mBART model that examines multilingual pre-training towards neural machine translation and was introduced by Liu et al. (2020).
Finally, there still is a scope to investigate the performance of pre-trained language models in other languages, as this work only covered English and German language. While this work initially examines the applicability of the BART model and its potential errors, further investigations can continue this work by quantitatively analyse the revealed errors and patterns described throughout this thesis. Additionally, new experiments can be conducted that take into account previously introduced pre-trained language models such as PEGASUS (Zhang et al., 2019) and fine-tune the model with available German datasets.

List of Figures

List of Tables

Bibliography

Aries, A., Hidouci, W. K., et al. (2019). Automatic text summarization: What has been done and what has to be done. *arXiv preprint arXiv:1904.00688*.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR, Conference Track Proceedings*, San Diego, California, USA.

Barzilay, R. and McKeown, K. R. (2005). Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Benikova, D., Mieskes, M., Meyer, C. M., and Gurevych, I. (2016). Bridging the gap between extractive and abstractive summaries: Creation and evaluation of coherent extracts from heterogeneous sources. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1039–1050, Osaka, Japan. The COLING 2016 Organizing Committee.

Bing, L., Li, P., Liao, Y., Lam, W., Guo, W., and Passonneau, R. (2015). Abstractive multi-document summarization via phrase selection and merging. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1587–1597, Beijing, China. Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Carbonell, J. and Goldstein, J. (1998). The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, New York, USA. Association for Computing Machinery.

Celikyilmaz, A., Bosselut, A., He, X., and Choi, Y. (2018). Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, page 310–318, USA. Association for Computational Linguistics.

Cheng, J. and Lapata, M. (2016). Neural Summarization by Extracting Sentences and Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Conroy, J. M., Schlesinger, J. D., and O'Leary, D. P. (2006). Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 152–159, Sydney, Australia. Association for Computational Linguistics.

Dantzig, G. B. and Thapa, M. N. (1997). *Linear Programming 1: Introduction*. Springer-Verlag, Berlin, Heidelberg.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short*

Papers), pages 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285.

Erkan, G. and Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479.

Fabbri, A., Li, I., She, T., Li, S., and Radev, D. (2019). Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., and Radev, D. (2020). SummEval: Re-evaluating summarization evaluation. *arXiv preprint arXiv:2007.12626*.

Fernández, S., Graves, A., and Schmidhuber, J. (2007). An application of recurrent neural networks to discriminative keyword spotting. In *Proceedings of the 17th International Conference on Artificial Neural Networks*, ICANN'07, page 220–229, Berlin, Heidelberg. Springer-Verlag.

Filippova, K. and Strube, M. (2008). Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 25–32, Salt Fork, Ohio, USA. Association for Computational Linguistics.

Ganesan, K. (2018). ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks. *arXiv e-prints*, page arXiv:1803.01937.

Gehrmann, S., Deng, Y., and Rush, A. (2018). Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Gers, F. A. and Schmidhuber, J. (2000). Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 3, pages 189–194, Como, Italy.

Goldstein, J., Mittal, V., Carbonell, J., and Kantrowitz, M. (2000). Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization - Volume 4*, NAACL-ANLP-AutoSum '00, pages 40–48, Seattle, Washington, USA. Association for Computational Linguistics.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232.

Grusky, M., Naaman, M., and Artzi, Y. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Haghighi, A. and Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Boulder, Colorado, USA. Association for Computational Linguistics.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1693–1701, Cambridge, Massachusetts, USA. MIT Press.

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

Hokamp, C., Ghalandari, D. G., Pham, N. T., and Glover, J. (2020). Dyne: Dynamic ensemble decoding for multi-document summarization. *arXiv preprint arXiv:2006.08748*.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Compu-*

tational Linguistics (Volume 1: Long Papers), pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Jing, H. (2002). Using hidden Markov modeling to decompose human-written summaries. *Computational Linguistics*, 28(4):527–543.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA.

Jurafsky, D. and Martin, J. H. (2019). Speech and Language Processing (3rd Edition draft). `https://web.stanford.edu/~jurafsky/slp3/`.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, San Diego, California, USA.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, pages 181–184 vol.1.

Kupiec, J., Pedersen, J. O., and Chen, F. (1995). A trainable document summarizer. In Fox, E. A., Ingwersen, P., and Fidel, R., editors, *SIGIR'95, Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (Special Issue of the SIGIR Forum)*, pages 68–73, Seattle, Washington, USA. ACM Press.

