TECHNICAL UNIVERSITY OF MUNICH

TUM Data Innovation Lab

Using NLP for Adaptive Fact Extraction and Text Summarization

Authors	Benedikt Anselment, Afag Hasanli, Anelia Petrova, Sergii Poluetkov, Marc Schneider
Mentors	Moritz Beutter, Abraham Duplaa faktual UG
Co-Mentor	Tim Fuchs (Department of Mathematics)
Project Lead	Dr. Ricardo Acevedo Cabra (Department of Mathematics)
Supervisor	Prof. Dr. Massimo Fornasier (Department of Mathematics)

Jul 2020

Abstract

The TUM Data Innovation Lab project is focused on automated summarization techniques. The project's concrete goal is to analyze the proceedings in the federal parliament of Germany, the Bundestag. The German parliament is an ideal example for the struggle journalists face when they conduct research.
Automated text summarization is rapidly gaining relevance in the NLP-landscape. We focus on abstractive summarization due to the necessity of creating fluent text that utilizes the input's semantic context more effectively.

This abstract was automatically generated by distil-BART, our best performing model for abstractive summarization.

Acknowledgements

We would like to thank our mentors Abraham and Moritz, for providing this great project and going well beyond the expectations set for mentors in assisting our team. We are also grateful to our co-mentor Tim for his invaluable support throughout the project, as well as to Prof. Fornasier and Dr. Cabra for enabling the Data Innovation Lab. We are extremely thankful to Meinolf Ellers and Roland Freund for their valuable insights into political journalism. Finally, we thank the Leibniz-Rechenzentrum for providing us access to their technical infrastructure.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Due to the rapid technological development which took place in the past three decades, humanity nowadays has an unprecedented and ever-growing amount of textual information at its disposal. To put it into perspective: "Every two days, humans produce more textual information than the combined output of humanity from the dawn of recorded history up through the year 2003" [12]. This blessing of information being only one click away, can, however, easily become a curse if the right tools for processing and analyzing huge amounts of data are not available.
One common example for information overload becoming such a curse is the widely-discussed issue of *fake-news*[1]. For a majority of people it becomes increasingly difficult to distinguish fake from real news [23]. This development affects multiple aspects of our society, but arguably the most worrisome impact can be documented in politics, where fake news creates confusion and bitter conflicts.
The only productive countermeasure which can stop the issue of fake news becoming even more severe than it already is, is an independent and free press. To guarantee the continued existence of such an institution in the 21^{st} century, journalists need to be provided with tools which enable them to scan and analyze the sheer unbearable amount of information they face on a daily basis during their research efforts. One of the fundamental properties of such a tool is summarization capability, therefore the ability to outline critical arguments of a given input document in a precise and concise manner. In order to contribute together with our close partner *faktual* a small step on the way to create such a tool, our TUM Data Innovation Lab project is focused on automated summarization techniques.

1.2 Project Goals

Our project's concrete goal is to analyze the proceedings in the federal parliament of Germany, the *Bundestag*. The two main types of textual data provided by the German parliament are the protocols of the parliamentary sessions and the printed matters (ger. Bundestagsdrucksachen)[2]. Records of these documents dating back until 1949 are publicly available on the website of the Bundestag [3]. However, there is no fast and easy way to analyze their content. Therefore, the German parliament is an example for the struggle journalists face when they conduct research and thus provides the ideal use case for our project.
The remainder of this report is structured as follows: We first describe our acquisition, preprocessing, and first data exploration of the parliamentary datasets in section 2. Chapter 3 presents our summarization approach and an evaluation of several different models. We then explore potential use cases for our summarization pipeline in section 4. Finally, we outline the results of our project and propose future improvements in section 5.

[1] *Facebook* records 80-200 million engagements of their users with fake-news per month [1].
[2] For brevity, we will refer to them as *Drucksachen* throughout this document.
[3] `https://www.bundestag.de/services/opendata`

2 Data Exploration

2.1 Data Acquisition

At the beginning of the project faktual kindly provided us with all session protocols and Drucksachen from the 14^{th} to 18^{th} legislative period of the German parliament in XML format. In addition to that we were supplied with 152 session protocols from the 19^{th} legislative period, which is ongoing at the time of writing. Table 1 provides an overview of the used data in its entirety.

Legislative period	Years	Protocols	Drucksachen
14	1998-2002	253	10 005
15	2002-2005	187	6 016
16	2005-2009	233	14 163
17	2009-2013	253	14 839
18	2013-2017	245	13 706
19	2017-	>152	>21 069

Table 1: Overview of the available textual data for the legislative periods 14-19 of the Bundestag.

As the parliament keeps working, new session protocols and Drucksachen are released on a regular basis. In order to acquire this latest data, we run appropriate interfaces for both session protocols and Drucksachen. Especially the newest Drucksachen are posing a challenge since they are published in PDF-format. We use `tika-python`, an interface for the Apache Tika parser, to transform them into plain text.

2.2 Data Pipeline

2.2.1 XML-Parser

At the beginning of the 19^{th} legislative period, a new document type definition (DTD) was introduced by the Bundestag. This new DTD adds considerably more structure to and enables more robust parsing of the parliamentary speeches. To take full advantage of this development, we map the structure of the newly formatted documents via an XML-parser to Python classes. The corresponding UML diagram is depicted in Figure 1.

2.2.2 Session Protocols

The session protocols from the legislative periods 14 to 18, though provided in XML-format, lack good structure and do not explicitly separate speeches. In order to use this older data, we build a parser, which structures the texts of the older session protocols into the same format as the one used for the session protocols from the 19^{th} period. Our implementation of this parser relies to a large extent on the *Bundestag open-source project*[4].

[4] `https://github.com/bundestag/plpr-scraper`

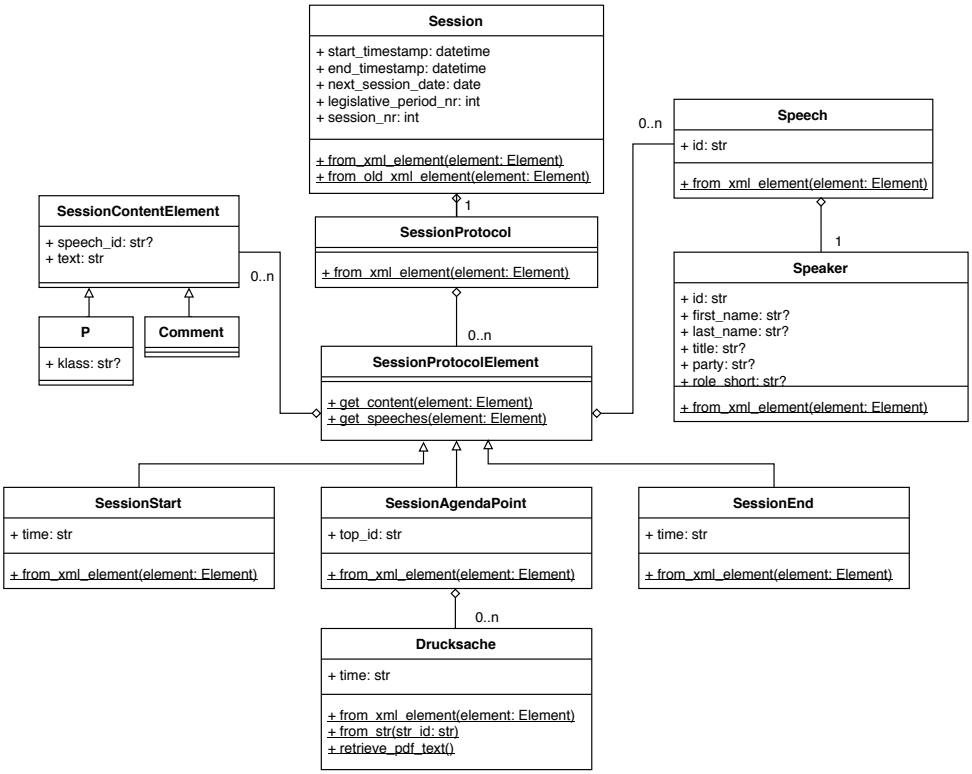

Figure 1: UML diagram of the session protocols.

2.2.3 Data Storage

In order to make our data filterable and searchable, it is stored in *Elasticsearch*, a search engine based on the `Lucene` library. It provides a distributed, multitenant-capable full-text search engine with an HTTP-web interface and schema-free JSON-documents. One of the main advantages of Elasticsearch is its support for full-text search in multiple languages. In addition, while indexing a text, Elasticsearch processes it with an array of language-specific tokenizers, stop word filters, normalizers and stemmers so that also non-exact, but still relevant matches can be found at a later point in time.

2.3 Data Processing

Within the first milestone of our project, we use common natural language processing techniques to gain a better understanding of our dataset. On a basic level, we examine the character, word, and speech distributions among parties. On a more advanced level, we identify the terms, named entities, and topics that represent each legislative period. Finally, we analyze the sentiment of each speech and the reactions of parties to certain speeches.

2.3.1 Character Distribution

The character length distribution of the parliamentary speeches in the 19^{th} period is bi-modal, with peaks around 900 and 4 700 characters per speech. Shorter speeches usually represent questions from the plenum or spontaneous answers, while longer speeches are

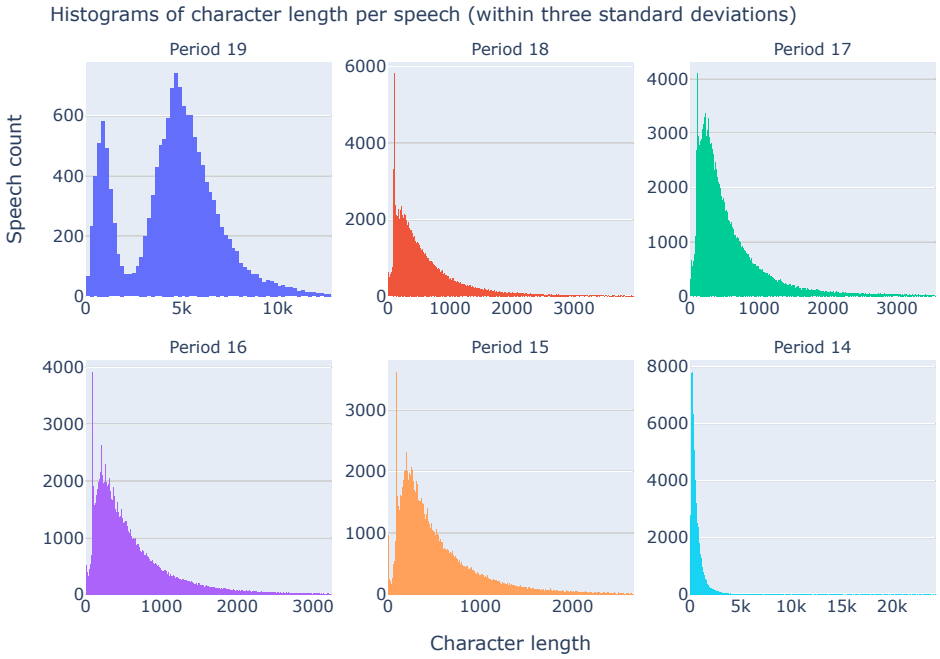

Figure 2: Character length-distribution of speeches from the 14^{th} to the 19^{th} legislative period.

prepared in advance. Figure 2 provides an overview of the character length-distributions for the six legislative periods covered in this report. We can see that although the outliers over three standard deviations are removed, the distributions of the speeches from periods 14 through 18 are right-skewed. This highlights the fact that our parser performs overoptimistic while splitting the speeches when it does not have sufficient information on their ending.
The character lengths per session observe periodicity, with shorter sessions following longer ones and vice versa. Additionally, the word count median of whole sessions stands at 56 000 words for the 19^{th} legislative session.

