Chancen und Risiken von Text Summarization im deutschsprachigen Raum

Am Beispiel von Bundestagsreden

Kevin Bönisch

25. März 2023

Inhaltsverzeichnis

1. Einleitung

Mit dem stetigen Wachstum des Internets steigt auch die Masse an Informationen in Form von Dokumenten und Artikeln. Dies erzeugt ein Verlangen nach komprimierteren Darstellungen dieser Texte, ohne den relevanten Informationsgehalt zu verlieren. Automatic Text Summarization ist der Vorgang, eine flüssige und korrekte Zusammenfassung des Inputs zu generieren und dabei die originale Kernaussage abzubilden (vgl. Allahyari u. a. 2017, S. 1).

Forschungen in diesem Themenfeld begannen bereits in den 1950er Jahren, als H.P. Luhn eine Zusammenfassung auf Basis von Wort- und Satzfrequenzen vorschlug (vgl. Luhn 1958, S. 160). Seitdem wurden zahlreiche Methoden entwickelt, die sich in die zwei grundlegenden Ansätze Extractive Text Summarization und Abstractive Text Summarization einordnen lassen.

Der extrahierende Ansatz versucht dabei, dem Text die wichtigsten Sätze anhand eines Rankings zu entziehen und mit diesem dann die resultierende Zusammenfassung zu erstellen (vgl. Luhn 1958, S. 160). Die Zusammenfassung ist eine Satz-Untermenge des originalen Textes (vgl. Allahyari u. a. 2017, S. 2).

Der abstrakte Ansatz hingegen versucht, erst den Text zu verstehen, um daraus dann eine neue Zusammenfassung mit eigenen Worten zu generieren (vgl. Tahseen u. a. 2022, S. 103). Abstrakte Zusammenfassungen sind in der Regel wünschenswerter, da sie mehr dem menschlichen Zusammenfassen ähneln (Siehe Abschnitt 2.4).

Letzterer Ansatz hat 2017 durch die von Google eingeführte Transformer-Technologie (vgl. Vaswani u. a. 2017) einen Durchbruch erlangt. Resultate dieses Durchbruchs sind Sprach-Modelle wie T5, BART, PEGASUS oder ChatGPT.

Im Rahmen dieser wissenschaftlichen Ausarbeitung wurden mehrere Python-Skripte, unter anderem mithilfe von SpaCy [1], geschrieben, welche verschiedene Methoden der Text Summarization implementieren. Weiterhin wird ein Grundverständnis von Text Summarization vermittelt, um danach den Fokus auf die abstrakte state-of-the-art Transformer Technologie im Allgemeinen und in Verbindung mit PEGASUS und BART zu legen. Schlussendlich werden anhand eines Versuchs, basierend auf den Reden des Deutschen Bundestags, die Chancen und Risiken von Automatic Text Summarization erörtert. Der Code zum Versuch befindet sich auf GitHub[2].

[1] Open-Source NLP Python Library (vgl. Altinok 2021)

[2] https://github.com/TheItCrOw/text_summarization/tree/main/BundestagsSummarizer

2. Extractive Text Summarization

2.1. Ablauf

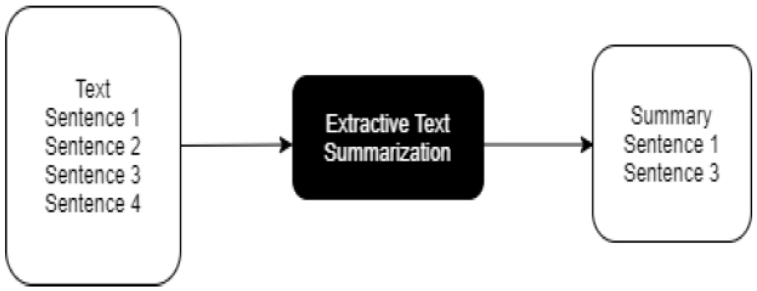


Abbildung 2.1.: Extractive Text Summarization (vgl. Tahseen u. a. 2022, S. 103)

Abbildung 2.2 zeigt den Ablauf einer Extractive Text Summarization und veranschaulicht die Aussage aus der Einleitung, dass die resultierende Zusammenfassung eine Satz-Untermenge des originalen Textes ist.

Welche Sätze in der finalen Zusammenfassung landen, entscheidet der jeweilige Algorithmus, der die extrahierende Zusammenfassung implementiert. Diese werden in Abschnitt 2.3 vorgestellt.

2.2. Architektur

Nach El-Kassas u. a. (vgl. 2021, S. 8) besteht die System-Architektur (siehe 2.2) für Extractive Text Summarization aus fünf Teilen:

1. **Pre-processing**: Beinhaltet unter anderem das Prüfen der Textlänge und möglicherweise Kürzen des Textes und das Unterteilen in einzelne Sätze.

2. **Creating a Representation of Text**: Abhängig vom Algorithmus wird der Text in eine geeignete numerische Darstellung konvertiert. Bei Graph-Algorithmen inkludiert dies zum Beispiel das Darstellen des Textes durch Kanten und Knoten (zitiert nach Joshi, Wang und McClean (2018)).

3. **Scoring of Sentences**: Anwenden des jeweiligen Algorithmus' und Sortieren der Sätze nach einem spezifischen Score (zitiert nach Nenkova und McKeown (2012)).

4. **Extraction of High-Scored Sentences**: Auswählen der Top N Sätze je nach Konfiguration und Verbinden dieser Sätze zur Zusammenfassung (zitiert nach Nenkova und McKeown (2012) und Junnan u. a. (2018)).

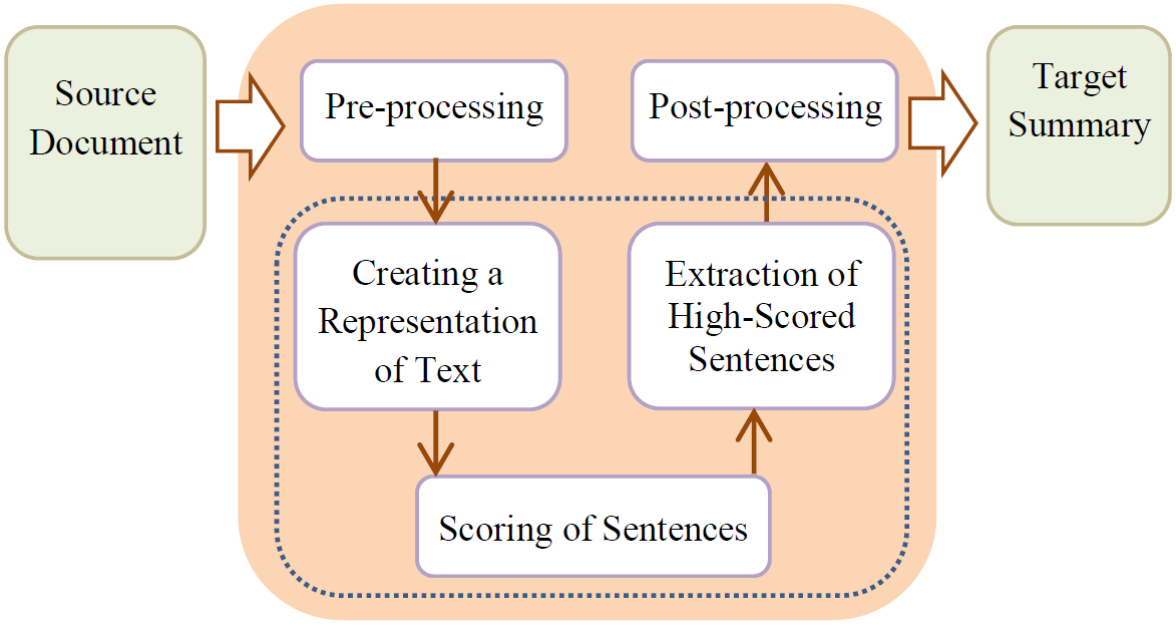


Abbildung 2.2.: Architektur einer Extractive Text Summarization (El-Kassas u. a. 2021, S. 8)

5. **Post-processing**: Beinhaltet unter anderem das richtige Anordnen der extrahierten Sätze und die Wiederherstellung des originalen Wortlauts dieser (zitiert nach Gupta und Lehal (2010a).

2.3. Methoden

Abschnitt 2.2 zeigt die grundlegende Architektur einer Extractive Text Summarization, die sich mithilfe von verschiedenen Algorithmen implementieren lässt. In Abbildung 2.3 ist eine Übersicht der vielzähligen Methoden-Kategorien zu sehen. Im Kontext dieser Ausarbeitung liegt der Fokus jedoch nur auf der Graph-basierten Methode TextRank, da diese im späteren Verlauf noch praktisch benutzt wird. Für alle anderen Methoden verweise ich auf El-Kassas u. a. (2021) oder Moratanch und Gopalan (2017).

2.3.1. TextRank

TextRank ist ein Graph-basierter Algorithmus zur Verarbeitung von Texten im Anwendungsbereich des Natural Language Processings, der vor allem seinen Nutzen im Extractive Text Summarization hat und auf dem bekannten PageRank Algorithmus von Google (Page u. a. 1999) aufbaut. Vorgestellt wurde er im Jahr 2004 von Rada Mihalcea und Paul Tarau.

Der folgende Abschnitt erläutert TextRank anhand des Original-Papers von Mihalcea und Tarau (2004). Es wurden seitdem Modifikationen und Adaptierungen des TextRanks vorgestellt, so zum Beispiel von Mallick u. a. (2019), die eine modifizierte Ähnlichkeitsfunktion

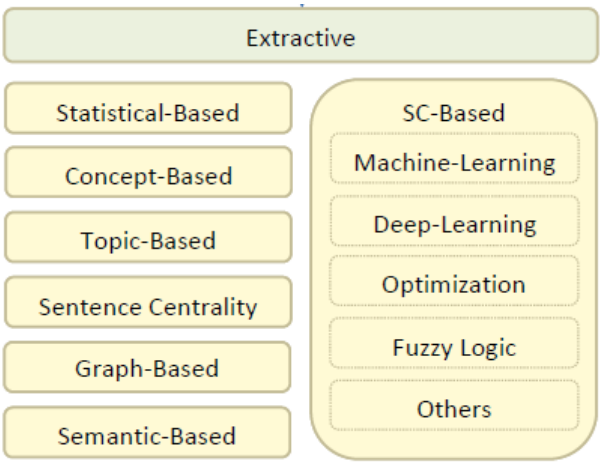


Abbildung 2.3.: Methoden von Extractive Text Summarization (ergänzt durch d. Verf., El-Kassas u. a. 2021, S. 8)

mithilfe von inverser Kosinus-Frequenz vorschlugen. In dieser Ausarbeitung wird jedoch das Original genutzt.