Lebanoff, L., Song, K., and Liu, F. (2018). Adapting the Neural Encoder-Decoder Framework from Single to Multi-Document Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (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*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating Wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings*, Vancouver, British Columbia, Canada. OpenReview.net.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation.

Liu, Y. and Lapata, M. (2019). Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019a). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, Y., Titov, I., and Lapata, M. (2019b). Single document summarization as tree induction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1745–1755, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, page 159.

Mani, I. and Bloedorn, E. (1999). Summarizing similarities and differences among related documents. *Inf. Retr.*, 1(1-2):35–67.

Mani, I. and Maybury, M. T. (2001). Automatic summarization. In *Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Companion Volume to the Proceedings of the Conference: Proceedings of the Student Research Workshop and Tutorial Abstracts*, page 5, Toulouse, France. CNRS.

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

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.

McKeown, K. and Radev, D. R. (1995). Generating summaries of multiple news articles. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, page 74–82, New York, New York, USA. Association for Computing Machinery.

McKeown, K. R., Klavans, J., Hatzivassiloglou, V., Barzilay, R., and Eskin, E. (1999). Towards multidocument summarization by reformulation: Progress and prospects. In Hendler, J. and Subramanian, D., editors, *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, 1999*, pages 453–460, Orlando, Florida, USA. AAAI Press / The MIT Press.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048, Makuhari, Chiba, Japan. ISCA.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, New York, USA. Curran Associates Inc.

Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA.

Nallapati, R., Zhou, B., dos Santos, C., GuÌ‡lçehre, Ç., and Xiang, B. (2016). Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Nenkova, A. and Passonneau, R. (2004). Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Owczarzak, K. and Dang, H. T. (2011). Overview of the TAC 2011 Summarization Track: Guided Task and AESOP Task. In *Proceedings of the Fourth Text Analysis Conference*, Gaithersburg, Maryland, USA.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Paul, O. and James, Y. (2004). An Introduction to DUC-2004. In *Proceedings of the 4th Document Understanding Conference*, Boston, Massachusetts, USA. National Institute of Standards and Technology.

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

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (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, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Radev, D. R., Jing, H., and Budzikowska, M. (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*.

Radev, D. R. and McKeown, K. R. (1998). Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.

Radev, D. R. and Tam, D. (2003). Summarization evaluation using relative utility. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, page 508–511, New York, New York, USA. Association for Computing Machinery.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ramachandran, P., Liu, P., and Le, Q. (2017). Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark. Association for Computational Linguistics.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Ruder, S. (2020). Why You Should Do NLP Beyond English. `http://ruder.io/nlp-beyond-english`.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). *Learning Internal Representations by Error Propagation*, page 318–362. MIT Press, Cambridge, MA, USA.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Schluter, N. (2017). The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain. Association for Computational Linguistics.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shen, C. and Li, T. (2010). Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, page 984–992, USA. Association for Computational Linguistics.

Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., and Wang, H. (2020). ERNIE 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, Massachussetts, USA. MIT Press.

Tan, J., Wan, X., and Xiao, J. (2017). Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada. Association for Computational Linguistics.

Teufel, S. and van Halteren, H. (2004). Evaluating information content by factoid analysis: Human annotation and stability. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 419–426, Barcelona, Spain. Association for Computational Linguistics.

Torres-Moreno, J. (2014). *Automatic Text Summarization*. Wiley.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Vinyals, O., Fortunato, M., and Jaitly, N. (2015). Pointer networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Williams, J. W. J. (1964). Algorithms. *Commun. ACM*, 7(6):347–349.

Witte, R. and Bergler, S. (2007). Next-generation summarization: Contrastive, focused, and update summaries. In *International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*, Borovets, Bulgaria.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Yasunaga, M., Zhang, R., Meelu, K., Pareek, A., Srinivasan, K., and Radev, D. (2017). Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada. Association for Computational Linguistics.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*.

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., and Huang, X. (2020). Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., and Zhao, T. (2018). Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia. Association for Computational Linguistics.

Zopf, M. (2018). Auto-hMDS: Automatic Construction of a Large Heterogeneous Multilingual Multi-Document Summarization Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3228–3233, Miyazaki, Japan. European Language Resources Association (ELRA).