2.3.2 Frequent Terms

To discover the most salient terms for each legislative period, we use TF-IDF [17], a metric that encourages the frequency of a term within a document but strictly penalizes terms that are common for the entire dataset. Table 2 presents the top performing terms according to their TF-IDF score of each legislative period analyzed in our project.

2.3.3 Named Entity Recognition

We leverage the named entity recognition capabilities of the `spacy` library to identify the most frequently discussed locations, people, and organizations. This allows us, similarly to the frequent term analysis, to determine the most salient entities for each time span. The top five ranking names for the 19^{th} legislative period are available in figure 3.
Throughout the 19^{th} period, the most discussed organization is the *Bundeswehr*, the

Legislative period	Year span	Terms
14	1998–2002	Kuba, Stammzellen, Öcalan
15	2002–2005	Moldau, HIV, Ukraine
16	2005–2009	Cannabis, Roma/Sinti, Sri Lanka
17	2009–2013	Glyphosat, Westsahara, Honig
18	2013–2017	Europol, Isis, Substanzen
19	2017–	Gigawatt, Adoption, LKWs

Table 2: Representative terms for each legislative period according to the TF-IDF metric.

unified armed forces of Germany. Possible events that explain this result are the budget increase of five million euros in 2018 and the discussion of whether the Bundeswehr should withdraw its troops from Afghanistan.

2.3.4 Reactions

Each session protocol notes audible reactions of parliamentary members to the current speech in parentheses. These feature applause, excitement, laughter, as well as shouts and can reveal support or tension within and between parties. Figure 3 demonstrates the distribution of these reactions within the parties in the 19^{th} legislative period with rows indicating the reacting party and columns the party represented by the current speaker. Unsurprisingly, parties reserve positive reactions (applause and excitement) for their own members, while laughter and shouts are directed to more polarizing parties.

2.3.5 Sentiment Analysis

Sentiment analysis uses a collection of methods that quantifies the opinions within a given text. As law proposals are formal documents, there is only limited sentiment to be extracted. Speeches, on the other hand, are highly rhetoric, as the speaker aims to appeal to his or her audience. Although working with text loses valuable information like body language and tone intonation, we can still glean a lot from the use of language. For this purpose, we calculate the sentiment-polarity [20] of each speech using the German version of the `TextBlob` Python library. This metric provides a value in the interval [-1,1], with negative values indicating negative and positive values indicating positive sentiment.
By performing sentiment analysis on the speeches from the 19^{th} period, we can gain a

Ranking	Location	Person	Organization
1	Vereinigte Staaten	Angela Merkel	Bundeswehr
2	Russland	Grigorios Aggelidis	EU
3	Berlin	Stephan Thomae	NATO
4	Türkei	Olaf Scholz	CO
5	Syrien	Hubertus Heil	Vereinten Nationen

Table 3: Most frequently mentioned locations, people, and organizations during the 19^{th} legislative period.

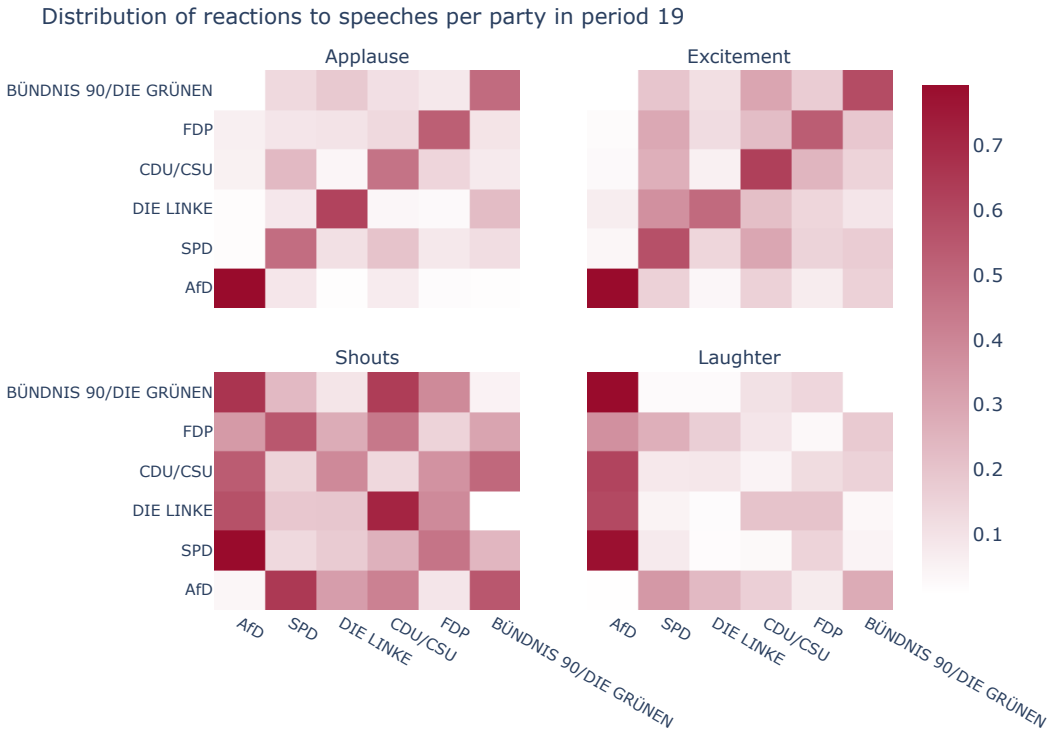

Figure 3: Distribution of the reactions in the Bundestag. The rows indicate the reacting party and the columns the party represented by the current speaker.

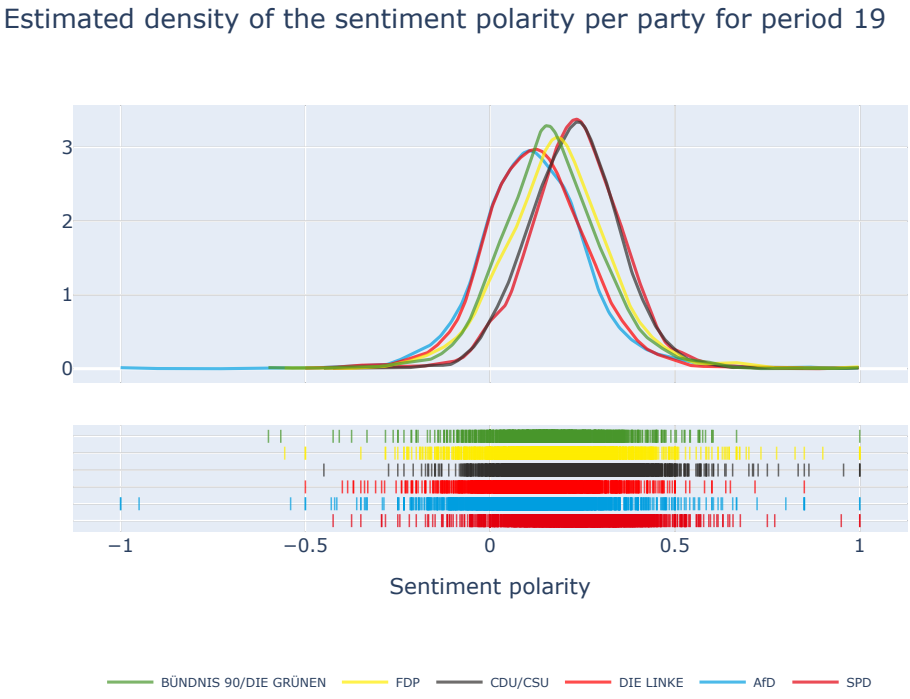

Figure 4: Estimated density of the sentiment-polarity in the 19^{th} legislative period.

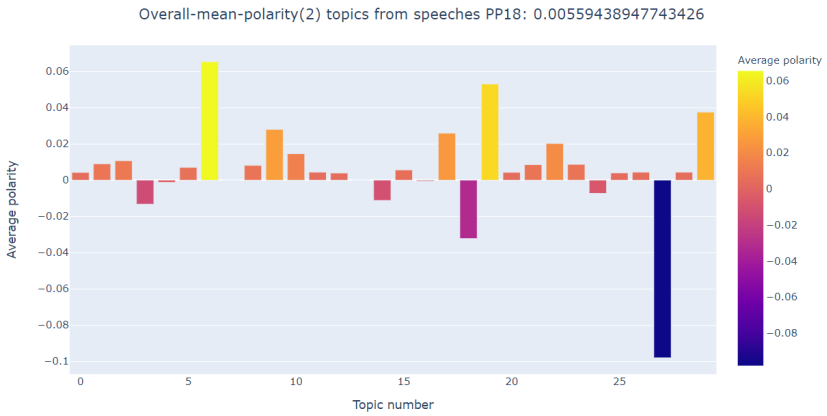

Figure 5: Polarity of retrieved topics for speeches in the parliamentary period 18.

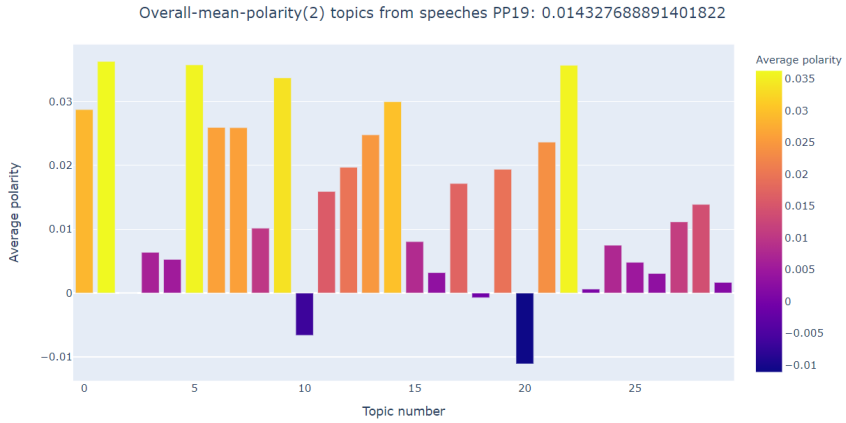

Figure 6: Polarity of retrieved topics for speeches in the parliamentary period 19.

broad overview of the tone each party establishes. In the distribution-plot presented in figure 4 we can see that on average every party exhibits mildly positive sentiment polarity. This is partially explained by the fact that texts predominantly feature neutral terms. Nevertheless, we can determine that the sentiment polarity of the parties outside the government cabinet is lower, especially in the cases of DIE LINKE and AfD. While these results are unsurprising, we can make use of them in our computation pipeline in the prototype of our summarization tool in section 4.

2.3.6 Topic Modelling

A topic in the context of natural language processing is represented by a probability distribution over a given vocabulary. We use *latent Dirichlet allocation (LDA)*, introduced in 2003 by Blei et al. [3], to extract topics from the parliamentary speeches of the 18^{th} and 19^{th} legislative period.
We extract 30 topics from the 18^{th} and 19^{th} legislative periods to analyze the average sentiment of each. Results are reported in Figure 5 and 6. Surprisingly, the average sentiment in period 19 is considerably more positive than the corresponding score in period 18, despite the entry to the Bundestag of the widely populist AfD in 2017.