Idee

Bei TextRank wird jedem Knoten im Graphen eine Wichtigkeit zugeordnet, welche rekursiv vom restlichen Graphen abhängt. Wenn ein Knoten mit einem anderen Knoten verbunden ist, gibt dieser eine Stimme (von dem Englischen ”Vote”, also im Sinne von ”Wahl treffen”) an den anderen Knoten ab. Je mehr Stimmen ein Knoten erhält, desto relevanter ist dieser. Weiterhin wird das Gewicht dieser Stimme daran berechnet, wie viel der wählende Knoten selbst an Stimmen erhält. Somit entsteht eine rekursive Abhängigkeit, bei der das Gewicht eines Knotens vom Gewicht aller anderen wählenden Knoten abhängig ist.

Formell

Gegeben sei ein gerichteter Graph G mit $G = (V, E)$, wobei V die Menge der Knoten und E eine Kanten-Untermenge von $V \times V$ ist. Für ein Knoten V_i gilt:
$In(V_i)$ ist die Menge an Kanten, die zu V_i zeigen.
$Out(V_i)$ ist die Menge an Kanten, zu von V_i weg zeigen.
Der Wert eines Knotens V_i ist definiert als (Page u. a. 1999):

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

wobei d einen Dämpfungsfaktor darstellt, der einen Wert zwischen 0 und 1 annehmen kann.

Ablauf

Gegeben sei ein Text T. Für jeden Satz in T wird ein Knoten erstellt. Eine Kante zwischen zwei Knoten wird dann hinzugefügt, wenn eine Ähnlichkeit zwischen den beiden Sätzen be-

steht. Die Ähnlichkeit wird anhand einer Überschneidungs-Funktion bestimmt, welche die Anzahl der gemeinsamen Token als Basis nutzt. Um zu verhindern, dass lange Sätze einen Vorteil durch die Anzahl der Token haben, wird ein Normalisierungsfaktor eingeführt. Dieser berechnet sich aus dem Quotienten der sich überschneidenden Länge beider Sätze mit der jeweiligen Ursprungs-Satzlänge. Abbildung 2.4 visualisiert diesen Vorgang. Es gilt:

Gegeben seien zwei Sätze S_i und S_j aus jeweils N_i Worten:

$$S_i = w_1^i, w_2^i, ..., w_{N_i}^i$$

Die Ähnlichkeit von S_i und S_j ist definiert als:

$$Similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)}$$

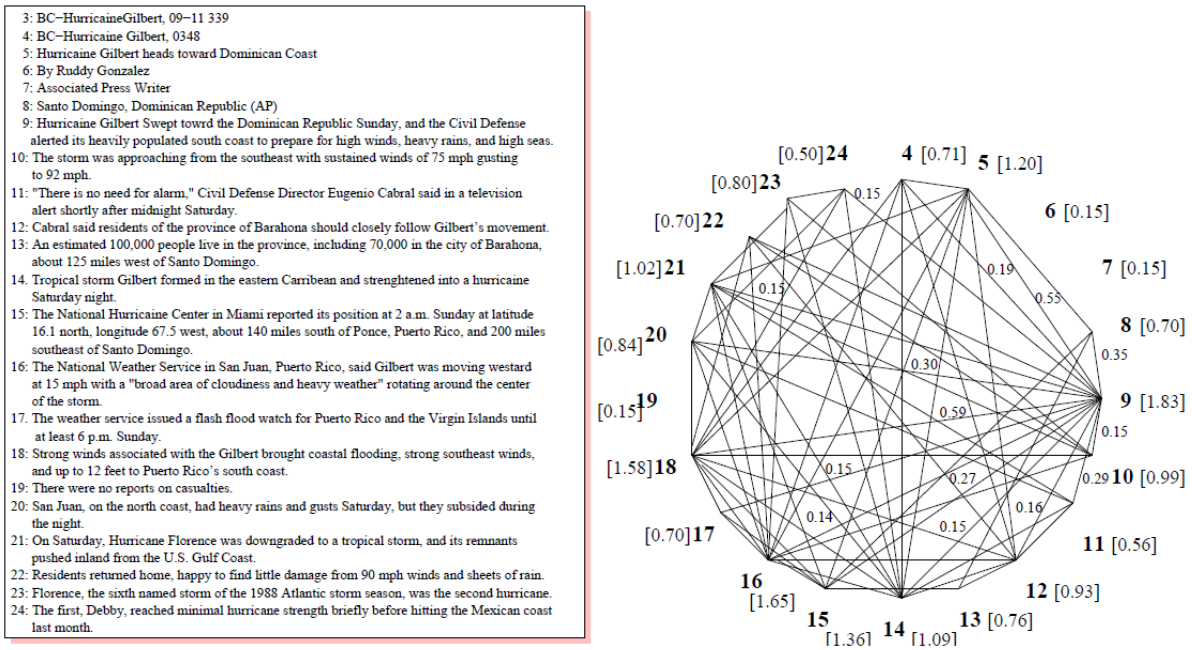


Abbildung 2.4.: Links: Original-Text nach Sätzen gegliedert. Rechts: Darstellung des Textes als TextRank-Graph (Mihalcea und Tarau 2004)

Die resultierende Zusammenfassung lässt sich vom Graphen anhand der am höchsten gewerteten Knoten ablesen (Top 4 Sätze):

> Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil De– fense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a ”broad area of cloudiness and heavy weather”rotating around the center of the storm. Strong winds associated with Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico's south coast.

2.4. Vor- und Nachteile

Vorteile (vgl. Tandel u. a. 2016, S. 2):

- Extractive Text Summarization ist einfacher und schneller als sein abstrakter Kontrahent.
- Die Terminologie des Textes wird übernommen und es gibt keine "Verständlichkeitsprobleme" beim Zusammenfassen. Dieser Punkt kann vor allem bei der Abstract Text Summarization ein großes Problem sein.

Nachteile:

- Extrahierte Sätze können unverhältnismäßig lang sein. (vgl. Gupta und Lehal 2010b, S. 1)
- Resultierende Zusammenfassung in Form und Wortlaut sind weit weg von einer menschlichen Zusammenfassung. (vgl. Hou, Hu und Bei 2018, S. 1)

Im Folgenden nach (Hou, Hu und Bei 2018, S. 259) zitiert nach J. Lin (2009) und Carenini, Chi und Cheung (2008):

- Informations-Redundanz entsteht durch das Extrahieren ganzer Sätze. Komprimierung ist durch eigene Worte nicht möglich.
- Information können über mehrere Sätze hinweg verteilt sein, was durch das Extrahieren von einzelnen Sätzen zerstört wird (Informations- und Kontextverlust).
- Widersprüchliche Informationen können falsch dargestellt werden.
- Extrahieren kann zu Zusammenhangsverlust führen, zum Beispiel im Fall von Anaphern. Pronomen stehen als Ersatz für Nomen und verlieren ihren Rückbezug, wenn der vorangehende Kontext/Satz nicht extrahiert wird. Dies führt zu unleserlichen Sätzen oder schlimmer: falsch zusammengewürfelten Kontexten.

2.5. Zusammenfassung

Extractive Text Summarization ist die älteste Form der automatischen Text Zusammenfassung und bis heute ein relevantes Themenfeld. Es birgt sowohl Vor- als auch Nachteile. Im Zuge dieser Ausarbeitung werden diese, anhand der Abgeordneten-Reden des Deutschen Bundestags, mithilfe des vorgestellten Algorithmus' TextRank untersucht und verglichen.

3. Abstractive Text Summarization

3.1. Ablauf

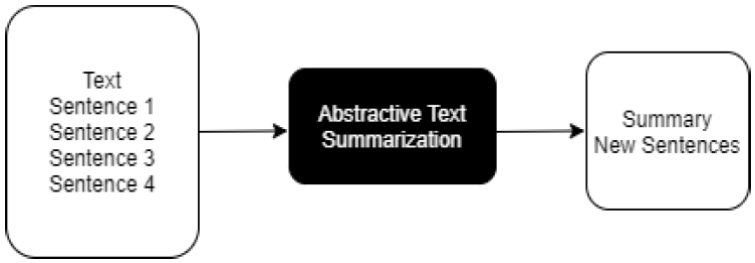


Abbildung 3.1.: Abstractive Text Summarization (vgl. Tahseen u. a. 2022, S. 103)

Abbildung 3.1 zeigt den Ablauf einer Abstractive Text Summarization und veranschaulicht die Aussage aus der Einleitung, dass die resultierende Zusammenfassung ein eigenständiger Text ist, der aus neu geformten Sätzen besteht.

Wie auch für die extrahierende Variante gibt es für die abstrakte Zusammenfassung von Texten mehrere Methoden und Ansätze (siehe Abbildung 3.2). Im Zuge dieser Ausarbeitung liegt der Fokus jedoch nur auf dem state-of-the-art, also auf der Transformer-based Encoder-Decoder-Technologie (In Abbildung 3.2 unter ”Deep-Learning Based“ eingeordnet), welche in Abschnitt 3.3 erläutert wird.