A. Appendices

A.1. Examples CNN/Dailymail

Table A.1.: Generated summaries by BART on CNN/DailyMail (sampled)

CNN/DailyMail	
Document (ID #6)	(CNN)A Duke student has admitted to hanging a noose made of rope from a tree near a student union, university officials said Thursday. The prestigious private school didn't identify the student, citing federal privacy laws. In a news release, it said the student was no longer on campus and will face student conduct review. The student was identified during an investigation by campus police and the office of student affairs and admitted to placing the noose on the tree early Wednesday, the university said. Officials are still trying to determine if other people were involved. Criminal investigations into the incident are ongoing as well. Students and faculty members marched Wednesday afternoon chanting "We are not afraid. We stand together," after pictures of the noose were passed around on social media. At a forum held on the steps of Duke Chapel, close to where the noose was discovered at 2 a.m., hundreds of people gathered. "You came here for the reason that you want to say with me, 'This is no Duke we will accept. This is no Duke we want. This is not the Duke we're here to experience. And this is not the Duke we're here to create,' " Duke President Richard Brodhead told the crowd. The incident is one of several recent racist events to affect college students. Last month a fraternity at the University of Oklahoma had its charter removed after a video surfaced showing members using the N-word and referring to lynching in a chant. Two students were expelled. In February, a noose was hung around the neck of a statue of a famous civil rights figure at the University of Mississippi...

Gold	Student is no longer on Duke University campus and will face disciplinary review. School officials identified student during investigation and the person admitted to hanging the noose, Duke says. The noose, made of rope, was discovered on campus about 2 a.m.
Model	The student was identified during an investigation by campus police and the office of student affairs. He admitted to placing the noose on the tree early Wednesday, the university said. Officials are still trying to determine if other people were involved. The incident is one of several recent racist events to affect college students.
ROUGE	R-1 = 35.18, R-2 = 5.66, R-L = 22.22
Document (ID #2911)	A 10-year-old Huddersfield Town supporter has been removed from his role as a mascot for Reading in their FA Cup semi-final against Arsenal on Saturday after a video emerged of him saying he hopes The Gunners win. Ryan Dearnley won the chance to walk out at Wembley when The Royals beat Huddersfield in the third round but footage of him backing Arsenal in the Huddersfield Examiner sparked a furious response from Reading supporters. A poll in getreading revealed 85% readers felt he should not lead Steve Clarke's side out at Wembley. A Huddersfield Town fan will no longer be a mascot for Reading in the FA Cup semi-final against Arsenal . The young supporter had said he hoped Arsenal beat Reading, prompting outrage from Royals fans . Championship side Reading eliminated Huddersfield in the third round on their way to meeting Arsenal . The Football Association reacted quickly, moving young Ryan away from the semi-final to avoid a nasty reception from fans, instead offering him the opportunity to be a mascot for England. 'Following Ryan's interview in the local media, and with agreement from his family, it was decided to move his prize over to an England mascot place later this year,' an FA spokesman is quoted as saying in getreading. The FA reportedly offered Ryan two tickets to watch the semi-final in the Arsenal end but he refused, preferring to watch Huddersfield take on Derby in the Championship.
Gold	Ryan Dearnley won the chance to be a mascot in the FA Cup semi-final. He was due to be one for Reading before saying he wanted Arsenal to win. Fan outrage caused The FA to move his prize to an England game instead.