3 Automated Text Summarization

3.1 Introduction

3.1.1 Extractive and Abstractive Summarization

Automated text summarization is rapidly gaining relevance in the NLP-landscape. The main objective of its corresponding models is to extract the most meaningful content from one (single-document summarization) or several (multi-document summarization) input documents in order to construct a factually accurate and grammatically correct summary without involving human intelligence [15]. We can distinguish two main types of summarization models: extractive and abstractive.
Extractive summarization methods produce summaries by concatenating several sentences (text units) from a given input-text precisely as they occur. The main task of such systems is, therefore, to determine which sentences are meaningful and should be included in a summary [5]. In contrast, abstractive summarization models scan given input documents, compare them with related information in their *memory*[5], and then create a summary by reformulating the key points of the input. Thus, according to Sanjabi (2018) [18], abstractive methods are more challenging to conduct than extractive ones, since an abstractive model is supposed to break the source corpus down to its tokens and regenerate the target sentences out of them. Obtaining meaningful and grammatically correct sentences in an abstractive setting, therefore, demands highly precise and sophisticated models.
Within the scope of our project, we focus on abstractive summarization due to the necessity of creating a fluent text that utilizes the input's semantic context more effectively.

3.1.2 Concepts and Methodology

As a subset of text generation problems, abstractive summarization largely relies on recurrent neural architectures like extended short-term memory networks (LSTMs). This structure has proven useful in capturing long-term dependencies and overcoming the issue of vanishing gradient during training.
The sequence-to-sequence (seq2seq) model architecture addresses the problem of finding the correspondence between source a sequence and an unknown target sequence [19]. It's an end-to-end system composed of two recurrent neural networks: an encoder, which compresses the input text token-by-token into a compact representation and a decoder, which generates text conditioned on the encoder's outputs. The seq2seq model has been successfully applied in machine translation [19], caption generation [22], and abstractive summarization.
In the standard seq2seq model, the last encoder hidden state represents all source information, causing a bottleneck. To handle this problem, the authors of [2] devised the attention mechanism that focuses on a different part of the source at each time step. It achieves this by computing a weighted sum of the encoder states and the current decoder state. One especially valuable property of this mechanism is increasing interpretability, as the resulting attention distribution reveals which part of the input each decoder hidden state focuses on.

[5] We refer to the learned weights of the underlying Neural Network-architecture as *memory*.

Building upon the attention mechanism, the authors of [21] propose the Transformer architecture that eschews recurrent and convolutional structures in favor of a more advanced attention mechanism, called *multi-head attention*. While the Transformer also consists of encoder and decoder components, these are built using blocks containing a multi-head attention sublayer and a fully-connected sublayer. This enables easy parallelization of neural networks, which utilize the transformer architecture [21], leading to state-of-the-art results in machine translation.

3.2 Summarization Datasets

Data for training and evaluating supervised summarization models is, at least in the context of a supervised environment, rare. The reason for this limited availability is that for each input-document in a given training set a corresponding *gold-standard* summary is required. These gold-standard summaries are usually handwritten by human experts and therefore expensive to obtain [6] .
As described in section 2.1, we have the corpus of parliamentary textual data as input-files for our summarization models at hand, but, at least at the beginning of our project, do not have corresponding summaries at our disposal. A quite challenging situation, further complicated by the fact that the data from the Bundestag is in German. Nevertheless, over the course of our project, we identify suitable datasets to train and evaluate our models on the one hand and find solutions to create them ourselves on the other hand.

- **SwissText:** For training purposes we use the dataset released for the SwissText2019 Summarization Challenge[6]. In order to create this dataset, researches of the Zurich University of Applied Sciences have taken the body of 100 000 german Wikipedia articles as source text and its respective lead section as summary. Thereby, they created the first German summarization dataset with significant size.

- **Heute im Bundestag:** To evaluate the performance of our models on actual data from the Bundestag, we have the need for labeling the given text with gold-standard summaries. Fortunately, the Bundestag provides a service called *Heute im Bundestag* [7] which publishes short press-releases of discussed Drucksachen. There is a total of 874 of these press-releases available on the website. As the nature and importance of different types of Drucksachen varies, however, we decided to focus on Drucksachen of type *Antrag* since they are often summarized by the service and particularly important for the work of the Bundestag. Therefore, we further reduce our dataset to 758 releases explicitly referencing Drucksachen of this type. However, out of these samples, a considerable amount mentions several Drucksachen, which makes them hard to use as summaries for specific texts. Therefore we restrict the considered labels to the ones mentioning exactly one Drucksache, that results in a further reduction to 91 datapoints. Cutting out the Drucksachen with a character-length not reasonably processable by our models, we eventually obtain a dataset consisting out of 50 samples with lengths between 2 810 and 12 665 characters.

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

[7] `https://www.bundestag.de/hib`

- **Single parliamentary speeches:** As the primary target for our summarization task is parliamentary speeches, we include a subset of them in a summarization-dataset for the evaluation of our models. Using the downloader presented in section 2.1, we collect speeches from the most recent parliamentary sessions (as of 13 July 2020) with lengths ranging between 4 000 and 6 000 characters. To enhance the diversity of the dataset, we only extract one speech per session agenda point, as usually all of the speeches in one session agenda point handle the same topic. The speech with the amount of characters closest to 5 000 is picked. As this dataset is used for inference, it is limited to 500 samples but can be easily extended if needed.
- **Parliamentary Speeches aggregated by session agenda points:** In order to be able to train multi-document summarization models as well, we need a dataset that features multiple input-files per datapoint. To create such a dataset, we take advantage of the fact that parliamentary speeches can be clustered by the agenda-points they cover. By applying this rationale for the legislative period 19, we are able to generate a dataset containing a total of 1 298 agenda points. These agenda points are, on average, covered by 22 parliamentary speeches, each having an average length of 3 653 characters.

3.3 Translation Services

Especially in the field of English text summarization, several achievements have been made in recent years. Prophetnet [25] from Microsoft Research, for example, improves results by using n-gram prediction instead of predicting only one word. Pegasus [27] from Google Research performs very well with only small amounts of training-data at its disposal by masking whole sentences instead of single words.
In order to profit from these developments, we established a procedure to use promising English summarization models with our German text-files. To be more concrete, we translate our German test data to English to then summarize the text in English. Once processing is finished, we translate the retrieved summarization back to German to obtain the final result. In this fashion, any English summarization model can also be used with German data. However, there are drawbacks which we need to consider as well: First, information gets lost during translations. Therefore, for German input-files, an English model used with our translation-procedure does not perform as good as a comparable model created specifically for German. Second, translation-providers usually limit their services. Therefore, we have to use different translation services to keep the costs of the project on a reasonable level. We use *GoogleTranslate*, *MyMemory* and *AWS-Translate*. This cost-issue also prevents us from fine-tuning English models since a tremendous amount of translated data would be needed. Thus, we have to use English models as they are.

3.4 Selected Models

3.4.1 Model from Scratch

As the first step in our project, we build a summarization model with an LSTM encoder-decoder attention-based architecture from scratch. By using the `tensorflow-` and `keras-`

packages, we are able to train our encoder- and decoder-structures with an additional attention-layer on top. The architecture used is based on an entry in an online-blog [14] and was originally designed for summarizing short, English text-data like Amazon reviews. Our goal in building a model from scratch is to get a thorough understanding of every single component that constitutes a summarization model as well as to establish a baseline to compare the more sophisticated supervised and unsupervised approaches presented in the following sections with. We build the encoder and decoder components using stacked LSTM layers and use the attention-layer implemented in the `attention_keras`[8] package. The architectures of the encoder and the decoder are depicted in figure 7. We then train this model to predict the target sequence offset by one-time step: given an input word from a sequence, the network should predict the next word.
Since no target sequence is available, the inference phase generates the text word-by-word, as pictured in figure 7. We first encode the input and pass the final encoder hidden state to the decoder, thereby generating the first word. For each decoder time step, we compute the probability distribution conditioned on the previous word in the output sequence. We always choose the highest scoring word, though the model is extensible to more advanced inference strategies such as beam search. The full model architecture can be found in the Appendix B in Figure 16. The model consists of 380 020 868 trainable parameters.

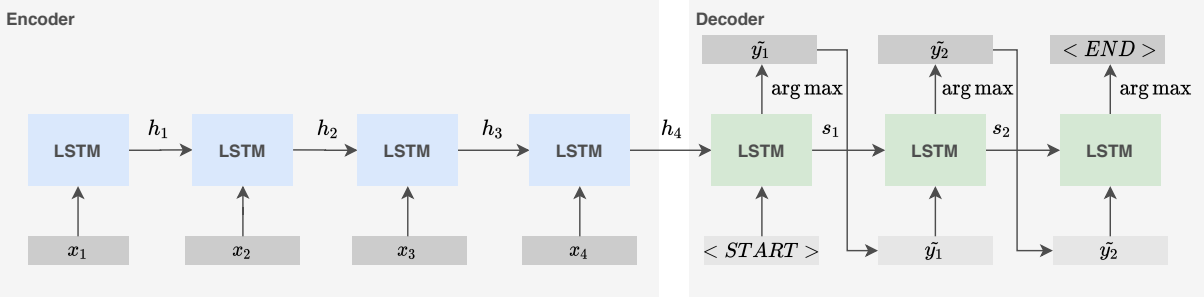

Figure 7: Neural architecture of the model built from scratch.

3.4.2 German BERT and BERTSumAbs

The idea behind the transformer architecture has been briefly introduced in section 3.1.2. One of the key features of the neural networks, which utilize the transformer architecture is that they are easily parallelizable [21]. This capability, among other factors, led to the emergence of numerous language models, which could be pretrained on huge amounts of text in a self-supervised fashion. One of the most successful pretrained models of this kind is *Bidirectional Encoder Representations from Transformers (BERT)* [7]. BERT's model architecture is a multi-layer bidirectional Transformer encoder based on the original implementation described in Vaswani et al. [21]. In contrast to previous efforts that look at a text sequence either from left to right or in a combined left-to-right-, right-to-left-manner, BERT's key technical innovation lies in applying bidirectional training to a Transformer model. The results from [7] as well as successful applications of the model on downstream tasks show that taking into account both the context after and before words allows BERT to have a deeper understanding of language than single direction language

[8] `https://github.com/thushv89/attention_keras/`

models. The architecture of the original BERT model is visualized in on the left in Figure 8.

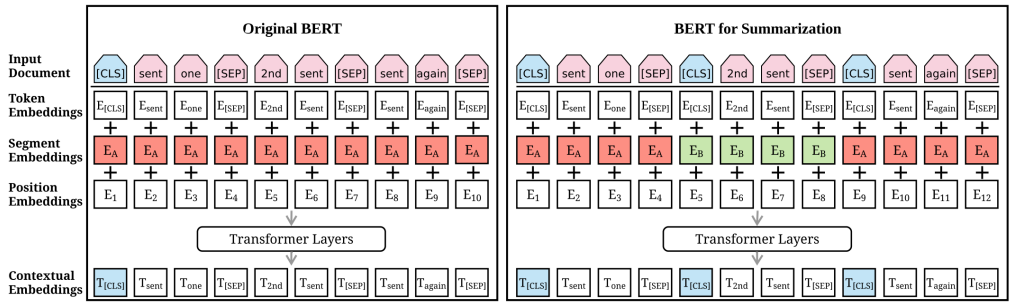

Figure 8: Overview of BERT architectures: The original BERT is displayed on the left, the for summarization adapted BERT on the right.