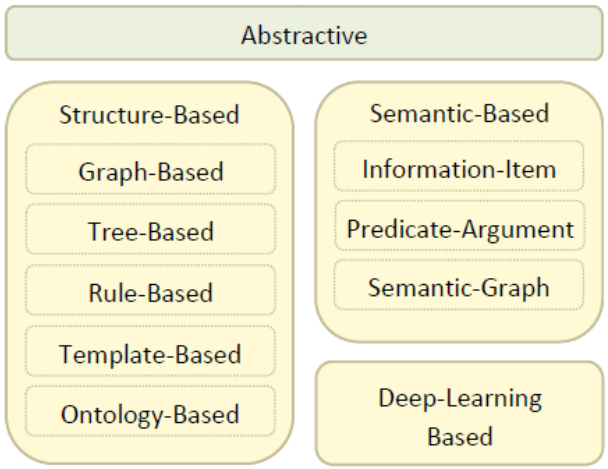


Abbildung 3.2.: Methoden von Abstractive Text Summarization (ergänzt durch d. Verf., El-Kassas u. a. 2021, S. 8)

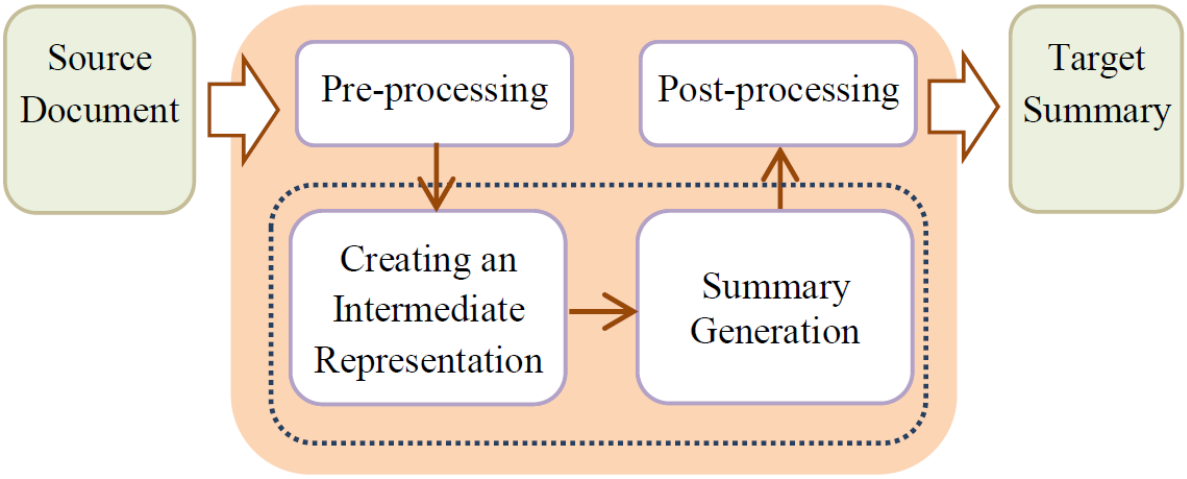


Abbildung 3.3.: Architektur einer Abstractive Text Summarization (El-Kassas u. a. 2021, S. 13)

3.2. Architektur

Analog zum Abschnitt 2.2 kann auch die Abstractive Text Summarization über eine grundlegende Architektur modelliert werden. Nach El-Kassas u. a. (vgl. 2021, S. 8) besteht die System-Architektur (siehe 3.3) für Abstractive Text Summarization aus vier Teilen:

1. **Pre-processing**: Beinhaltet unter anderem das Prüfen der Textlänge und möglicherweise Kürzen des Textes und das Unterteilen in einzelne Sätze (Analog zur Extractive Text Summarization).

2. **Creating an Intermediate Representation of Text**: Darstellen des Textes als numerische Datenstruktur zur normalisierten Weiterverarbeitung in den Transformern (Vektoren).

3. **Summary Generation**: Generieren einer menschenähnlichen, abstrakten Zusammenfassung auf Basis von NLP-bezogenen Techniken (vgl. Chitrakala u. a. 2018, S. 150).

4. **Post-processing**: Überführen der generierten Zusammenfassung in einen finalen, leserlichen Text.

3.3. Transformer

Das Paper ”Attention Is All You Need“ (Vaswani u. a. 2017) stellte eine neue Form von Neural Network Architektur vor: den Transformer. Bis dahin basierten die state-of-the-art Sequence-to-sequence (Seq2Seq) Modelle auf Recurrent Neural Networks. Diese sind jedoch nur sehr schwer zu trainieren (Optimierung ist NP-Vollständig) und weisen Probleme mit langfristigen Kontextverlusten bzw. Abhängigkeiten auf (vgl. Lipton, Berkowitz und Elkan 2015, S. 13).

Transformer versuchen diese Probleme anhand einer Self-Attention Schicht (Siehe Unterabschnitt 3.3.3) zu lösen.

Die folgende Aufarbeitung zum Transformer stützt sich auf Vaswani u. a. (2017) und Alammar (2018).

3.3.1. Aufbau

Der Transformer besteht aus einem Encoder und einem Decoder (Siehe Abbildung 3.4 jeweils linke (Encoder) und rechte (Decoder) Hälfte). Diese zwei Komponenten wiederum bestehen jeweils aus einem Stack von $N = 6$ identischen Schichten, die in Abbildung 3.4 immer nur einzeln dargestellt sind und durch ein "$N\times$" angedeutet werden.

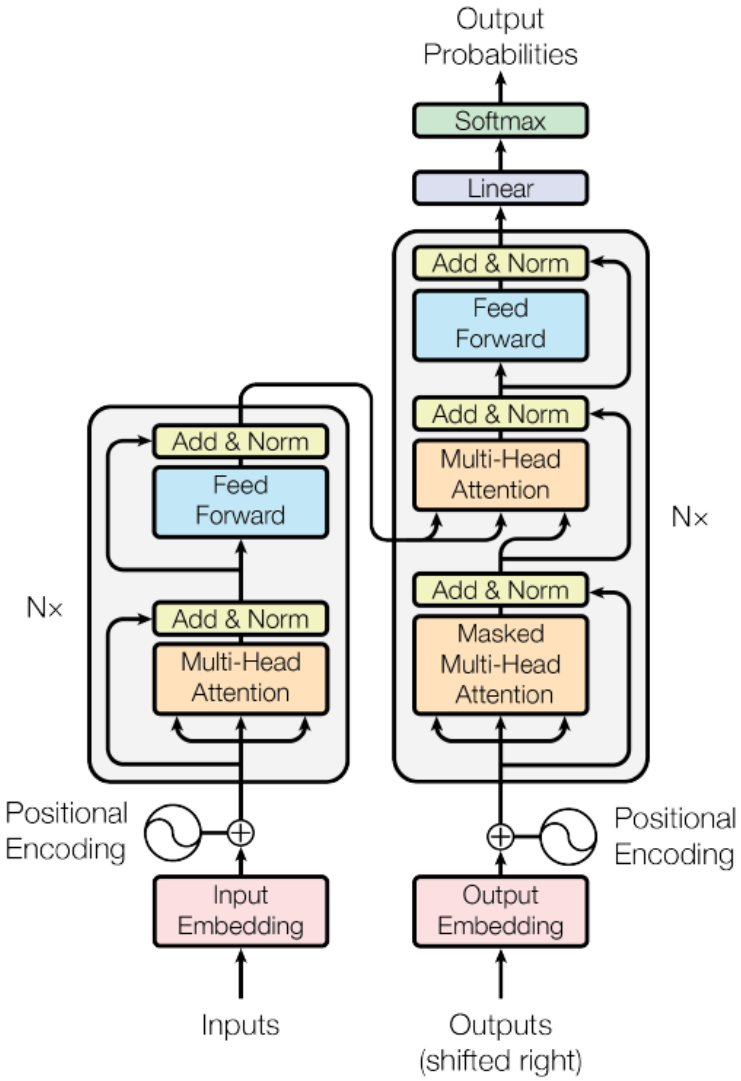


Abbildung 3.4.: Architektur des Transformer Models (vgl. Vaswani u. a. 2017, S. 3)

3.3.2. Encoder

Der Encoder entzieht dem Input wichtige Eigenschaften, welche dem Decoder in Form eines Kontext-Vektors übergeben werden, damit dieser dann den Output (zB. eine Zusammenfassung) generieren kann. Jede Schicht im Encoder besteht aus zwei Bausteinen, die jeweils durch eine ”Add & Norm“ Schicht quittiert werden:

1. Multi-Head Attention bzw. Self-Attention
2. Feed Forward Neural Network

Die Kommunikation zwischen diesen Bausteinen geschieht über Vektoren. Dazu muss der Input, wie es im NLP üblich ist, in eine numerische Darstellung überführt werden. Dies passiert durch einen Word Embedding Algorithmus, welcher Worte in Vektoren darstellt. Ignoriert man vorerst das Positional Encoding, so visualisiert Abbildung 3.5 die resultierende Abfolge.

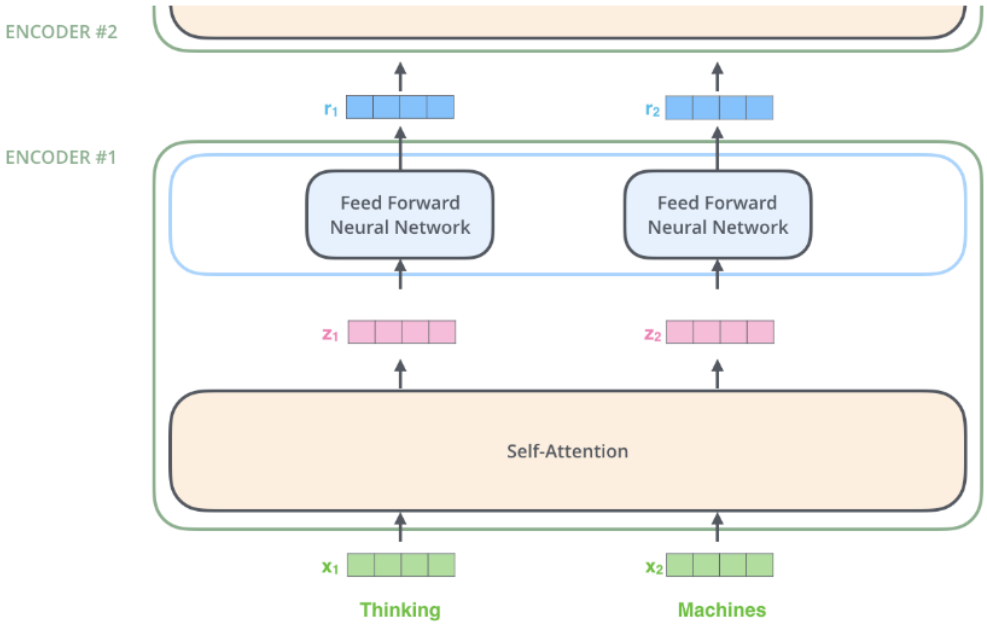


Abbildung 3.5.: Encoder (vgl. Alammar 2018)

Es ist zu sehen, dass jedes Wort seinen eigenen Pfad im Encoder durchläuft. Zwischen diesen Pfaden herrschen Abhängigkeiten in der Self-Attention Schicht, jedoch nicht im Feed Forward Neural Network, was dazu führt, dass jedes Wort parallel durch den Encoder fließen kann. Um den Encoder/Decoder und damit den Transformer zu verstehen, muss ”Self-Attention“ erläutert werden.

3.3.3. Self-Attention

Im Folgenden erläutert nach Bloem (2020).

Vereinfacht

Gegeben sei der Satz:

> This restaurant was not too terrible, it was actually good!

Für eine korrekte Zusammenfassung ist es wichtig zu verstehen, was das Personalpronomen ”it“ substituiert, da dies unabdinglich für den Kontext ist. Außerdem ist es essentiell, dass ”not“ auf ”terrible“ eine negierende Funktion hat und beides in Kombination eine Aussage über ”restaurant“ trifft. Self-Attention ermöglicht, diese Verbindungen zu erkennen und zu speichern.