Model	Ryan Dearnley won the chance to walk out at Wembley when Reading beat Huddersfield in the third round. The 10-year-old had said he hoped Arsenal beat Reading, prompting outrage from Royals fans. Football Association moved him away from the semi-final to avoid a nasty reception from fans. He has been offered the opportunity to be a mascot for England.
ROUGE	R-1 = 40.32, R-2 = 16.39, R-L = 24.19
Document (ID #8075)	This is the moment when a family of ducks followed staff back to their office and made themselves at home after staff helped them across a busy road. Employees at the recruitment company had spotted the mother and ten ducklings trying to cross the busy high street in Sutton Coldfield, West Midlands, and dashed to help them across. But they didn't bargain for the ducks following them back to the office and staying for the morning, wondering the corridors and drinking from water bowls. Scroll down for video . A mother and ten ducklings waddled into an office in the West Midlands and drank from the water bowl . They ushered the family of ducks out of a back door to safety and they then disappeared into a nearby park. Resourcer Melissa Patrick, 18, who filmed the amusing episode on her mobile phone said it had the whole office in stitches. She added: 'They were walking along the high street and we stopped the traffic so they didn't get hurt. Making themselves at home: Ducks wonder through the corridors in the office after staff helped them across the road on the busy Sutton Coldfield high street. The ducks followed them back for a visit . Midas recruitment's office and the busy street that the ducks crossed en-route . 'After we helped them cross, the mum and her chicks headed into the office and they stayed for about an hour. 'I've never known anything like it, it's so weird. They were just wandering around the office corridors. 'Our director was here and he thought it was really funny. Everyone was having a laugh. 'We gave them some water to drink and let them out the back of the office. They were all fine.' Follow the leader: The ducklings follow their mother around the office of the West Midlands firm.

Gold	Duck and ten ducklings followed staff back to an office in Sutton Coldfield. Visit was filmed by a member of staff on her mobile phone. Employee said even company director found the whole thing hilarious.
Model	A mother and ten ducklings waddled into an office in the West Midlands and drank from the water bowl. Employees at the recruitment company had spotted the mother and 10 ducklings trying to cross the busy high street in Sutton Coldfield, West Midlands. They ushered the family of ducks out of a back door to safety and they then disappeared into a nearby park. Resourcer Melissa Patrick, 18, filmed the amusing episode on her mobile phone.
ROUGE	R-1 = 27.63, R-2 = 12.00, R-L = 17.10

A.2. Examples Multi-News

Table A.2.: Generated summaries by BART on Multi-News (sampled)

	Multi-News
Document (ID #1824)	Charlie Sheen Enrages 9/11 Conspiracy Group A group of 9/11 conspiracy theorists are threatening to protest in front of Charlie Sheen 's live shows ... all because they feel the actor has betrayed them.In case you forgot, Sheen was VERY outspoken about his belief that 9/11 was a conspiracy – and famously remarked how the collapse of the World Trade Center buildings looked like a "controlled demolition."Now, Mark Dice, a prominent member of The 9/11 Truth Movement, tells us his fellow conspiracy theorists are pissed – because they feel Charlie has abandoned their cause.Mark tells us Charlie should be "asking hard questions about what happened on 9/11 and the resulting wars ... not bragging about smoking crack and sleeping with hookers."Mark says he's reached out to Charlie – to no avail – but he tells us if Charlie doesn't address the issue soon, he and several theorists will protest in front of Sheen's upcoming shows.Sorry, the page you requested was....

Gold	Weird news from the land of Charlie Sheen: The troubled actor has managed to irk a group of 9/11 truthers, which is now threatening to protest in front of his live shows. Members of the 9/11 Truth Movement, who believe 9/11 was a conspiracy, feel Sheen has betrayed them. Sheen was once very vocal about his belief in the truther movement (he said the World Trade Center collapse looked like a "controlled demolition"), but instead of "asking hard questions about what happened on 9/11 and the resulting wars," a member tells TMZ, Sheen has been "bragging about smoking crack and sleeping with hookers." In other Sheen-related news from TMZ, ex Brooke Mueller is back in rehab after a weeklong binge, having refused a drug test and engaged in some other strange behavior—click here for more.
Model	A group of 9/11 conspiracy theorists are threatening to protest in front of Charlie Sheen's live shows—all because they feel the actor has betrayed them. In case you forgot, Sheen was VERY outspoken about his belief that9/11 was a conspiracy—and famously remarked how the collapse of the World Trade Center buildings looked like a "controlled demolition." Now, Mark Dice, a prominent member of The 9/ 11 Truth Movement, tells us his fellow conspiracy theorists is pissed—because they feel Charlie has abandoned their cause. Mark tells us Charlie should be "asking hard questions about what happened on 9/9 and the resulting wars ... not bragging about smoking crack and sleeping with hookers."
ROUGE	R-1 = 67.52, R-2 = 43.96, R-L = 46.15
Document (ID #3987)	More than two years after China's lunar rover got away to a wonky start on the surface of the moon, the country's National Space Administration has released the stunning hi-def pictures it took. On December 14, 2013, Yutu, or "Jade Rabbit" became the third unmanned spacecraft to land on the moon. But within a month, the rover started having some problems preparing for the lunar nights, and was losing its precious solar power supplies. It stopped moving on January 25, 42 days into a three-month mission. By mid-February ...