The sequence on the top constitutes the input document, which is extended with the special token *[CLS]* in the beginning and with the special tokens *[SEP]* to mark the borders of the sentences. Each of these tokens is equipped with three different embeddings: token embeddings, segment embeddings, and positional embeddings. In the end, these embeddings are added. These summed vectors subsequently used as inputs to several bidirectional transformer layers, which generate contextual vectors for each token. These contextual vector corresponds to the *[CLS]*-tokens from the beginning and possesses the information about the entire document. Thus, it can be used as a decoder input for downstream-tasks.
The reason why we elaborate on the BERT language-model in such a detailed manner is that one of the state-of-the-art summarization model at the time of writing is based on it. Yang Liu and Mirella Lapata (2019) [11] introduce in their paper adjustments to the encoder-structure of BERT in order to transform it into a summarization model which they call *BERTSumAbs*. The concrete adaptions depicted on the right in Figure 8, are the following: First, in order to be able to summarize more than two sentences, the authors insert multiple *[CLS]*-tokens before each sentence. For the same reason, the authors replace the segment-embeddings with interval-embeddings, which mark each even sentence with an array of zeros and each odd sentence with an array of ones. Moreover, the decoder is a 6-layered Transformer which is initialized randomly. This can make the fine-tuning unstable (encoder overfits and decoder underfits or vice versa). That is why Liu and Lapata also suggest a special fine-tuning scheduler, which separates the optimizers of the encoder from the ones of the decoder by setting different time-dependent learning rates for each of them.
As a basis for our BERTSumAbs-implementation the work done by Microsoft [9] has been taken. Due to the modularity of the `Huggingface transformers` library [24], it was quite easy to adapt this implementation to use the German BERT pretrained model, kindly provided by deepset. In the remainder of the document, we refer to this model as dist-bertabs-ger.

[9] `https://github.com/microsoft/nlp-recipes`

3.4.3 Pretrained English Model BART

In 2019, Facebook AI released an encoder that can be seen as a generalization of *BERT* [7] and *GPT* [16]. As mentioned in Section 3.4.2, BERT uses a bidirectional encoder, which means it takes the right- and left-sided context into consideration to then predict the masked tokens from the input. The drawback of this approach is that tokens are predicted independently from each other, meaning that they are not generated in a coherent manner, which becomes a problem when BERT is employed for text generation tasks. This issue is solved by GPT, however, due to the autoregressive decoder, it employs. This autoregressive decoder takes its outputs as input for the next iteration to be able to predict the next token coherently. However, GPT therefore, only takes the left context from the predicted token into consideration, which is solved by the Bidirectional and Auto-Regressive Transformers (BART) [9] depicted in Figure 9. This bidirectional but still auto-regressive approach enables BART to be pretrained on complex noise transformations that replace whole text-spans of arbitrary length[10] in the input document to predict how many and which words have been masked.

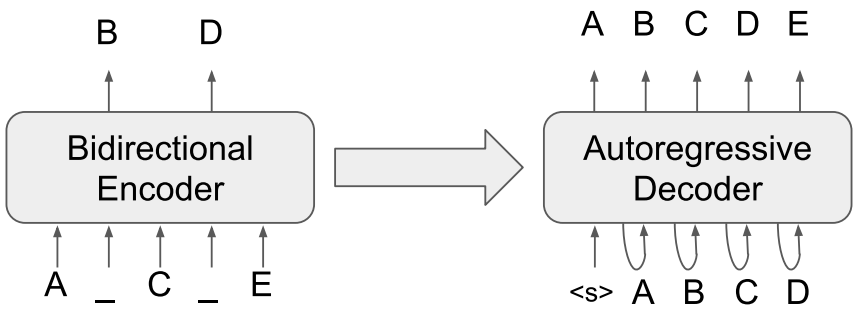

Figure 9: Scematic architecture of the Bidirectional and Auto-Regressive Transformer.

Facebook pretrained BART on 160 GB of English news, books, stories, and web text and then finetuned it on the *cnn-dailymail*-dataset which contains 1,27 GB of news articles and corresponding summaries. This pretrained BART is accessible through the `HuggingFace transformers` library [24]. We use a smaller version of the full BART created by Sam Shleifer [11] as this version is faster. It is called *distil-bart-eng* in the remaining parts of the document.

3.4.4 Unsupervised Model

As noted in chapter 3.2, the previously presented supervised models require summarization-datasets which not only provide input-documents but also corresponding gold-standard summaries to be trained. However, these kinds of datasets are very limited in their availability. Furthermore, adapting summarization-models trained on one specific type of document, e.g., news articles, to documents from another domain is not straightforward [6], and it is often not feasible in a real-world setting to justify the high cost of obtaining

[10] Also including text-spans of length zero.

[11] `https://huggingface.co/sshleifer/distilbart-cnn-12-6`

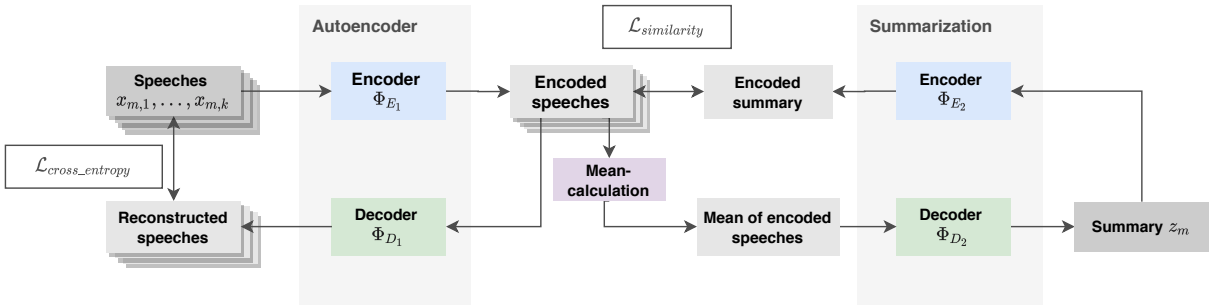

Figure 10: Neural architecture of the unsupervised summarization model [6]. Note that the weights of the two encoders (indicated in green), as well as those for the two decoders (indicated in blue), are tied.

summarization-datasets tailored for a specific use case[12].
In our project, the drawbacks described above are especially prevalent for parliamentary speeches. While Drucksachen is not 100% congruent with the Wikipedia-snippets of the SwissText-dataset described in section 3.2, it can still be argued that there is sufficient similarity in sentence structure and wording. The parliamentary speeches, however, differ significantly in style. We therefore also explore unsupervised model architectures for abstractive summarization.
Unsupervised abstractive summarization is still a very young scientific field and the available literature is very limited. Our review of this literature establishes two main directions of research. One of them is lead by Dohare et al. (2008) [8]. They use so-called Abstract Meaning Representation(AMR)-graphs to extract the key elements of a given input document in a first and to abstractively reproduce these main points in a second step. However, the AMR-annotations necessary for training these kinds of models are as sparsely available as summarization-datasets [6]. Thus, AMR-related models do not offer any considerable advantage over supervised models regarding real-world applicability.
The other main direction of research in unsupervised abstractive summarization concerns multi-document summarization. A recent promising approach in this field is presented by Chu and Liu (2019) [6]. Their model *MeanSum* was originally designed to summarize multiple Yelp-reviews of a given business. For our project, we adapt this model to summarize different parliamentary speeches grouped into the same agenda point[13]
MeanSum is based on a pretrained English language model[14] and tailored for English inputs. Thus, we train and evaluate our model-adaption with translated versions of the parliamentary speeches. For translation we use the services described in chapter 3.3.
The network contains two main, jointly trained components: an autoencoder- and a summarization-submodule, both of which contain an LSTM-encoder $\Phi_{E_i, i\in\{1,2\}}$ and -decoder $\Phi_{D_i, i\in\{1,2\}}$ and are initialized with the mentioned pretrained language model. During training, the autoencoder learns vector-representations for each speech given as

[12]In most cases the only viable option is to use crowd-workers. Taking into account the high amount of cognitive work required to summarize large documents, costs to obtain summarization-training-data at a large scale are exorbitant.

[13]We are not processing Drucksachen with the MeanSum-model since different Drucksachen are usually not related to each other.

[14]Unfortunately Chu and Liu (2019) do not specify which language model exactly.

an input to the model. It uses a reconstruction-loss $\mathcal{L}_{reconstruction}$ by computing the cross-entropy [4] between the original speeches $\{x_1, \ldots, x_K\}$ and the reconstructed speeches to ensure that the information-loss during the encoding-process of the speeches is kept at the smallest possible level:

$$\mathcal{L}_{reconstruction} = \sum_{k=1}^{K} \mathcal{L}_{cross\_entropy}(x_k, \Phi_{D_1}(\Phi_{E_1}(x_k))) \ . \tag{1}$$

In a subsequent step, the mean of the speeches' vector-representations is calculated. The summarization-module decodes this mean to generate the desired summary. This module is trained with a loss function based on the similarity between input-speeches and generated summaries $\{z_1, \ldots, z_N\}$. Specifically, the cosine-distance d_{cos} [4] between each encoded speech $x_{m,k}$ and the corresponding encoded summary z_m of a given agenda point $a_m, m \in \{1, \ldots, M\}$ is calculated to guarantee that the machine-generated summary are related to the input-speeches to a sufficient degree:

$$\mathcal{L}_{similarity} = \sum_{m=1}^{M} \sum_{k=1}^{K} d_{cos}(\Phi_{E_2}(x_{m,k}), \Phi_{E_2}(z_m)) \ . \tag{2}$$

The equally weighted sum of the results of equation 1 and 2 eventually constitutes the overall loss-function of the model:

$$\mathcal{L}_{model} = \mathcal{L}_{cross\_entropy} + \mathcal{L}_{similarity} \ . \tag{3}$$

The complete architecture of the MeanSum-model is presented in Figure 10.

3.5 Evaluation Metrics

3.5.1 Quantitative Evaluation

In the field of natural language processing, it is common practice to evaluate model performance automatically. Along with labeled datasets, we require a standardized way to measure the similarity between different text-files. The field of automatic summarization primarily uses two main concepts: ROUGE and BERT.
The ROUGE family was introduced in 2004 by Chin-Yew Lin [10] and is still widely used. Its underlying idea is to count the co-occurring tokens between a candidate and a reference summary. This score is then normalized by dividing by the total number of relevant tokens. The most critical metrics in the ROUGE-family are the following: ROUGE-1, which counts co-occurring words, ROUGE-2, which counts co-occurring pairs of words (also called *bigrams*), and ROUGE-L, which computes the longest overlapping sequence of words.
Another, more recent evaluation metric for summarization is represented by the *BERT-score*. The BERT-score relies on BERT-based contextual embeddings, which allow a more precise evaluation of semantic equivalence. The BERT-score is, therefore, more suitable for evaluating the quality of abstractive summaries. However, it is also more computationally intensive, as it needs to generate the contextual embeddings of the given input-documents and machine-generated summarizations at every step.