Ablauf

Der erste Schritt zur Kalkulation von Self-Attention ist das Einführen von drei neuen Vektoren pro embedded Input-Wort (”This“, ”restaurant“, usw.). Dies umfasst jeweils einen:

- Query-Vektor
- Key-Vektor
- Value-Vektor

Diese Vektoren werden anhand einer Matrix-Multiplikation des Input-Worts mit drei Matrizen (welche während des Trainings adjustiert werden) berechnet. Die Bedeutung dieser Vektoren wird später ersichtlich.

Der zweite Schritt ist das Berechnen eines Werts, der bestimmt, wie relevant andere Worte des Inputs für das jetzige Wort sind. Um an den vorherigen Beispielsatz anzuknüpfen: Wie wichtig ist zum Beispiel ”restaurant“ für das Encoding von ”it“ und lohnt es sich, dass dieses Wort mitbetrachtet wird? Dieser bestimmende Wert wird durch das Skalarprodukt des jetzigen Query-Vektors mit den jeweiligen Key-Vektoren der anderen Worte errechnet.

Der dritte und vierte Schritt besteht darin, diese Werte durch die Wurzel der Dimension der Key-Vektoren zu teilen ($\sqrt{d_k}$) und die Resultate durch eine softmax Operation zu führen. Softmax normalisiert die Werte, sodass die Summe eins beträgt.

In den letzten beiden Schritten wird der Value-Vektor mit den eben errechneten softmax-Werten multipliziert (Dies filtert irrelevante Worte aus und verstärkt jene, die relevant sind.), um danach die Summe über alle gewichteten Value-Vektoren zu bilden, die dann den Output-Vektor für ein Wort in der Self-Attention Schicht darstellt.

Veranschaulichung

Abbildung 3.6 visualisiert diesen Ablauf. Es ist zu sehen, dass der Self-Attention Wert für das Wort ”terrible“ berechnet wird. Wie in **Schritt zwei** erläutert, wird dazu das Skalarprodukt des Query-Vektors q_i von ”terrible“ mit dem jeweiligen Key-Vektor k_j aller anderen Worte (hier gerade ”not“) gebildet. Das Resultat wird durch das goldfarbige Skalar dargestellt, welches noch **Schritt drei und vier** durchläuft, bevor es dann mit dem jetzigen Value-Vektor

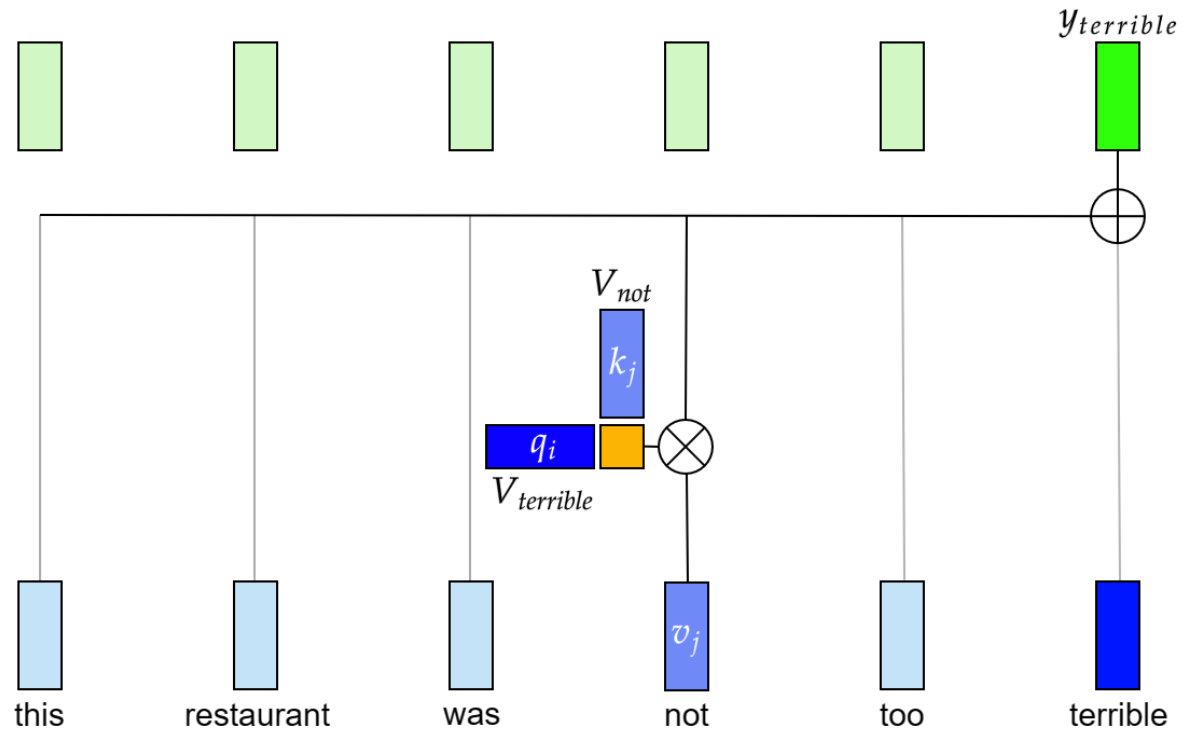


Abbildung 3.6.: Self-Attention (Die dargestellte Vektor-Multiplikation gilt für alle Input-Wörter, wurde aber zur besseren Übersicht nur einmal dargestellt)

v_j von ”not“ multipliziert wird. Zum Schluss wird die Summe über alle gewichteten Value-Vektoren jedes Wortes gebildet, was den finalen Self-Attention-Vektor für ”terrible“ bildet.

Ohne den Self-Attention Mechanismus würde jedes Wort nur individuell zum Output-Wert beitragen (Bag-of-words), was in Abbildung 3.6 dazu führen könnte, dass ”terrible“ einen negativen Kontext um ”restaurant“ kreiert, ohne dabei das negierende ”not“ zu betrachten.

Um diesen Vorgang numerisch umzusetzen, wurde der Vektor von ”not“ so trainiert, dass dieser ein möglichst kleines Skalarprodukt mit dem Vektor von ”terrible“ formt. Falls beide Worte in einem Satz vorkommen, wird die Wahrscheinlichkeit minimiert, dass ”terrible“ eine negative Wirkung auf ”restaurant“ hat, da die Wahrscheinlichkeit besteht, dass ”not“ diese Wirkung negiert. **Die Vektoren Key, Query und Value** werden gebildet, damit ein Vektor pro Wort verschiedene Formen und Werte für verschiedene Aufgaben annehmen kann und diese aufgabenspezifisch angepasst bzw. trainiert werden können.

Wieso funktioniert das?

Wie erzeugt ein Skalarprodukt von zwei numerischen Vektoren einen Wert, der bestimmt, wie relevant Wörter füreinander sind? Abbildung 3.7 zeigt diesen Vorgang anhand eines Beispiels.

Gegeben sei ein Film f. Dieser wird durch seinen Feature-Vektor V_f beschrieben. Je blasser

die Farbe, desto geringer ist die Ausprägung der Eigenschaft e_i des Vektors. Jeder Index steht für eine Eigenschaft bzw. ein Feature (”gruselig“, ”witzig“ und ”dramatisch“). Der Wert im Index i beschreibt die Ausprägung der Eigenschaft im Film. Film f ist also sehr gruselig, mit vielen humoristischen Stellen, jedoch nicht dramatisch.

Analog gibt es einen Nutzer n, der abwägen muss, ob Film f für ihn geeignet ist. Nutzer n mag Komödien sehr und ist auch den anderen Eigenschaften nicht abgeneigt.

Wenn jetzt das Skalarprodukt der beiden Feature-Vektoren berechnet wird, tragen die Eigenschaften, die einen hohen Wert besitzen, auch viel zum $score$ bei. Ein Nutzer n, der Komödien mag, erhält einen höheren $score$ mit Film f, als ein Nutzer m, der gar keine Komödien mag. Noch schlechter hätte es Nutzer u, der ausnahmslos nur Dramen mag, da diese hohe Ausprägung durch die niedrige Eigenschaft des Films beim Multiplizieren minimiert wird. Der $score$ von Film f für Nutzer u wäre dementsprechend klein.

Im Kontext der Self-Attention Schicht bedeutet dies, dass der Key-Vektor von ”not“ beim Errechnen des Skalarprodukt mit dem Query-Vektor von ”terrible“ einen möglichst kleinen Wert erzeugen soll (wird beim Trainieren angelernt). Dies steht repräsentativ dafür, dass ”not“ das Wort ”terrible“ in seiner Wertigkeit einschränkt und minimiert. ”[T]errible“ hat dann keine große Auswirkung mehr auf ”restaurant“, wie es im Satz auch der Fall ist.

Formell

In der Praxis werden die Vektoren nicht einzeln berechnet, sondern zu Matrizen zusammengefügt und mit diesen der Self-Attention Wert bestimmt. Für eine Query-Matrix Q, Key-Matrix K und Value-Matrix V ist der Output definiert als (vgl. Vaswani u. a. 2017, S. 4):

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

Multi-Head Attention

Der letzte Baustein zur Self-Attention steckt in der Multi-Head Architektur. Die Notwendigkeit dieser wird in Abbildung 3.8 dargestellt. Worte verweisen auf andere Worte in verschiedenen Relationstypen. Das Wort ”too“ hat eine moderierende Wirkung auf ”terrible“, während ”not“ eine negierende besitzt. ”[T]errible“ ist wiederum eine Eigenschaft von ”restaurant“. Das Modell muss diese verschiedenen Relationen in einer Self-Attention Operation beachten, daher teilen wir die Operation in verschiedene ”Heads“ oder Köpfe auf. Jeder Kopf ist dabei eine eigene Self-Attention Schicht, die parallel ausgeführt und am Ende mit anderen zusammengefügt wird.

Formell bedeutet dies (vgl. Vaswani u. a. 2017, S. 5):

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

mit

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

, wobei die Projektionen aus Parameter-Matrizen $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ und $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ bestehen.