Gold	– Its mission had a rocky start, but China's Jade Rabbit moon rover is now sending back some nifty high-resolution images of the lunar surface, reports CNN. China has made them available to download here, though Business Insider finds the process "frustratingly complicated." Luckily, it adds, Emily Lakdawalla of the Planetary Society has done much of the heavy lifting, so head here if you'd like to see many, many more.
Model	– China has released hundreds of high-resolution photos taken by its Chang'e-3 lunar lander and rover, showing the moon's surface in vivid detail. The country's National Space Administration made the images, video clips, and scientific data available on its website in a rare show of openness for the country's usually secretive space program, CNN reports. China sent its first unmanned lunar probe, the Yutu, or "Jade Rabbit," to the moon in 2013, becoming only the third nation after the United States and Russia to land on the moon. But within a month, the rover started having some problems preparing for the lunar nights, and was losing its precious solar power supplies.
ROUGE	R-1 = 25.86, R-2 = 3.47, R-L = 11.20
Document (ID #5125)	– Let this be a lesson to you: When you're in court before the judge, and you doff your cap in deference, make sure your cocaine does not fall out of your hat and onto the floor where both the judge and the cops can see it. And it would help if you weren't in court on a separate felony drug charge in the first place. Juan Jose Vidrio Bibriesca, 43, was one of three defendants at the podium before Eagle County Court Judge Katharine Sullivan. Bibriesca was in Judge Sullivan's company for a bond violation on a separate felony drug charge. Apparently, Bibriesca was raised to be polite. When he reached the podium in the middle of Sullivan's courtroom, he removed his hat, as a gentleman would, and held it behind his back in both hands. As he swung his hat behind his back, a small square of folded paper fell from it and landed on the floor behind him. A local law enforcement ...

Gold	– Authorities say a man who was in a Colorado court for violating his bond on a drug charge is in even more trouble after a wad of cocaine fell from his hat while he was in front of the judge. The Vail Daily reported Wednesday that 43-year-old Juan Jose Vidrio Bibriesca was standing next to two other defendants at an Eagle County District Court podium when he took his hat off and a square of folded paper fell out. A police officer watched the paper filled with cocaine fall to the floor, reports the AP, and after reviewing surveillance footage, authorities determined it fell from Bibriesca's hat. Bibriesca was then walked to the county jail, and now faces new charges of narcotics possession and another bond violation. Booking documents don't indicate if he has hired an attorney. The Vail Daily notes that Bibriesca is in the country illegally, meaning his legal woes aren't over.
Model	– Let this be a lesson to you: When you're in court before the judge, and you doff your cap in deference, make sure your cocaine does not fall out of your hat and onto the floor where both the judge and the cops can see it. And it would help if you weren't in court on a separate felony drug charge in the first place. Juan Jose Vidrio Bibriesca, 43, was one of three defendants at the podium before Eagle County Court Judge Katharine Sullivan for a bond violation on a drug charge. When he reached the podium in the middle of Sullivan's courtroom, he removed his hat, as a gentleman would, and held it behind his back.
ROUGE	R-1 = 50.41, R-2 = 11.66, R-L = 22.31

A.3. Examples Auto-hMDS

Table A.3.: Generated summaries by BART on auto-hMDS (sampled)