For the different metrics of the ROUGE-family as well as for the BERT-score, further subscores can be computed. The *recall* score measures how much of the information in a reference summary is featured in the candidate and is defined by the ratio:

$$\frac{\text{number of overlapping n-grams}}{\text{total number of n-grams in reference summary}} . \tag{4}$$

Precision measures how much of the information in the candidate summary is also present in the reference and is computed with the formula:

$$\frac{\text{number of overlapping n-grams}}{\text{total number of n-grams in generated summary}} . \tag{5}$$

Precision scores are a useful way to measure the extractive properties of a model at hand. In particular, the ROUGE-L precision is equal to 1 if the summary is a non-interrupted extract of the source.
In order to penalize large length differences between the candidate and the reference summary, we can use the *F1-score*, or the harmonic mean of precision and recall.

3.5.2 Qualitative Evaluation

To evaluate our models more comprehensively, we complement our quantitative analyses described in section 3.5.1 with a qualitative examination of our results. This qualitative evaluation is conducted in the form of a questionnaire that is composed of two main components: A language-related and a content-related part. The language related-part is further distinguished in two questions to assess the grammatical correctness on the one, and the fluency of a given summary on the other hand:

- On a scale from 0 *(miserable)* to 10 *(excellent)*, evaluate the grammatical correctness of the machine-generated summary.
- On a scale from 0 *(miserable)* to 10 *(excellent)*, evaluate the fluency of the machine-generated summary.

The content-related part is designed in accordance with the findings of Maynez et al. (2020) [13]. According to them, there are three important aspects which should be considered when evaluating the content of a summary: First, a high-quality summary should capture all key-points of a respective document. Second, summaries should not mix up the content of a given input. This *behavior* of summarization-models is labeled as "intrinsic hallucination" by Maynez et al. (2020) [13] and can lead to confusion of the reader at best and to point-blank factually incorrect summaries at worst. And third, a model should not exhibit "extrinsic hallucination" [13], therefore not augment its summaries with additional information that the source text does not contain.
Taking into account these findings of Maynez et al. (2020) [13], we designed the following three questions for our content-related part of the questionnaire:

- On a scale from 0 *(no key points captured)* to 10 (all key points captured), evaluate if the machine-generated summary has captured all important key-points of the input-text.

Model	r1-score	r2-score	rl-score	BERT-score	precision
distil-bart-eng	0.2024 (0.07)	0.0456 (0.04)	0.1955 (0.07)	**0.6251 (0.05)**	0.7274
distil-bertsumabs-ger	**0.2222 (0.12)**	**0.0801 (0.10)**	**0.2479 (0.13)**	0.6224 (0.08)	0.4454
bertsumabs-ger	0.1866 (0.08)	0.0470 (0.05)	0.1997 (0.08)	0.5813 (0.05)	0.3447
lead-sum	0.1882 (0.08)	0.0412 (0.05)	0.1777 (0.07)	0.6112 (0.05)	**1.0**
english-lead-sum	0.1782 (0.07)	0.0341 (0.04)	0.1682 (0.06)	0.6049 (0.04)	0.7317

Table 4: Model performances on SwissText dataset. The best scoring model for each metric is indicated in bold.

- On a scale from 0 *(content is so mixed-up, that meaning is changed fundamentally)* to 10 *(content is not mixed up at all)*, evaluate if the machine-generated summary has mixed up the content of the input-text.
- On a scale from 0 *(summary covers a completely different subject)* to 10 *(summary does not add any misleading information)*, evaluate if the machine-generated summary has added misleading information compared with the input-text.

3.6 Results

3.6.1 Model from Scratch

We train our baseline-model on 20 000 samples of the SwissText dataset introduced in 3.2. Since one epoch takes approximately 9-10 hours to complete on a single Nvidia P100 GPU, our training amounts to six epochs with a batch size of 150 and a final validation loss of 1.31.
Therefore, the resulting summaries are rather unsatisfactory, as the model-output just tends to repeat the same words multiple times in a row. A possible reason for this is that the model does not scale well for larger amounts of text, as it was originally intended for much shorter Amazon reviews. Another explanation is that we did not have enough training samples, meaning that the network was not able to build a sufficiently large language model to accurately predict the next word in a sequence. This further supports our main approach of using pretrained models, as these have a better syntactic and semantic understanding of the German language.

3.6.2 Supervised Approaches

Employing the three supervised models *distil-bart-eng*, *distil-bertsumabs-ger* and *distil-bertsumabs-ger* on the SwissText dataset and evaluating the results with respect to the scores introduced in Sec 3.5 one obtains the results depicted in Table 5.
For the sake of better judgment, we introduce three baseline models:

- **lead-sum** This benchmark is quite common in the field. It consists of taking the first three sentences of a text as its summary. If a summarization model does not beat this threshold, its practical usefulness can be considered very low.
- **english-lead-sum** This was introduced to evaluate how much information is actually lost in the process of translation. The model translates the first few sentences of a German text to English, takes the first three and translates them back to German.

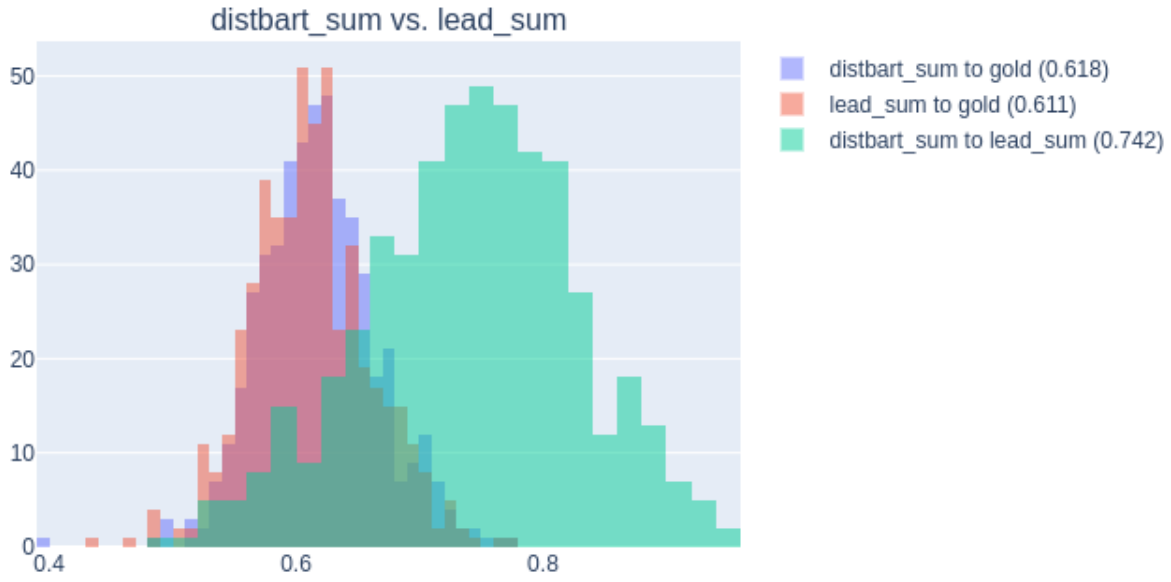

Figure 11: Distribution of BERT-score between results of *distil-bart-ger* and *lead-sum*.

- **rand-sum** (see Table 6) *rand-sum* ignores the source text and returns a random gold summary as candidate summary. This mimics a model over-fitting on the training data and not considering the input enough. Also, it gives us a way to evaluate how similar the different gold summaries are one to another.

Regarding the *english-lead-sum*, which relies on translation, we observe a consistent decline in all scores of about 0.01 which we can expect to be present in all models relying on translation to English.
Futher, *distil-bertsumabs-ger* performs better than *distil-bertsumabs-ger* in all metrics, which is the reason why we will only consider the distilled version as it also has a shorter runtime.
The model relying on BART has almost the same precision as the *english-lead-sum*. This can be explained by investigating further into what BART actually does when processing text. Qualitative analysis of the produced results shows that *distil-bart-ger* strongly relies on involving the first few sentences of a text into the produced summary. This shows that *dist-bart-ger* is actually more similar to *lead-sum* than to the gold summaries, which is confirmed by Fig 11. These results support the hypothesis of Fangfang Zhang, Jin-Ge Jao and Rui Yan that current state-of-the-art methods for abstractive summarization have highly extractive behavior. [26]
When looking at the results of *distil-bertsumabs-ger*, which was finetuned on the SwissText dataset, we see that it has a strong variation in its scores. This manifests, for example, in an almost doubled standard error in BERT-score compared to the other models. In Fig 12the concrete distribution is depicted.
The model does not seem to produce a normal distribution as there is a second peak at 0.7. By looking at the three top-scoring summaries in Table 5, one understands how this comes. The reason the finetuned model performs well on singular data points lies in the nature of the dataset. Wikipedia articles about similar topics are written in a similar

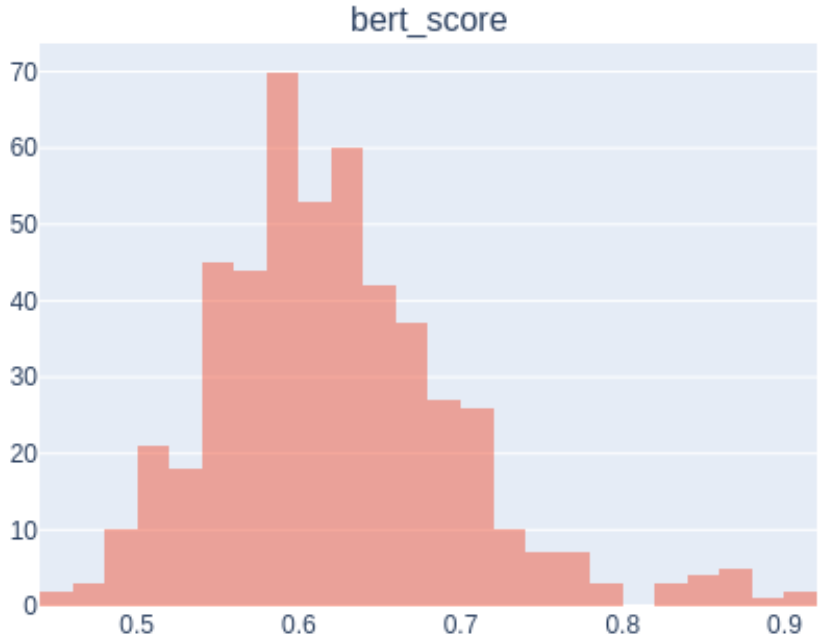

Figure 12: Distribution of BERT-score using *distil-bertsumabs-ger* on the SwissText dataset.

John Stewart war ein US - amerikanischer Politiker . Zwischen 1823 und 1823 vertrat er den Bundesstaat Pennsylvania im US - Repraesentantenhaus.
William Brown war ein US - amerikanischer Politiker . Zwischen 1847 und 1823 vertrat er den Bundesstaat Virginia im US - Repraesentantenhau .
Samuel Flander war ein US - amerikanischer Politiker . Zwischen 1847 und 1823 vertrat er den Bundesstaat New York im US - Repraesentantenhaus .