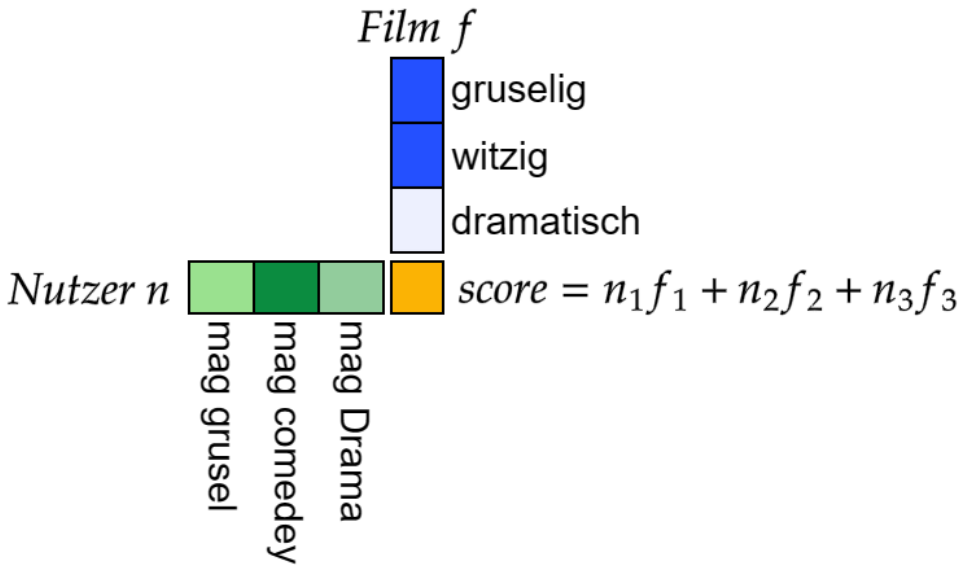


Abbildung 3.7.: Effekt des Skalarprodukts

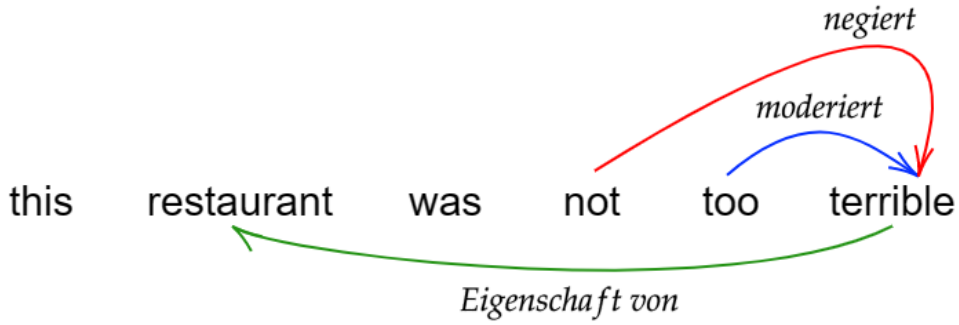


Abbildung 3.8.: Multi-Head Attention

3.3.4. Add & Norm

Abbildung 3.4 zeigt mehrere ”Add & Norm“ Bausteine, vor allem nach den Neural Network Schichten. Diese tragen zum effizienteren und schnelleren Trainieren bei.

Normalization

Es sei ein Neural Network N gegeben. Die Daten, die in N übergeben werden sollen, bestehen aus zwei Features:

1. Alter einer Person
2. Jahresgehalt einer Person

Ein Feature-Vektor könnte dann so aussehen:

$$\begin{bmatrix} 35 \\ 70.000 \end{bmatrix} \tag{3.1}$$

Es ist zu sehen, dass die Werte des Features ”Alter“ in der Regel im Intervall $[0, 100]$ liegen, während das Features ”Jahresgehalt“ Werte von $[0, 1.000.000]$ und noch mehr annehmen kann. Die beiden Feature haben sehr verschiedene Wertebereiche, was dazu führt, dass N nur sehr schwer die optimalen Gewichte berechnen kann, um die Kosten oder Fehler zu minimieren. Außerdem ruft es das Unstable Gradient Problem hervor. Eine simple Lösung wäre es, den Input einmal zu normalisieren, jedoch besteht die Chance von Unstable Gradients in tieferen Schichten des Netzwerks weiterhin.

Layer Normalization

Ba, Kiros und Hinton (2016) stellten die Layer Normalization vor. Bevor der Output einer Aktivierung als Input an das nächste Layer übergeben wird, wird dieser normalisiert. Abbildung 3.9 zeigt ein Beispiel für einen Input mit vier Features und einer Batch-Größe von drei. Die Normalisierung findet immer entlang der Layers statt.

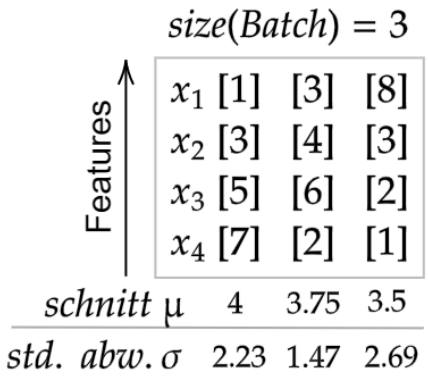


Abbildung 3.9.: Layer Normalization Beispiel (vgl. C 2022, ergänzt durch d. Verf.)

Es gilt nach Ba, Kiros und Hinton (2016):

$$\mu^l = \frac{1}{H} \sum_{i=1}^{H} a_i^l$$

und

$$\sigma^l = \sqrt{\frac{1}{H} \sum_{i=1}^{H} (a_i^l - \mu^l)^2}$$

mit $H =$ Dimension Feature-Vektor in

$$y = \frac{x_i - \mu_l}{\sqrt{\sigma_l^2 + \epsilon}} * \gamma + \beta$$

wobei γ und β lernfähige Parameter darstellen und y der normalisierte Output ist.

Add

Das ”Add“ in ”Add & Norm“ steht für eine Residualverbindung. Vorgestellt wurde es von He u. a. (2015) mit dem ResNet Modell. Abbildung 3.10 visualisiert diesen Vorgang.

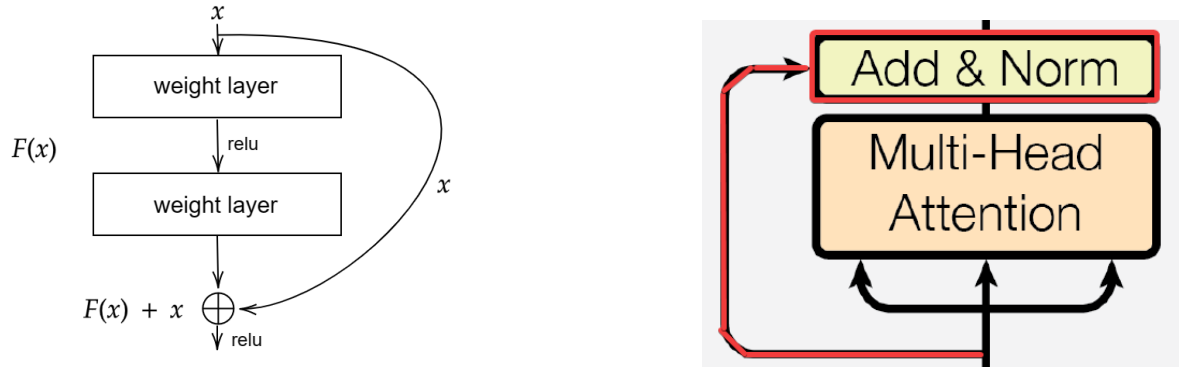


Abbildung 3.10.: Links: Residualverbindung; Rechts: Verbindung im Transformer Modell (vgl. Vaswani u. a. 2017, ergänzt durch d. Verf.)

Über die rot markierte Verbindung wird eine unbearbeitete Kopie der Input-Matrix an den Ausgang der Self-Attention Schicht geschickt, damit an dieser der Output mit der Kopie addiert werden kann. Danach findet die Layer Normalization statt. Beides versucht das Vanishing Gradient Problem zu lösen und das Trainieren des Modells zu vereinfachen (vgl. He u. a. 2015, S. 6).

3.3.5. Feed Forward Neural Network

Jedes Wort des Outputs der Self-Attention Schicht durchläuft jeweils ein eigenes Feed Forward Neural Network. Dieses besteht aus zwei einfachen linearen Transformationen mit einer ReLU Aktivierung dazwischen. Es gilt nach Vaswani u. a. (2017, S. 5):

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

3.3.6. Positional Encoding

Der Transformer nutzt Self-Attention Schichten, die, im Gegensatz zu RNNs, keine ordinalen Abhängigkeiten speichern. Das bedeutet, dass die beiden Sätze

> Max ate a banana.
> A banana ate Max.

die gleiche Semantik für den Transformer hätte. Deswegen werden Positionsdaten in den embedded Input-Vektor injiziert. Dies geschieht durch Vektor-Addition. Für die Kodierung einer Wortposition gilt nach Vaswani u. a. (2017, S. 6):

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right)$$

, wobei pos die Position und i die Dimension des Worts ist.

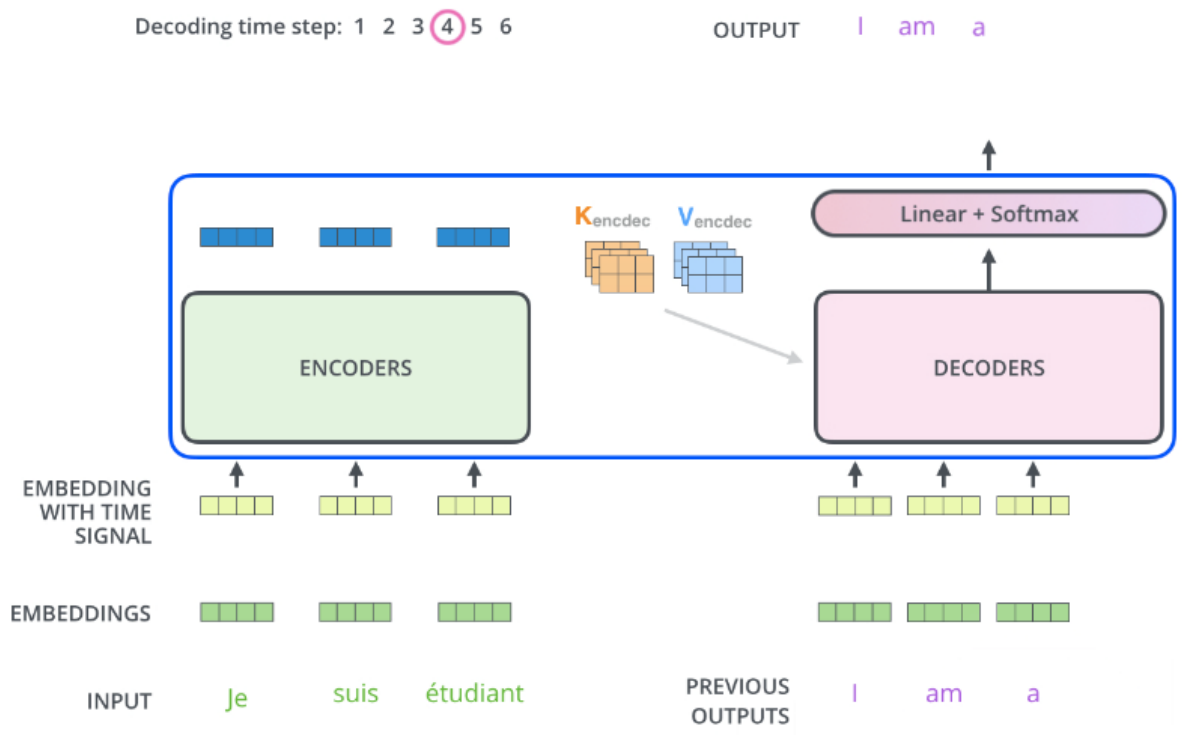


Abbildung 3.11.: Ablauf Decoder (vgl. Alammar 2018)

3.3.7. Decoder

Nachdem der Encoder im Detail besprochen wurde, sind auch die meisten Funktionen des Decoders erläutert. Die generelle Aufgabe des Decoders besteht darin, die nächste Sequenz, also das nächste Wort, zu prognostizieren. Abbildung 3.11 visualisiert diesen Vorgang.