	auto-hMDS
Document (ID #80)	Agnes Bernauer († 1435) - Die unerwünschte Schwiegertochter Die traurige Liebesromanze zwischen Agnes Bernauer (Abb. 4), einer Augsburger Baderstochter, und Albrecht III., dem zukünftigen Herzog von Bayern-München, begann wahrscheinlich im Februar 1428. Zu dieser Zeit um Fasching herum scheinen sich die beiden, zum ersten Male begegnet zu sein. Vielleicht hatte der damals 27-jährige Albrecht III. – wie so viele seiner Zeitgenossen – einen Teil der ausgelassenen Faschingszeit wegen der kalten Witterung in den beliebten Badestuben verbracht, die in den Städten in fast jeder Straße zu finden waren. Außer den Juden war jedem der Eintritt gegen eine kleine Gebühr erlaubt. Hier konnte man sich dann – je nach Wunsch – vom Bademeister rasieren, schröpfen oder zur Ader lassen, oder die Bademägde, die an ihren trägerlosen, weißen, dünnen Kleidern, deren Ausschnitt im Rücken fast bis zum Po reichte, und die an ihren gestrickten Haarnetzen zu erkennen waren, um eine Massage bitten oder sich von ihnen von oben bis...
Gold	Agnes Bernauer (* um 1410 wohl in Augsburg; † 12. Oktober 1435 bei Straubing) war die Geliebte und vielleicht auch die erste Ehefrau des bayerischen Herzogs Albrecht III. Durch diese nicht standesgemäße Verbindung geriet Albrecht in Konflikt mit seinem Vater Ernst, der Agnes Bernauer 1435 in der Donau ertränken ließ. Ihr Leben und Sterben wurde in zahlreichen literarischen Werken verarbeitet, zu den bekanntesten zählen Friedrich Hebbels gleichnamiges Trauerspiel Agnes Bernauer und Die Bernauerin von Carl Orff. Alle vier Jahre finden in Straubing und in Vohburg Agnes-Bernauer-Festspiele statt, bei denen die Geschichte von Laienschauspielern in Szene gesetzt wird.

Model	Die traurige Liebesromanze zwischen Agnes Bernauer (Abb. 4), einer Augsburger Baderstochter, und Albrecht III., dem zukünftigen Herzog von Bayern-München, begann wahrscheinlich im Februar 1428. Zu dieser Zeit um Fasching herum scheinen sich die beiden, zum ersten Male begegnet zu sein. Vielleicht hatte Agnes ihrem Vater in seinem Geschäft geholfen, und viel Leichtathleten das Glück, von ihr bedient zu werden. Über Agnes selbst wissen wir nur sehr wenig. Ihre Kindheit, ihre Gedanken und ihren Charaktereigenschaften bleiben im Dunkeln. Das einzige, was wir über sie sagen können, ist, daß sie sehr schön und von sehr zarter Statur ist.
ROUGE	R-1 = 22.33, R-2 = 1.96, R-L = 13.59
Document (ID #134)	Mit der Lufthansa „Business Class“ Edition bringt BAG TO LIFE neue Standards in den Reisealltag. mehr erfahren Flugzeugnostalgie im Taschenformat Die bordbar design GmbH aus Köln hatten eine einfache Idee mit einer großen Vision. Angetrieben vom Drang ausrangierte Objekte in etwas Neues zu verwandeln und... mehr erfahren Die limitierte A340 Series stammt aus der DNA „des fliegenden Botschafter der Swiss“, der... mehr Produktinformationen "Swiss Airbus A340 HB-JMK – weiss" Die limitierte A340 Series stammt aus der DNA „des fliegenden Botschafter der Swiss“, der A340 HB-JMK, auch unter dem Namen Aarau bekannt. Der Airbus A340 zählte zu den grössten Maschinen der Swiss Flotte und flog in seinen 10 Dienstjahren für Swiss durch die halbe Welt. Zu seinen Destinationen zählten unteranderem Sao Paulo, Tokio und Shanghai. Nach 20 Jahren über den Wolken beginnt für sie eine neue Geschichte als dein Aviationtag. A340 Series – gemacht für alle Wanderlust-Sehnsüchtigen...