Table 5: Best performing summaries created with *distil-bertsumabs-ger* on SwissText.

way, and therefore, the model can easily learn the structure of the sentences. As we see in the summaries, the dates are not in the right order, and by looking at the non-depicted source texts, one sees that also the name of the state and even the name of the respective politician are often wrong. This shows that BERT-score, while better than ROUGE, is still not a good metric when it comes to separating sentences with similar structure but crucially different content from one another. We can expect *distil-bertsumabs-ger* to perform significantly worse on the *Heute im Bundestag* dataset as it has not seen the data before and can therefore not exploit similar sentence structures.
The performance of *lead-sum* (see Table 6) on the *Heute im Bundestag* dataset is significantly worse than on SwissText because the Drucksachen commence with organizational sentences having nothing to do with the content. Although we observe that distil-bart behaves similarly to *lead-sum*, it still captures where the interesting content actually lies and therefore performs better than *lead-sum*.
As expected, *distil-bertsumabs-ger* does not perform as well on this data as it has on SwissText. The high ROUGE scores of *rand-sum* indicate that the different gold summaries are strongly resembling each other. This gives hope that fine-tuning the model also on this

Model	r1-score	r2-score	rl-score	BERT-score
distil-bart-english	0.1275	0.0288	0.1383	**0.5994**
distil-bertsumabs-ger	0.0983	0.0053	0.0784	0.4867
rand-sum	**0.2300**	**0.0338**	**0.1833**	0.5863
lead-sum	0.0123	0	0.174	0.4713

Table 6: Model performances on the *Heute im Bundestag* dataset. The best scoring model for each metric is indicated in bold.

Model	r1-score	r2-score	rl-score	BERT-score	precision
distil-bart-ger	0.1889 (0.08)	0.0460 (0.05)	0.1824 (0.08)	0.6184 (0.05)	**0.8662**
distil-bart-eng	0.2024 (0.07)	0.0456 (0.04)	0.1955 (0.07)	**0.6251 (0.05)**	0.7274
distil-bertsumabs-ger	**0.2222 (0.12)**	**0.0801 (0.10)**	**0.2479 (0.13)**	0.6224 (0.08)	0.4454

Table 7: Performance of distil-bart-ger on SwissText dataset. The best scoring model for each metric is indicated in bold.

dataset might increase the score, but as we only have 50 samples, the effect will probably be small. Still, a model that needs fewer samples for fine-tuning, like pegasus [27], might achieve a further increase in performance. The fact that *distil-bart-english* outperformed *rand-sum* in BERT-score reflects the fact that BERT-score is better in capturing semantic resembling than ROUGE.

While evaluating *distil-bart-eng*, we also tried evaluating the distilled BART directly on german text (further called *distil-bart-ger*) and obtained decent results on the SwissText dataset as we can see in Table 7. This economizes the translation overhead with the drawback that performs a little worse than *distil-bart-eng* in all metrics. Still, this approach is more reliable when it comes to keeping the exact words. In particular, it also puts direct fine-tuning on German text into reach.
As the shortcomings of the automatic evaluation metrics became evident when evaluating *distil-bertsumabs-ger* on the SwissText dataset, we decided to further evaluate our best performing model *distil-bart-eng* on the *Heute im Bundestag* and *Parliamentary Speeches* datasets using the questionnaire introduced in the Section 3.5.2. The results of this evaluation are presented in Figure 13. On both datasets the model managed to be rather fluent and not add any information not mentioned in the source text for the majority of examples. The model has mediocre performance in terms of mixing up the content of the source on both datasets (avg. 6.66 on *Heute im Bundestag* and avg. 7.16 on *Parliamentary Speeches*). Moreover, the model performed quite poorly in terms of being able to capture important key-points. However, the performance on the *Heute im Bundestag* dataset is slightly better, which can be attributed to the more structured nature of the Drucksachen compared to the speeches. Lastly, the model produced mostly grammatically correct summaries. Slightly better performance on the *Parliamentary Speeches* dataset can be explained by the fact that the speeches are more similar to the data, *distil-bart-eng* was trained on, than the Drucksachen.

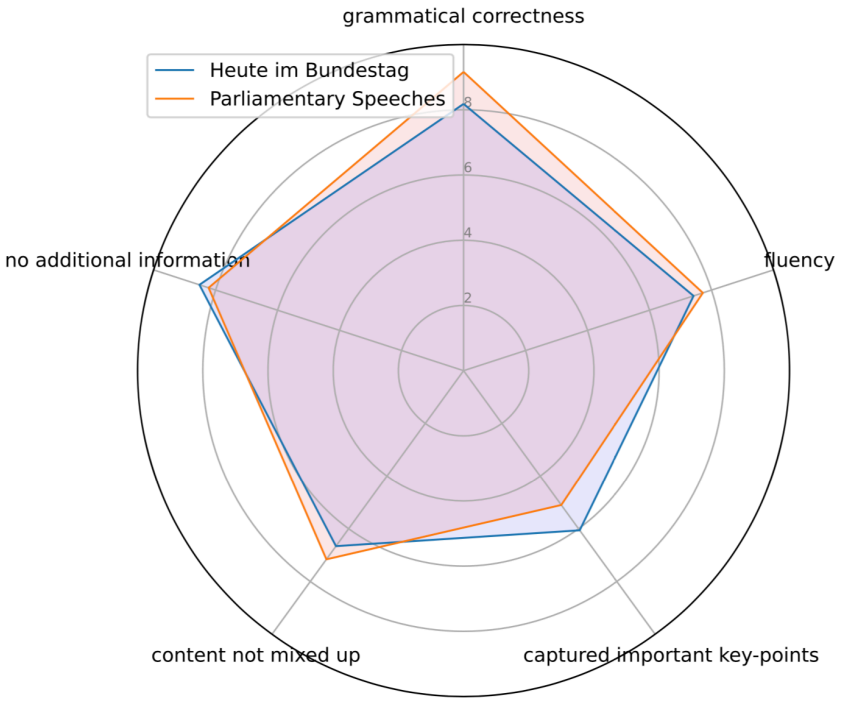

Figure 13: Qualitative evaluation of *distil-bart-eng*.

Model	r1-score	r2-score	rL-score
MeanSum	0.20149	0.04388	0.10882

Table 8: Average ROUGE-F1-scores between input-speeches and output-summaries for the MeanSum-model.

3.6.3 Unsupervised Approach

We train and test our adaption of the *MeanSum*-model with the translated version of the multi-document speech-dataset described in section 3.2. Examples of the summaries created by our model can be found in Appendix A.
Our evaluation is divided into two main components: quantitative and qualitative. As a starting point for the quantitative assessment, we consider the model's ROUGE-scores with respect to the speech inputs. To elaborate: since the whole idea of working in an unsupervised setting is to avoid the requirement of having gold-standard summaries at one's disposal, we can not compute ROUGE-scores between our machine-generated summaries and some kind of reference summaries. But what we can do, is calculating the ROUGE-score between our model output and the respective speeches which this output is intended to summarize. Average results are reported in Table 8.
In terms of ROUGE-L, our model reports quite low results. When interpreting these finding, a certain amount of caution is required however: When evaluating extractive summarization models, ROUGE-scores are - as elaborated in section 3.5.1 - a well-established and reliable measure for informativeness. As [13] argue though, in the setting of abstractive summarization the interpretation of ROUGE-scores is much more ambiguous. Since the ROUGE-metric basically measures n-gram-overlap, a low rL-score does not have to

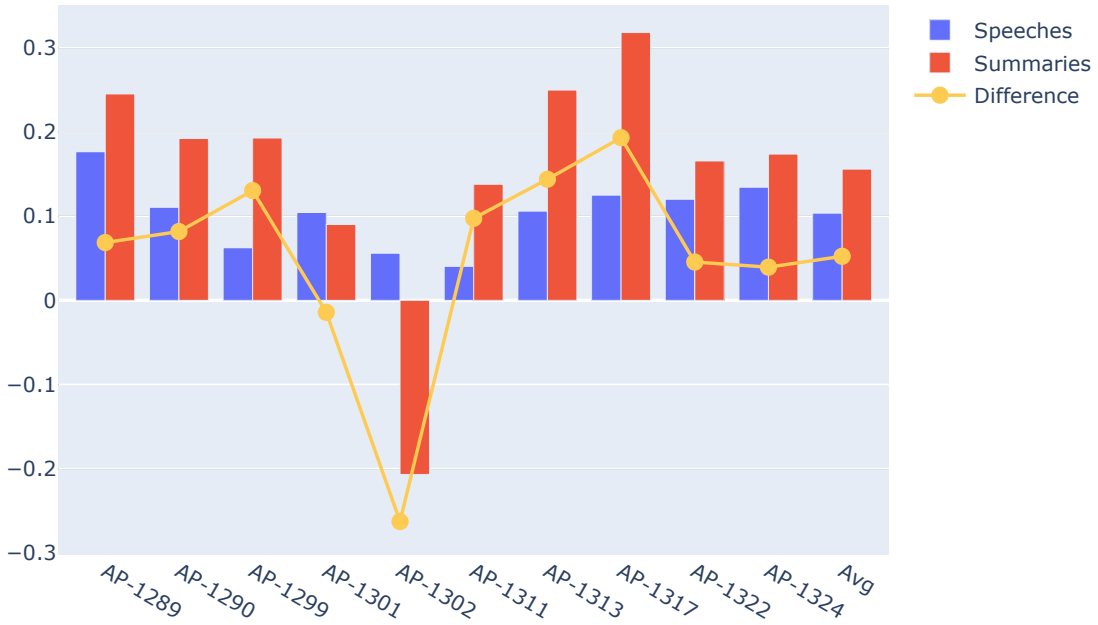

Figure 14: Polarity-comparison of parliamentary speeches and corresponding summaries.

be considered as disappointing per se. After all, an abstractive summarization model is supposed to rephrase when reflecting the key points of a given input document. Thus, we argue that the low ROUGE-L-score of our adaption of the MeanSum-model is actually positive since it can be interpreted as an indicator for high abstraction. That informativeness is not impeded considerably as a result of this increased abstraction is shown by the fact the ROUGE-1- and ROUGE-2-results are still comparably decent.
This notion is further supported when the sentiment of input-speeches and of their corresponding summary is compared. According to Chu and Liu (2019) [6], "a useful summary should reflect and be consistent with the overall sentiment of the [input]". Figure 14 displays the polarity-results for ten agenda points from the test-dataset. As can be seen, in nine out of the ten cases the machine-generated summary is categorized as having the same sentiment (either positive or negative) as the corresponding speeches. Although the machine-generated summaries are biased towards being slightly more positive than their corresponding speeches, the mean polarity-difference between model-input and -output is only 0.0523 on a polarity-scale interval of [-1,1]. These results can therefore, be considered as a quality indicator for the MeanSum-model.
For the qualitative evaluation, we use the questionnaire introduced in section 3.5.2 on the machine-generated summaries. Figure 15 illustrates the average results.
MeanSum scores considerably lower than our best supervised model, the distilled English BART, especially in the categories *reading-fluency* and *mixed-up content*. Regarding the