Der Decoder nimmt als Input die Kontext-Vektoren K und V des Encoders und zusätzlich seinen eigenen Output aus dem vorherigen Decoding-Schritt an. In Abbildung 3.11 ist dies durch ”PREVIOUS OUTPUTS: I am a“ dargestellt. Der jetzige Schritt des Decoders wäre es demnach, das Wort ”student“ zu prognostizieren und als Output auszugeben. Der letzte Schritt wäre es dann, auf Basis des eigenen Outputs ”I am a student“ und dem Kontext des Encoders das Satzende ”<end of sentence>“ auszugeben.

Masked Multi-Head Attention

Die Self-Attention Schichten im Decoder operieren leicht unterschiedlich im Vergleich zur Encoder-Seite. Während auf der Encoder-Seite die Self-Attention Schicht jede Position betrachten soll, wird auf der Decoder-Seite der Informationsfluss in die ”Zukunft“ unterbunden. Dies passiert durch Ausmaskieren aller illegalen Positionen durch $-\infty$. Das vorgestellte Transformer-Modell ist ein auto-regressive Transformer, was bedeutet, dass die Vorhersage des nächsten Worts nur auf Basis der letzten Worte geschieht. Wenn der Rest, also die ”Zukunft“, schon bekannt wäre, gäbe es nichts vorherzusagen. Daher muss dieser Teil maskiert

werden.

Am Beispiel von Abbildung 3.11 erläutert: Wenn der Decoder von Anfang an den vollen Output ”I am a student <end of sentence>“ betrachten darf, was muss er dann vorhersagen? Ein Trainingseffekt kann so nicht stattfinden.

3.3.8. Zusammenfassung

Der Transformer ist eine Neural-Network Architektur, die paralleles Trainieren mithilfe von Self-Attention auf großen Datensätzen für (primär) Sprachmodelle erlaubt. Transformer finden ihren Einsatz in Sprachmodellen wie T5, PEGASUS oder ChatGPT, bei denen für den jeweiligen Use-Case (zB. PEGASUS=Abstract Text Summarization), Finetuning auf den jeweiligen Modellen stattfindet, sodass der Transformer nicht einfach nur Sequenzen voraussagen kann, sondern Zusammenfassungen kreiert oder Fragen beantwortet (ChatGPT).

3.4. Modelle

In Abschnitt 3.3 wurde der Transformer im Detail erläutert. Dieser bildet die Basis für die jetzt in Folge behandelten Modelle zur Abstract Text Summarization. Im Umfang dieser Ausarbeitung werden zwei state-of-the-art Modelle betrachtet und im Folgenden angewandt:

- BART
- PEGASUS

Bevor diese zwei Modelle erklärt werden, müssen noch die Begriffe ”Pretraining“ (zu Deutsch ”Vortrainieren“) und ”Finetuning“ (zu Deutsch ”Feinschliff“) im Kontext von Transformern erläutert werden.

3.4.1. Pretraining und Finetuning

Um ein Sprachmodell vermeintlich schlau und für einen Use-Case einsetzbar zu machen, wird das Training typischerweise in zwei Schritte aufgeteilt.

Pretraining

Beim Pretraining wird dem Transformer die generelle menschliche Sprache antrainiert, ohne diesen auf einen domain-spezifischen Use-Case abzurichten. Es wird eine Basis geschaffen, mit welcher der Transformer dann auf die spezifische Aufgabe (Text Summarization, Language Translation, etc.) abgerichtet werden kann. Mathematisch betrachtet wird den Gewichten im Neural-Network ein Startwert gegeben, damit diese nicht jedes mal bei null anfangen müssen. Das Abrichten ist dann das Finetuning.

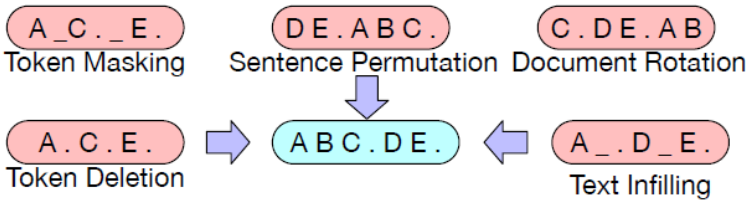


Abbildung 3.12.: Verrausch-Techniken (vgl. Lewis u. a. 2019, S. 3)

Finetuning

Im Finetuning lernt das Modell den speziellen Use-Case. Bevor ein Modell passable Lösungen präsentieren kann, bedarf es großer Mengen an Testdaten. Im Falle von PEGASUS wurde das Pretraining auf den Datensätzen C4 (350 Millionen Websites) und HugeNews (1,5 Milliarden Artikel) (vgl. Zhang u. a. 2019, S. 4) durchgeführt, was dem Modell eine Lernbasis vermittelt hat. Dadurch bedarf es nach Zhang u. a. (2019, S. 8) für state-of-the-art Resultate im Abstract Text Summarization-Bereich nur 1.000 beschriftete (labeled) Datensätze, um hochwertige Zusammenfassungen zu generieren.

3.4.2. BART

Lewis u. a. (2019) stellten einen neuen Autoencoder zum Vortrainieren von Seq2Seq Modellen vor: Den "Bidirectional and Auto-Regressive Transformer" (BART). BART nutzt die Standard-Transformer-Architektur[1] aus Abschnitt 3.3 und eine Technik zum Verrauschen ("noise") und Entrauschen ("denoise") des Input-Texts, um das Modell vorzutrainieren. Dieses Vortrainieren hat zwei Schritte (vgl. Lewis u. a. 2019, S. 1):

1. Der Input-Text wird durch eine Rauschfunktion geführt, welche den Text korrumpiert.
2. Das Seq2Seq Modell lernt, diesen korrumpierten Text wiederherzustellen.

Abbildung 3.12 zeigt die genutzten Methoden zum Verrauschen bzw. Korrumpieren eines Input-Texts. Es gilt im Folgenden nach Lewis u. a. (2019, S. 3–4):

- **Token Masking**: Zufällige Tokens werden maskiert (Durch [MASK] ersetzt).
- **Token Deletion**: Zufällige Tokens werden gelöscht. Hier muss das Modell nicht nur den Token wiederherstellen, sondern auch die Position.
- **Text Infilling**: Eine feste Anzahl sequentieller Token werden gelöscht. Das Modell muss Anzahl, Position und Inhalt wiederherstellen.
- **Sentence Permutation**: Die Position von Wörtern in den Sätzen wird verändert.

[1] Mit zwei Anpassungen: ReLU Aktivierung wird zu GeLU geändert und Parameter werden mit $\mathcal{N}(0, 0.02)$ initialisiert (vgl. Lewis u. a. 2019, S. 2)

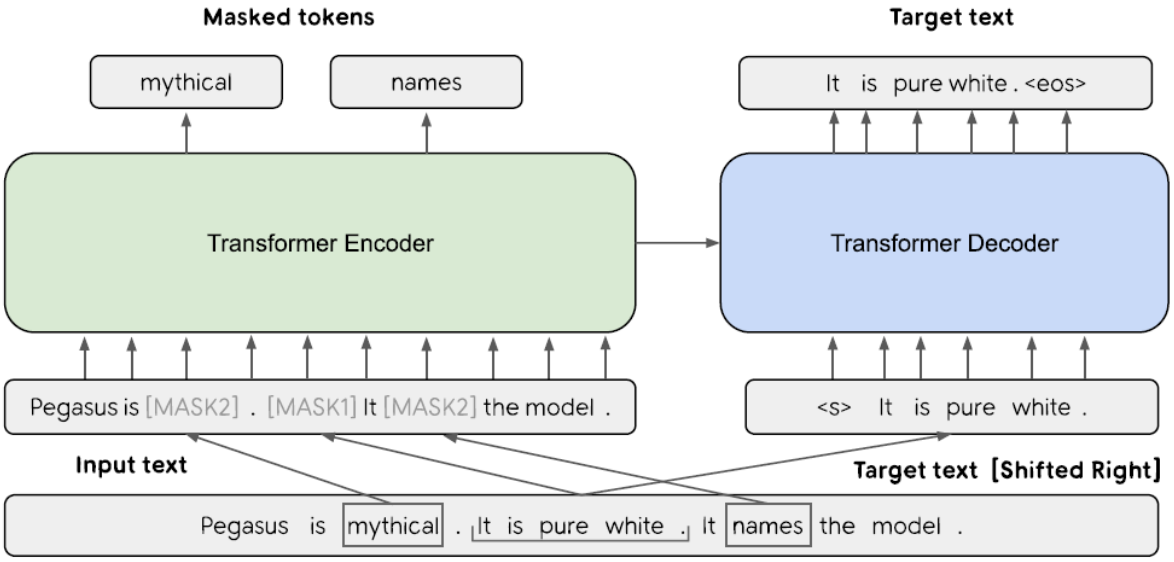


Abbildung 3.13.: PEGASUS Architektur (vgl. Zhang u. a. 2019, S. 1)

- **Document Rotation**: Ein Token wird zufällig gewählt und das Dokument wird so gedreht, dass der Token der Start ist. Das Modell muss so den richtigen Start des Dokuments finden.

Dieser Vorgang erlaubt BART ein unsupervised Pretraining, welches durch ein supervised Finetuning mit nur wenigen labeled Datensätzen auf verschiedene Use-Cases finalisiert werden kann.