Gold	Der Airbus A340 ist ein vierstrahliges Großraumflugzeug des europäischen Flugzeugherstellers Airbus. Der Erstflug dieses Langstreckenflugzeugs fand am 25. Oktober 1991 in der Version A340-300 statt. Der Tiefdecker war das erste Verkehrsflugzeug von Airbus mit vier Triebwerken und das erste zivile Großraumflugzeug weltweit mit digitaler Fly-by-wire-Steuerung. Der Airbus A340 wurde parallel mit dem Schwestertyp Airbus A330 entwickelt, mit dem er viele Merkmale und Baugruppen gemeinsam hat. Die Endmontage der A340 fand in Toulouse auf einer gemeinsamen Taktstraße zusammen mit der A330 statt. Bis Mai 2013 wurden alle 377 bestellten Flugzeuge dieses Typs ausgeliefert, von denen Ende Mai 2016 noch 295 im Dienst waren. Am 10. November 2011 gab Airbus bekannt, die Produktion der vierstrahligen A340 komplett einzustellen. Das Nachfolgemodell ist der zweistrahlige Airbus A350. Bis Dezember 2015 standen bei Airbus 22 gebrauchte A340, die nach Wegfall der Iran-Sanktionen vom Iran gekauft werden konnten.
Model	Der Airbus A340-300 ist ein vierstrahliges Passagierflugzeug des europäischen Flugzugherstellers Airbus für die Langstrecke. Mit einer Länge von 63,68 m bietet die A340 zwischen 295 bis maximal 440 Passagieren Platz und verfügt über eine maximale Reichweite bis 12.500 Kilometer. Der Tiefdecker war das erste Verkehrsflugzenug von Airbus mit vier Triebwerken und das der erste zivile Großraumflüsse weltweit mit digitaler Fly-by-wire-Steuerung. Er gilt als eines der wichtigsten Flugzeugs der Welt und wurde in seinen 10 Dienstjahren für Swiss durch die halbe Welt. Zu seinen Destinationen zählten unteranderem Sao Paulo, Tokio und Shanghai.
ROUGE	R-1 = 46.67, R-2 = 21.15, R-L = 33.33
Document	

(ID #172)	Die Reichsabtei Salem war ein Kloster des Zisterzienserordens und eine der wohlhabendsten und bedeutendsten reichsunmittelbaren Abteien des Bodenseeraums. Das 1137/1138 gegründete Kloster konnte im späten Mittelalter seine Privilegien und die Stellung innerhalb des Ordens weit ausbauen. Im 17. Jahrhundert von Kriegen bedrängt und durch einen Brand fast vollständig zerstört, erlebte es im 18. Jahrhundert seine zweite Blütezeit als Zentrum des südwestdeutschen Rokoko. Die weitläufige barocke Klosteranlage (erbaut 1697–1706 von Franz Beer) mit dem hochgotischen Salemer Münster (ca. 1285–1414) ging...
Gold	Die Reichsabtei Salem in der heutigen Gemeinde Salem im Linzgau (Baden-Württemberg) war ein Kloster des Zisterzienserordens und eine der wohlhabendsten und bedeutendsten reichsunmittelbaren Abteien des Bodenseeraums. Das 1137/1138 gegründete Kloster konnte im späten Mittelalter seine Privilegien und die Stellung innerhalb des Ordens weit ausbauen. Im 17. Jahrhundert von Kriegen bedrängt und durch einen Brand fast vollständig zerstört, erlebte es im 18. Jahrhundert seine zweite Blütezeit als Zentrum des südwestdeutschen Rokoko mit dem Bau der Wallfahrtskirche Birnau und der Gründung der ersten Sparkasse Deutschlands. Die weitläufige barocke Klosteranlage (erbaut 1697–1706 von Franz Beer) mit dem hochgotischen Salemer Münster (ca. 1285–1414) ging 1802 durch Säkularisation in den Besitz der Markgrafen von Baden über. Seither trägt die Anlage den Namen „Schloss Salem“ und dient als Wohnsitz der markgräflichen Familie sowie seit 1920 als Sitz des Internats Schule Schloss Salem. Im Frühjahr 2009 veräußerte das Haus Baden den größten Teil der Anlage an das Land Baden-Württemberg.
Model	Die Reichsabtei Salem war ein Kloster des Zisterzienserordens und eine der wohlhabendsten und bedeutendsten reichsunmittelbaren Abteien des Bodenseeraums. Das 1137/1138 gegründete Kloster konnte im späten Mittelalter seine Privilegien und die Stellung innerhalb des Ordens weit ausbauen. Im 17. Jahrhundert von Kriegen bedrängt und durch einen Brand fast vollständig zerstört, erlebte es im 18. Jahhundert seine zweite Blütezeit als Zentrum des südwestdeutschen Rokoko. Die weitläufige barocke Klosteranlage (erbaut 1697–1706 von Franz Beer) mit dem hochgotischen Salemer Münster (ca. 1285–1414) ging 1802 durch Säkularisation in den Besitz der Markgrafen von Baden über.

ROUGE	R-1 = 78.64, R-2 = 66.67, R-L = 78.64

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudiengang IT-Management und -Consulting selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel - insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen - benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Hemmingen, 18.12.2020

Ort, Datum

Timo Johner

Unterschrift