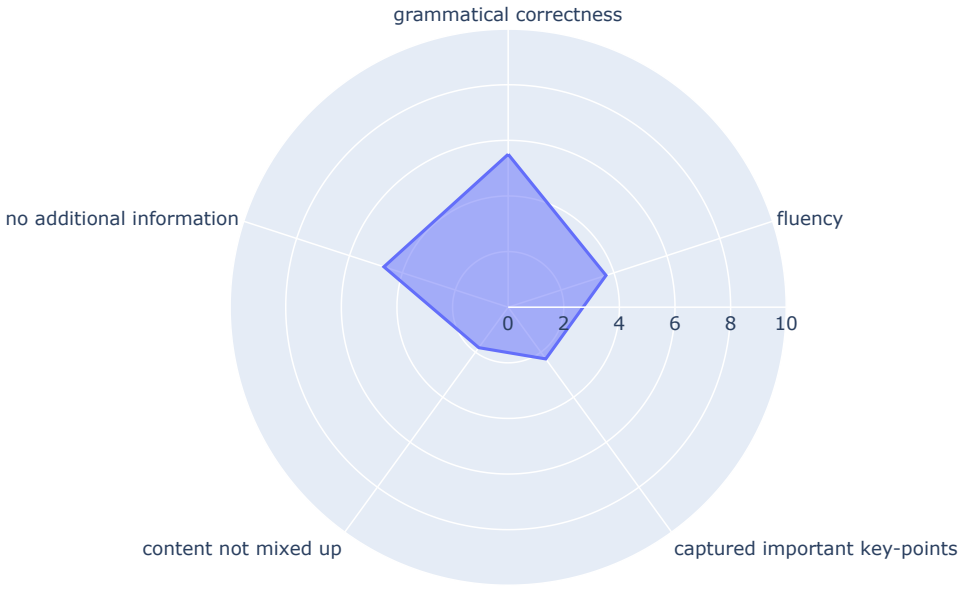

Figure 15: Average results of our questionnaire-evaluation of the MeanSum-model.

former, we suspect that the abstraction-level of the model is too high. A possible explanation for the mixed-up content is that the MeanSum-model was originally designed to handle short input-documents like Yelp or Amazon reviews. As a speech is, on average, several thousand characters in length, the encoder/decoder-architecture in most likely overtaxed in its current form. As a result, a considerable amount of information and structure is lost during the encoding of the speeches. This loss cannot be balanced out in the subsequent stages of the model. Therefore, the resulting summaries feature severely mixed-up content.
One approach to make the encoder/decoder-architecture of *MeanSum* more robust towards longer input-sequences, is to integrate attention- or pointer mechanisms. According to Chu and Liu (2019) [6] this can, on the one hand, reduce the abstraction-level of the model to a more sensible level and, on the other hand, enhance the model's ability to capture the critical points of the input-documents massively.
Taking into account the promising results of *MeanSum* in our sentiment-analysis, we reckon that enhancing the model-architecture with an attention-mechanism could provide one of the first competitively performing unsupervised summarization models.

4 Use Cases

Thanks to our collaboration with faktual, we have envisioned several ways to apply our data exploration results and summarization models in the industry. This is the basis for our first prototype.

4.1 Journalism

At the beginning of our project, the target group for our findings was journalists covering the events in the German parliament. We envisioned a tool that would reduce desk research by providing a summary for the speeches in each parliamentary session. Our first ideas for this tool are now included in a minimum viable product.
Throughout the project, we received extremely valuable feedback from Meinolf Ellers, the Chief Digital Officer of the Deutsche Presse-Agentur[15] and Roland Freund, the state office manager of the Deutsche Presse-Agentur in Bavaria. They emphasized to us that speed and accuracy are of utmost importance when covering the political landscape on a federal and a state level. The parliamentary protocols are usually released three to five days after the session, meaning that an automated solution would not be fast enough. We, therefore, consider journalists covering local parliaments and employees of compliance departments as potential users.

4.2 Compliance

Every week, 65 Drucksachen are proposed on average. Not all of them are extensively covered in press releases and many are discussed in a session months after their initial proposal. A promising target group for our product are compliance experts who must stay informed of the newest governmental discussions on specific topics. The techniques used in Milestone 1 and Milestone 2 can be repurposed into a knowledge base where users can track the development of law proposals from their begin to their execution. This knowledgebase would contain a historical overview of a variety of topics and the change in sentiment for each.

4.3 Minimal Viable Product

Using the ideas from the previous two subsections, we present a prototype by extending our Elasticsearch storage system with an interactive dashboard using Kibana, as depicted in Appendix B in Figure 17. Our tool currently contains an overview of the summaries from speeches of the 19th legislative period and is easily extendable for further text inputs. Additionally, we provide our sentiment analysis results and information about matching law proposals for each speech. Users can easily aggregate data and search for topics of interest using the Kibana interface.

[15] `https://www.dpa.com/en/`

5 Conclusion

In this Data Innovation Lab project, we use standard natural language processing techniques and novel deep learning architectures to process the parliamentary speeches in the German Bundestag. Although multiple models exist for English text, our main challenge is to adapt abstractive summarization for German data.
We attempt to train a network from scratch but have to realize that pretrained language models are necessary for the generation of fluent text. By finetuning German BERT, we obtain a model that performs well on the SwissText dataset but does not generalize well to documents from the German Bundestag. On the contrary, the language model *BART* is not finetuned on the datasets but still produces the best results in terms of ROUGE- and BERT-scores. Our self-designed evaluation questionnaire also supports this finding. Furthermore, we also explore an promising unsupervised approach for multi-document summarization.
Beside the trained models, we produce several other artifacts. We construct a parser for parliamentary speeches and Drucksachen, along with a translation interface for several providers. We use press releases published on the Bundestag's website to create a small dataset for quantitative and qualitative evaluation. Finally, a prototype of our summarization tool is available as an Elasticsearch instance.
Our cooperation with faktual involved talking to experienced people in the journalism field to evaluate the use cases of our product. As a result, we hav extended our targeted user base from journalists to corporate compliance departments. With these use cases in mind, we strongly believe our project will enable new exciting summarization approaches for German text.
In the future, our parsing tools can facilitate the creation of larger corpus of German parliamentary data. Our translation pipeline can also be used to create English summarization models in the future. Furthermore, our presented approaches can be extended for query-based or topic-based summarization and generation of summaries of user-defined length. Finally, as more sophisticated pretrained language models are released every year, we are excited to see how they will spearhead many new powerful abstractive summarization methods.

References

[1] Hunt Allcott, Matthew Gentzkow, and Chuan Yu. Trends in the diffusion of misinformation on social media. *Research & Politics*, 6(2):2053168019848554, 2019.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[3] David M Blei, Andrew Y Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.

[4] Malik Boudiaf, Jérôme Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. Metric learning: cross-entropy vs. pairwise losses, 2020.

[5] Olexandra Klymenko, Daniel Braun and Florian Matthes. In *Automatic Text Summarization: a State-of-the-art Review*, 2020.

[6] Eric Chu and Peter Liu. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International Cvonference on Machine Learning*, pages 1223–1232, 2019.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[8] Shibhansh Dohare, Vivek Gupta, and Harish Karnick. Unsupervised semantic abstractive summarization. In *Proceedings of ACL 2018, Student Research Workshop*, pages 74–83, Melbourne, Australia, July 2018. Association for Computational Linguistics.

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

[10] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[11] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders, 2019.

[12] Christopher Lucas, Richard A. Nielsen, Margaret E. Roberts, Brandon M. Stewart, Alex Storer, and Dustin Tingley. Text analysis for comparative politics, 2015.

[13] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919. Association for Computational Linguistics, 2020.

[14] Aravind Pai. Comprehensive guide to text summarization using deep learning in python. 2019.

[15] Mohamed Abd Elaziz Mohammed A. A. Al qaness, Ahmed A. Ewees, Abdelghani Dahou. In *Recent Advances in NLP: The Case of Arabic Language*, 2019.

[16] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[17] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[18] Nima Sanjabi. Abstractive text summarization with attention-based mechanism. Master's thesis, Universitat Politècnica de Catalunya, 2018.

[19] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[20] Leimin Tian, Catherine Lai, and Johanna Moore. Polarity and intensity: the two aspects of sentiment analysis. In *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*, pages 40–47, Melbourne, Australia, 2018. Association for Computational Linguistics.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[22] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[23] Sam Wineburg, Sarah McGrew, Joel Breakstone, and Teresa Ortega. The cornerstone of civic online reasoning. stanford digital repository. 2016.

[24] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[25] Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training, 2020.

[26] Fangfang Zhang, Jin-ge Yao, and Rui Yan. On the abstractiveness of neural document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 785–790, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[27] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019.

A Example summaries for all our models

In the following we present a speech of the CDU/CSU-deputy Herrmann Färber about the Drucksache 14 745 Beschlussempfehlung und Bericht des Ausschusses für Ernährung und Landwirtschaft (10. Ausschuss) and corresponding summaries created with all the models introduced in this report. For the sake of authenticity, the speech as well as the presented summaries are in German.

- **Source:** Sehr geehrter Herr Präsident! Meine Damen und Herren! Wir debattieren heute das Direktzahlungen-Durchführungsgesetz und damit die einmalige Erhöhung der Umschichtung um 1,5 Prozentpunkte – das entspricht einer Erhöhung um 4,50 Euro pro Hektar – aus der ersten in die zweite Säule. Damit schaffen wir die dringend benötigte Rechtssicherheit für Landwirte, die an den Agrarumweltprogrammen der zweiten Säule teilnehmen. Wir erwarten jedoch, dass die 1,5-Prozentpunkte-Umschichtung ausschließlich an die landwirtschaftliche Mittelvergabe gebunden bleibt. Und wir stimmen diesem Gesetzentwurf nur unter der Maßgabe zu, dass diese Umschichtung für ein Jahr gilt und eine einmalige Maßnahme bleibt. Es darf nicht zu einem schleichenden Ausstieg aus den Direktzahlungen kommen. Es wird auch nicht funktionieren, immer mehr praktische Leistungen für Umwelt und Naturschutz von den Bauern einzufordern und im Gegenzug die finanziellen Leistungen immer mehr zu kürzen.Ich möchte an dieser Stelle mein Wort an die Kollegen von der FDP richten: Bitte heucheln Sie heute nicht schon wieder! Im Bundesrat haben auch die Länder mit FDP-Beteiligung wie Rheinland-Pfalz oder Schleswig-Holstein ihre Zustimmung zu einem Antrag für eine Umschichtung von sogar 8,5 Prozent gegeben.Noch im Sommer haben Mitglieder Ihrer Fraktion die komplette Abschaffung der Direktzahlungen an Landwirte gefordert.Hier dann eine Umschichtung auf 6 Prozent abzulehnen, wie Sie es schon angekündigt haben, halten wir für geradezu unseriös.In den Anträgen der Fraktionen Die Linke und Bündnis 90/Die Grünen wird eine Weidetierprämie für Schafe und Ziegen in Form von gekoppelten Zahlungen aus den Direktzahlungen der ersten Säule gefordert. Wir lehnen aber gekoppelte Zahlungen grundsätzlich ab, weil dadurch falsche Anreize gesetzt werden.Den Bundesländern steht zur Förderung der Schaf- und Ziegenhalter bereits jetzt ein breites Maßnahmenspektrum zur Verfügung. Und gerade durch die heute zu beschließende Umschichtung erhalten die Bundesländer die Möglichkeit, diese Programme für Schaf- und Ziegenhalter finanziell aufzustocken und zusätzlich zu stärken.Die Fraktionen Die Linke und Bündnis 90/Die Grünen erkennen in ihrem Antrag völlig richtig, dass die Schafhalter mit ihrer Arbeit zum Natur-, Arten-, Hochwasser- und Klimaschutz beitragen und uns in der Bevölkerung gleichzeitig mit hochwertigen Produkten versorgen. Aber sie ziehen halt die falschen Schlüsse daraus. Die Schäferinnen und Schäfer dürfen nicht zu bloßen Almosenempfängern degradiert werden.Vielmehr müssen wir dafür sorgen, dass ihre Produkte wieder mehr Wertschätzung in der Gesellschaft erhalten und marktfähig werden.Wenn die Gesellschaft eine solche Art der Tierhaltung will, dann muss sie es auch mit Taten belegen. Die Wolle und das Fleisch aus heimischer Schafhaltung, aus extensiver Schafhaltung sind am Markt leider Gottes so gut wie wertlos. Es kann nicht sein, dass man die extensive Weidehaltung verlangt, aber dann intensiv

gemästetes Lammfleisch aus Neuseeland auf dem Teller liegen hat.An dieser Stelle sind wir alle verantwortlich; im Übrigen auch die Gastronomie des Parlaments.Wer Produkte aus Schafwolle verwendet, wer heimisches Lammfleisch oder ein Lammfell kauft, der ist nicht schuld am Tod eines Tieres, sondern der schafft überhaupt erst die Voraussetzungen dafür, dass Lämmer und Schafe hierzulande auf Wiesen, auf Weiden und auf Dämmen leben und die Schäferinnen und Schäfer von ihrer Arbeit leben können.Vielen Dank.Vielen Dank. – Zweiter Redner in der Debatte ist der Kollege Peter Felser für die Fraktion der AfD.