3.4.3. PEGASUS

Zhang u. a. (2019) stellten eine Architektur vor, die schon das Pretraining auf Abstract Text Summarization-Aufgaben abstimmen soll. Dies soll durch ”Pre-training with Extracted Gap-sentences for Abstractive Summarization”(PEGASUS) auf den Transformern geschehen. Abbildung 3.13 visualisiert diesen Vorgang. Die Architektur von PEGASUS basiert auf dem Standard encoder-decoder Transformer aus Abschnitt 3.3 und besitzt Überschneidungen mit den Trainings-Techniken von BART. Bei PEGASUS werden Sätze des Input-Texts maskiert (In Abbildung 3.13 wird der Satz ”It is pure white.“ mit [MASK1] maskiert.) und auch einzelne Token (”mythical“ und ”names“). Gleichzeitig werden dem Decoder die extrahierten Sätze als Target übergeben, damit dieser dann, wie in Unterabschnitt 3.3.7 beschrieben, die Sätze wieder prognostizieren kann. Das Maskieren von einzelnen Token wurde im finalen Modell nicht übernommen, da dieses den Trainingsprozess nicht verbessert und sich hier als irrelevant herausstellt (vgl. Zhang u. a. 2019, S. 4).

Mit diesem unsupervised Pretraining erreichen PEGASUS-Modelle state-of-the-art Ergebnisse mit Finetuning auf nur 1000 Datensätzen (Siehe Abbildung 3.14).

Datasets	**XSum** mean (p-value)	**CNN/DailyMail** mean (p-value)	**Reddit TIFU** mean (p-value)
Experiment 1: pretrain comparison			
Human-written	3.0 (-)	3.1 (-)	3.2 (-)
$\text{PEGASUS}_{\text{LARGE}}$ (HugeNews)	**3.0** (0.6)	**3.6** (0.0001)	**3.2** (0.7)
$\text{PEGASUS}_{\text{LARGE}}$ (C4)	**3.1** (0.7)	**3.5** (0.009)	**3.1** (0.3)
$\text{Transformer}_{\text{BASE}}$	2.0 (3e-10)	**2.9** (0.06)	1.4 (5e-23)
Experiment 2: low resource			
Human-written	3.2 (-)	3.2(-)	3.3 (-)
$\text{PEGASUS}_{\text{LARGE}}$ (HugeNews) 10 examples	**2.8** (0.1)	**3.4** (0.007)	2.6 (0.006)
$\text{PEGASUS}_{\text{LARGE}}$ (HugeNews) 100 examples	**3.2** (0.5)	**3.4** (0.08)	2.1 (4e-8)
$\text{PEGASUS}_{\text{LARGE}}$ (HugeNews) 1000 examples	**3.4** (0.3)	**3.6** (0.07)	2.7 (0.01)
$\text{PEGASUS}_{\text{LARGE}}$ (HugeNews) full supervision	**3.4** (0.3)	**3.3** (0.1)	**2.8** (0.05)

Abbildung 3.14.: PEGASUS Ergebnisse auf einer Likert-Skala im Vergleich zu menschlichen Ergebnissen (vgl. Zhang u. a. 2019, S. 8).

4. Der Versuch

Die vorherigen Kapitel haben eine Einführung in Automatic Text Summarization geboten, mit Fokus auf den Methoden:

- TextRank (Extractive Text Sumamrization)
- BART (Abstractive Text Summarization)
- PEGASUS (Abstractive Text Summarization)

Es wurden die state-of-the-art Transformer im Bereich der Abstract Text Summarization im Detail erläutert, damit der technische Hintergrund für die jetzt folgende Anwendung und Auswertung gelegt wurde.

4.1. Ziel

Die erläuterten Modelle und Methoden werden auf ihre pratkische Anwendung hin untersucht. Es ist bekannt, dass PEGASUS und BART state-of-the-art Resultate hervorbringen (siehe Abbildung 3.14 oder Abbildung 4.1). Dies umfasst jedoch primär Datensätze aus dem englischsprachigen Raum (XSum[1], CNN/DailyMail[2], Gigaword[3]) mit starkem Bias auf News-Articles und Webseiten.

Wie ist der praktische Stand der Text Summarization im deutschsprachigen Raum unter Verwendung von Input-Texten, die nicht dem Format News-Artikel oder Webseite zuzuordnen sind?

4.2. Die Testdaten

Die zu untersuchenden Testdaten bestehen aus deutschsprachigen Reden der Abgeordneten des Deutschen Bundestags. Diese werden frei zugänglich vom Open-Data Service des Bundestags bereitgestellt[4]. Desweiteren haben Abrami u. a. (2022) einen mit SpaCy annotierten Korpus aus historischen und aktuellen politischen Texten im deutschsprachigen Raum erstellt, der ebenfalls Reden des Deutschen Bundestags bereitstellt. Zum Zwecke dieser Untersuchung wurden kanpp 4.000 Reden (41 Sitzungsprotokolle) der laufenden zwanzigsten Legislaturperiode (Stand 20.02.2023) untersucht und den drei Methoden/Modellen TextRank, PEGASUS und BART unterzogen. Die Skripte zur Durchführung wurden alle in Python geschrieben und die jeweiligen Transformer mithilfe der Huggingface Community implementiert.

[1] https://huggingface.co/datasets/xsum
[2] https://huggingface.co/datasets/cnn_dailymail
[3] https://huggingface.co/datasets/gigaword
[4] https://www.bundestag.de/services/opendata

R1/R2/RL	XSum	CNN/DailyMail	Gigaword
BERTShare (Rothe et al., 2019)	38.52/16.12/31.13	39.25/18.09/36.45	38.13/19.81/35.62
MASS (Song et al., 2019)	39.75/17.24/31.95	42.12/19.50/39.01	38.73/19.71/35.96
UniLM (Dong et al., 2019)	-	43.33/20.21/40.51	38.45/19.45/35.75
BART (Lewis et al., 2019)	45.14/22.27/37.25	**44.16**/21.28/40.90	-
T5 (Raffel et al., 2019)	-	43.52/**21.55**/40.69	-
$\text{PEGASUS}_{\text{LARGE}}$ (C4)	45.20/22.06/36.99	43.90/21.20/40.76	38.75/**19.96/36.14**
$\text{PEGASUS}_{\text{LARGE}}$ (HugeNews)	**47.21/24.56/39.25**	**44.17/21.47/41.11**	**39.12/19.86/36.24**

Abbildung 4.1.: Vergleich PEGASUS und BART auf verschiedenen Finetune-Datensätzen. (vgl. Zhang u. a. 2019, S. 7)

4.3. Die Durchführung

4.3.1. TextRank

Da TextRank eine Extractive Text Summarization-Methode darstellt, die auf Basis von Frequenzen und Abhängigkeiten die Sätze nur extrahiert, bedarf es hier keiner weiteren Voraussetzungen. Die Reden können dem TextRank einfach übergeben und die resultierenden Zusammenfassungen ausgewertet werden.

4.3.2. PEGASUS und BART

Bei den Transformern muss entschieden werden, welche Finetuning-Datensätze und Vorbereitungen am ehesten geeignet sind, um die besten Resultate im Versuch zu erzielen. Wie im Folgenden beschrieben, war dies mit vielen Problemen verbunden.

Finetuned Modelle

google/pegasus-xsum[5] ist ein PEGASUS-Modell, welches mit dem Datensatz XSum verfeinert wurde und verglichen mit CNN/DailyMail und Gigaword die besten Resultate erzielt (vgl. Zhang u. a. 2019, S. 7). Erste Tests zeigten jedoch schnell, dass der Transformer große Schwierigkeiten mit der deutschen Sprache aufweist. Die Zusammenfassungen waren alle ähnlich zerstückelt und unbrauchbar, selbst wenn die Reden auf eine optimale Länge von maximal 512 Token gekürzt wurden, wie im Folgenden Beispiel demonstriert wird.

Gekürzte Rede auf 474 Token:

> Das war aber jetzt eine Wiederholungstat. – Frau Präsidentin! Meine sehr geehrten Damen und Herren! Liebe Kolleginnen und Kollegen! Ein gutes Gesetz benötigt im Kern zwei Dinge: Es muss nicht nur gut gedacht, sondern auch gut gemacht sein. Der Entwurf, der uns vorliegt und den wir heute diskutieren, ist beides. Das Gesetz zur Stärkung der Aufsicht bei Rechtsdienstleistungen ist zunächst einmal dringend erforderlich und damit gut gedacht. Aktuell leidet der Bereich

[5] https://huggingface.co/google/pegasus-xsum

> der außeranwaltlichen Rechtsdienstleistungen unter einer enormen Zersplitterung in unserem Land; denn die Aufsicht über diesen Bereich obliegt den Justizbehörden der Bundesländer, die diese Aufsichtspflichten wiederum an 38 verschiedene Gerichte und Staatsanwaltschaften im Land übertragen haben. Dafür gibt es im Saarland eine Behörde, allein in Niedersachsen 14. Das heißt, es gibt aktuell keinerlei einheitliche Kontrolle und Aufsicht über einen hochsensiblen Bereich, der bundesweit exponentiell wächst. Genau so entstehen Rechtsunsicherheiten zulasten aller Beteiligten. Das gilt für die Verbraucherinnen und Verbraucher ebenso wie für die Schöpfer neuartiger, innovativer und wichtiger Angebote, unter anderem in dem rasant wachsenden Markt für den sogenannten Legal-Tech-Bereich.

Resultierende Zusammenfassung:

> Das Gesetz Strkung der Auf bei Rechtsdienstleistungen ist einmal dringend erforderlich und damit gutgedacht.

Es sind zwei Probleme zu erkennen:

1. Die Syntax ist falsch und der Satz dadurch insgesamt sinnlos.
2. Die deutschen Umlaute fehlen.

Andere Modelle wie *pegasus-cnn_dailymail*[6] und *google/pegasus-gigaword*[7] bringen keine Veränderung.

Im Gegenzug wurde auch BART mit *facebook/bart-large-cnn* (finetuned auf CNN/DailyMail) getestet und das Ergebnis ist überraschend (Original-Text ist identisch zu Unterunterabschnitt 4.3.2)

Resultierende Zusammenfassung:

> Frau Präsidentin: Ein gutes Gesetz benötigt im Kern zwei Dinge: Es muss nicht nur gut gedacht, sondern auch gut gemacht sein. Das Gesetz zur Stärkung der Aufsicht bei Rechtsdienstleistungen ist zunächst einmal dringend erforderlich.

Die Zusammenfassung ist extrahierend (vergleiche rote Sätze in Unterunterabschnitt 4.3.2), jedoch soll für diesen Versuch eine abstrakte Zusammenfassung generiert werden. Es stellt sich auch die Frage, warum die Zusammenfassung plötzlich extrahierend ist? All diese Beobachtungen bestätigen die Aussagen aus Unterabschnitt 4.3.3.