- **lead-sum:** Sehr geehrter Herr Präsident! Meine Damen und Herren! Wir debattieren heute das Direktzahlungen-Durchführungsgesetz und damit die einmalige Erhöhung der Umschichtung um 1,5 Prozentpunkte – das entspricht einer Erhöhung um 4,50 Euro pro Hektar – aus der ersten in die zweite Säule.

- **Model built from scratch:** vereins vereins vereins vereins verbandsfreie verbandsfreie verbandsfreie ratmeyer

- **MeanSum:** Hallo, liebe Kollegen! Liebe Gäste! Es geht darum, die Landwirtschaft zu fördern, auch wenn man sich das Ganze in einem Biergarten ansieht, der jetzt genutzt wird. Dies ist das Ergebnis der Bundesregierung und 2020. Meine Damen und Herren, meine Damen und Herren, lassen Sie uns auf die Erde gehen. Wir sind mitten im Budget. Dies ist ein großer Schritt nach vorne. Das kann man nicht einfach essen. Sie können auch darüber sprechen. Sie fordern das Steuerprivileg des EEG-Zuschlags, zum Beispiel für ein Video von 418 Millionen, das dann natürlich von der Regierung finanziert wird. Dies ist ein sehr wichtiger Schritt in Richtung Weltklima- und Umweltpolitik.

- **distil-bertsumabs-ger:** Die Simmulation ist ein Verfahren zurückragung, bei der für die für die für die führte. Das Verfahren ist in der Regel für das Verfahren zurückragung der Regel führt der Regel führte das Verfahren zurückragungen von der Regel führte - und das Verfahren zurührt - das Verfahren zur Verfügung der Regel führt der Regel fürt - Änderung der Regel führte das Verfahren der Regel führte.

- **distil-bart-ger:** Wir debattieren heute das Direktzahlungen-Durchführungsgesetz und damit die einmalige Erhöhung der Umschichtung um 1,5 Prozentpunkte . Wir erwarten jedoch, das entspricht Erhöhung um 4,50 Euro pro Hektar – ausschließlich an die landwirtschaftliche Mittelvergabe.

- **distil-bart-eng:** Die AfD debattiert über das Gesetz zur Umsetzung von Direktzahlungen und die einmalige Erhöhung der Umschichtung um 1,5 Prozentpunkte. Dies entspricht einer Steigerung von 4,50 EUR pro Hektar von der ersten zur zweiten Säule der Agrarumweltprogramme. Die Gruppen Die Linke und Bündnis 90 / Die Grünen erkennen in ihrer Bewerbung zu Recht, dass die Schafzüchter mit ihrer Arbeit uns qualitativ hochwertige Produkte in der Bevölkerung liefern.

B Complementing figures

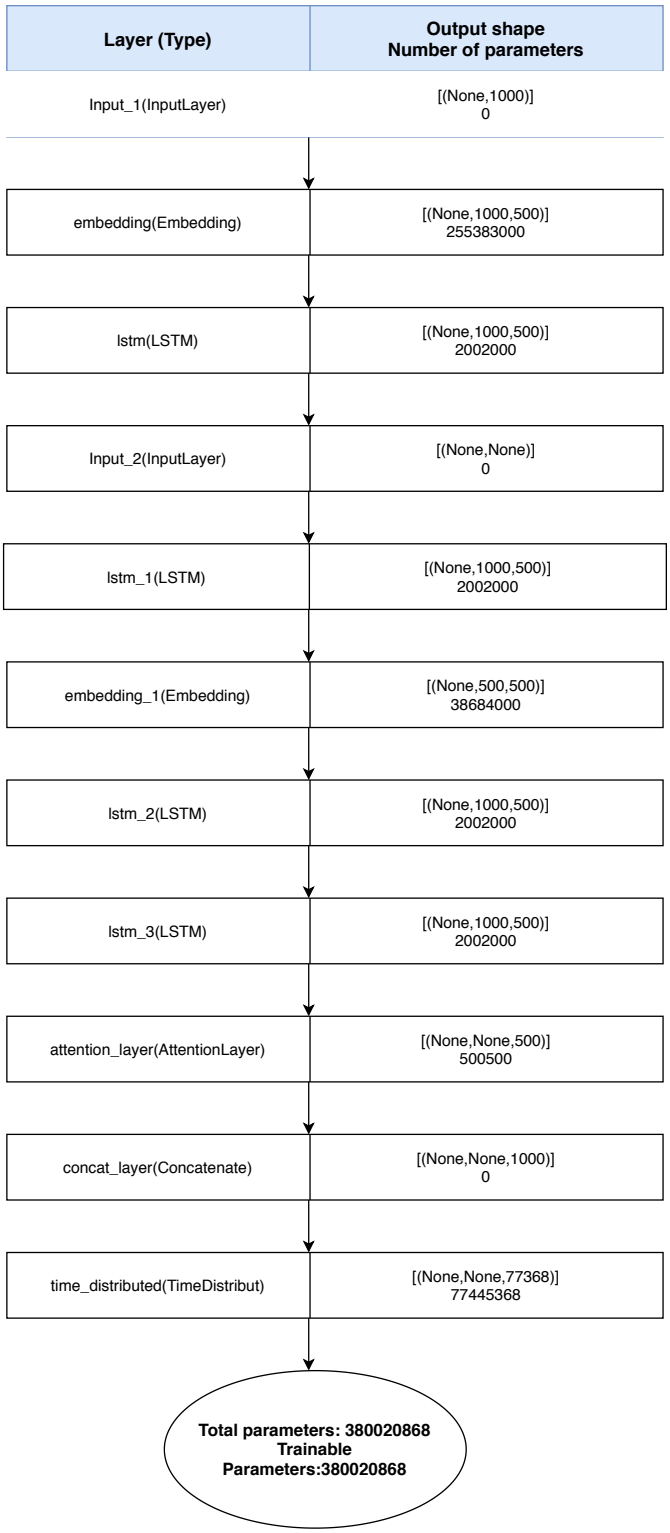

Figure 16: Neural architecture of the model built from scratch.

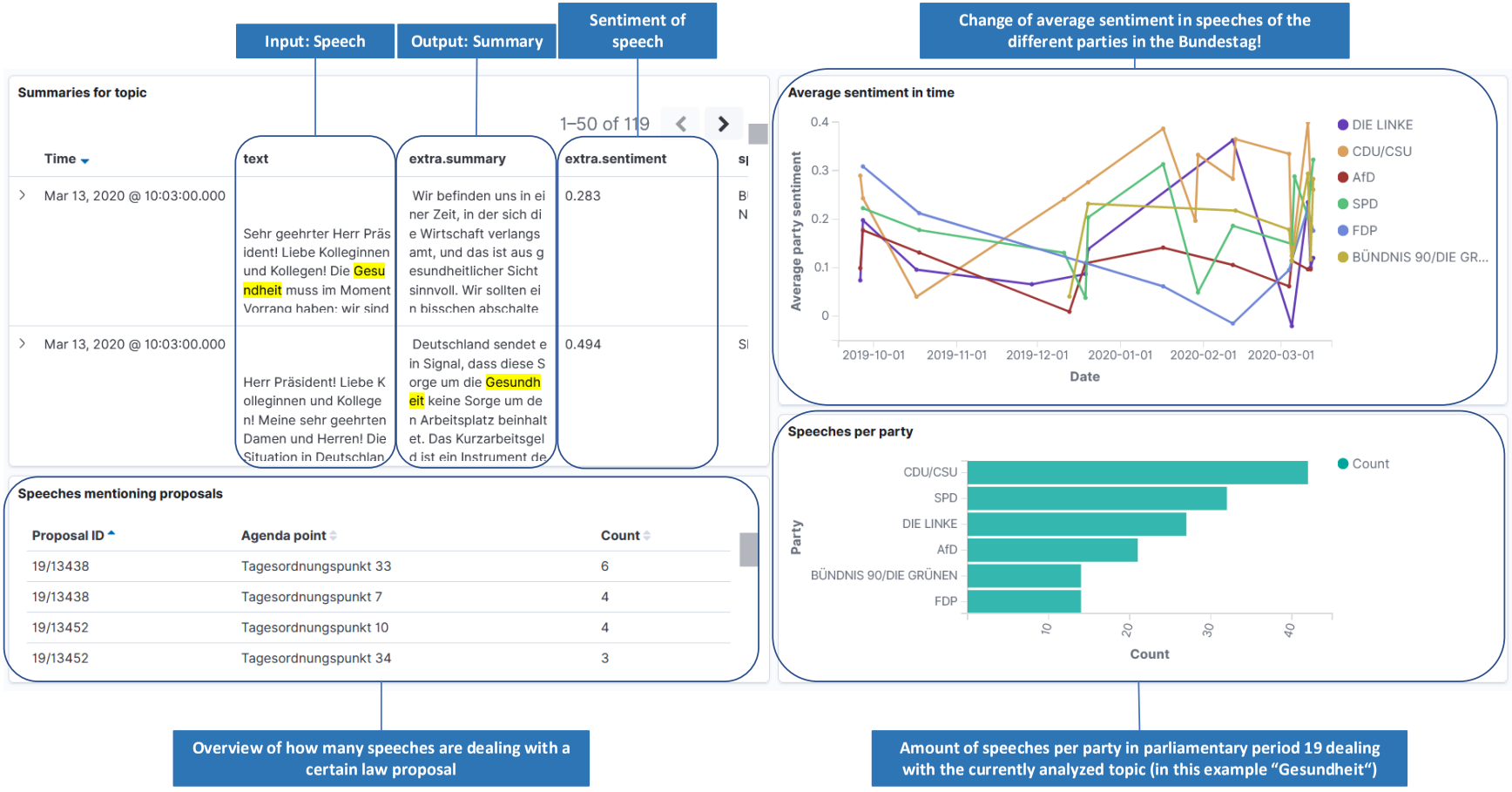

Figure 17: Screenshot of our prototype.