4.3.3. On the State of German (Abstractive) Text Summarization

Aumiller, Fan und Gertz (vgl. 2023, S. 3) haben in ihrem Paper "On the State of German (Abstractive) Text Summarization" bzgl. abstrakter Text Summarization im deutschsprachigen Raum drei Qualitätsprobleme formuliert, welche sich mit meinen Resultaten decken:

1. "Due to positional biases, text snippets may be directly copied from the beginning of the input text, constituting an extractive instead of an abstractive summary. Especially considering the computational requirements of neural systems being orders of magnitudes greater than simple extractive summarizers, this undermines the quality of neural text generations."

[6] https://huggingface.co/google/pegasus-cnn_dailymail
[7] https://huggingface.co/google/pegasus-gigaword

2. ”Generated outputs may contain (severe) syntactic errors, to the point of becoming illegible or hard to interpret.“

3. ”Semantic mistakes introduce factual errors, leading to incorrect conclusions from the summary alone. This problem is exacerbated for longer input documents, where a structured content understanding is necessary to maintain factual consistency.“

Desweiteren wurden sieben deutschsprachige Datensätze untersucht, die zum Finetuning auf Transformern öffentlich zugänglich sind. Die Resultate zeigten massive Qualitätsprobleme in den beliebtesten Datensätzen ML-SUM und MassiveSumm mit über 25% fehlerhafter Datensätze (vgl. Aumiller, Fan und Gertz 2023, S. 15). Außerdem wurden ”catastrophic failures“ gefunden und es wird beschrieben, dass die Modelle mit langen Input-Texten nicht umgehen können (vgl. Aumiller, Fan und Gertz 2023, S. 18). Es entsteht außerdem ein starker Bias aufgrund der hohen Dichte an News-Artikeln in den Datensätzen (vgl. Aumiller, Fan und Gertz 2023, S. 14).

4.3.4. Translate-then-Summarize (Trans-Sum)

Eine mögliche Lösung des Problems ist es, den deutschen Redetext erst ins Englische zu übersetzen, danach zusammenzufassen und die Zusammenfassung wieder ins Deutsche zurück zu übersetzen. Wan, Li und Xiao (2010) zeigten, dass Texte aus dem Englischen anhand einer Translation Quality Prediction Architektur erfolgreich auf chinesisch zusammengefasst werden konnten (vgl. Wan, Li und Xiao 2010, S. 923). Zhu u. a. (2020) bauten darauf auf und stellten einen Transformer vor, der die Übersetzung (Chinesisch -> Englisch) in den encoder-decoder Prozess inkorporierte und gute Ergebnisse präsentieren konnte (vgl. Zhu u. a. 2020, S. 8).

Aufgrund dieser Erkenntnisse und den ernüchternden Resultaten aus Unterunterabschnitt 4.3.2, die sich mit den Aussagen von Aumiller, Fan und Gertz (2023) decken, werde ich für meinen Versuch die deutschen Reden ins Englische übersetzen, dann mit PEGASUS und BART (TextRank kann direkt den deutschen Text nehmen) zusammenfassen und wieder zurückübersetzen. Eine zusätzlich notwendige Metrik muss dazu eingeführt werden, denn die Zusammenfassungen sind jetzt stark von der Übersetzungs-Qualität abhängig, die ebenfalls mitbetrachtet werden muss.

4.3.5. LaBSE

Feng u. a. (2020) stellten ein ”Language-agnostic BERT Sentence Embedding“ (LaBSE) Modell vor, welches auf das Erkennen von semantischen Ähnlichkeiten zwischen bilingualen Texten trainiert wurde. Für den Versuch wird das Python-Package ”translation-quality-estimator“ genutzt, welches mithilfe des besagten Modells einen Wert zwischen null und eins für zwei Input-Texte ausgibt. Je höher der Wert, desto ähnlicher soll die semantische Überschneidung der beiden Texte sein. Tabelle 4.1 zeigt zwei Beispielresultate. Für den Versuch wird ein Ähnlichkeitswert (basierend auf einer Kosinus-Ähnlichkeit) pro Rede berechnet. Dieser Wert bestimmt, wie gut die Übersetzung vom Deutschen ins Englische gelingt.

Tabelle 4.1.: Beispiel Ähnlichkeitswert LaBSE

Deutsch	Englisch	Ähnlichkeitswert
Das Wetter ist sehr angenehm.	Why are you here?	0.266
Was ist dein Name?	What is your name?	0.891

4.3.6. Übersetzung

Die Übersetzung der deutschen Reden ins Englische und wieder zurück wird mit OPUS-MT (Tiedemann und Thottingal 2020), einem state-of-the-art transformer-based neural machine translation (NMT) Modell, durchgeführt. Wie sich später herausstellt, werden die Übersetzungen von der eingeführten Ähnlichkeitsmetrik aus Unterabschnitt 4.3.5 im Durchschnitt als sehr gut (0.89) bewertet. Manuelle Stichproben bestätigen diese Einschätzung.

4.3.7. Finale Modelle

In Unterabschnitt 4.3.2 wurden mehrere verfeinerte Modelle auf deutsche Reden getestet und als unbrauchbar eingestuft. Die Lösung des Problems war, die deutschen Reden ins Englische zu übersetzen und diese dann zusammenfassen zu lassen. Testdurchläufe zeigten klare Verbesserungen, jedoch nicht die erhofften Resultate einer flüssigen, abstrakten Zusammenfassung (siehe Tabelle 4.2).

Die Beispiel-Zusammenfassungen wurden auf der englischen Übersetzung des Textes aus Unterabschnitt 4.3.2 durchgeführt:

> Madam President, ladies and gentlemen, this was a repeat performance. Dear colleagues! A good law essentially requires two things: it must not only be well thought out, but also well made. The draft before us, which we are discussing today, is both. The Act to Strengthen Oversight of Legal Services is, first of all, urgently needed and thus well thought out. Currently, the area of non-attorney legal services suffers from enormous fragmentation in our country; because supervision of this area is the responsibility of the judicial authorities of the federal states, which in turn have delegated these supervisory duties to 38 different courts and public prosecutors' offices in the country. There is one authority for this in Saarland, and 14 in Lower Saxony alone, which means that there is currently no uniform control and supervision whatsoever over a highly sensitive area that is growing exponentially nationwide. This is precisely how legal uncertainties arise, to the detriment of all involved. This applies to consumers as well as to the creators of new, innovative and important offerings, including in the rapidly growing market for the so-called legal tech sector.

Es ist weiterhin zu beachten, dass diese Resultate auf einer künstlich gekürzten Rede (auf knapp 512 Token) erreicht wurden und nicht den Ist-Stand der Bundestagsreden widerspiegeln. Der Durchschnitt liegt dort bei 688 Tokens pro Rede.

Die Ergebnisse sind wie folgt zu deuten:

- **google/pegasus-xsum** versucht eine eigenständige, abstrakte Zusammenfassung zu generieren, scheitert jedoch an allen Stellen. Die Rede ist weder von der (Ex-)Kanzlerin Angela Merkel, noch fand die Rede an einem Mittwoch statt, noch gibt sie irgendwelche nennenswerten Informationen aus dem Originaltext wieder.

Tabelle 4.2.: Beispiel-Ergebnisse Testdurchlauf (1) auf gekürzten Reden

Modell	Zusammenfassung
google/pegasus-xsum	The full text of Chancellor Angela Merkel's speech to the Bundestag on Wednesday:
google/pegasus-cnn_dailymail	There is currently no uniform control and supervision whatsoever over a highly sensitive area that is growing exponentially nationwide .<n>This applies to consumers as well as to the creators of new, innovative and important offerings, including in the rapidly growing market for the so-called legal tech sector .
facebook/bart-large-cnn	The Act to Strengthen Oversight of Legal Services is urgently needed and thus well thought out. Currently, the area of non- attorney legal services suffers from enormous fragmentation. This is precisely how legal uncertainties arise, to the detriment of all involved. This applies to consumers as well as to the creators of new, innovative and important offerings.
google/pegasus-gigaword	good law requires well thought out and well made law requires well thought out and well made law requires well thought out and well made law requires well thought out

- **google/pegasus-ccn_dailymail** und **facebook/bart-large-cnn** bringen eine grammatikalisch korrekte und lesbare Zusammenfassung hervor, jedoch besitzt auch diese wieder sehr viele extrahierende Sätze mit nur marginalen Eigenformulierungen. *google/pegasus-ccn_dailymail* speist außerdem unbekannte Textformatierungen ("<n>") in die Zusammenfassung ein, die eine nachbearbeitende Textsäuberung mit sich ziehen würde.
- **google/pegasus-gigaword** scheitert noch mehr als google/pegasus-xsum und kreiert eine komplett sinnfreie Zusammenfassung.

Meine Vermutung ist, dass die Modelle einen starken Bias auf News-Artikeln und Webseiten-Inhalte haben und diese mit den dialogartigen Formen der Bundestagsreden nicht gut arbeiten können.

4.3.8. SAMSum

SAMSum ist ein labeled Datensatz, bestehend aus 16.000 Chatverläufen samt derer Zusammenfassungen (vgl. Gliwa u. a. 2019). Bundestagsreden besitzen diskutierende Elemente mit gegenseitigen Ansprachen. Wenn man alle Reden eines Tagesordnungspunkts hintereinander schalten würde, so ähnelte die Form der einer Diskussion oder eines Chatverlaufs. Meine These ist, dass PEGASUS und BART, verfeinert auf dem SAMSum Datensatz, bessere Ergebnisse erzielen als auf den bekannten News- und Webseiten-basierten Datensätzen. Die These

hat sich, sogar auf den vollständigen Reden, nach ersten Tests bewahrheitet (siehe Tabelle 4.3):

Die vollständige Rede:

> But that was a repeat performance - Madam President! Ladies and gentlemen! Ladies and gentlemen At its core, a good law needs two things: It must not only be well thought out, but also well made. The draft before us, which we are discussing today, is both. The Act to Strengthen Supervision of legal services is first and foremost urgently needed and therefore well thought out. At present, the area of non-lawyer legal services suffers from an enormous suffers from enormous fragmentation in our country, because the supervision of this area is the responsibility of the judicial authorities of the federal states, which in turn delegate these supervisory duties to 38 different courts and public prosecutors' offices in the country. There is one authority for this in Saarland, and in Lower Saxony alone 14. This means that there is currently no uniform control and supervision whatsoever over a highly sensitive area that is growing exponentially nationwide. Precisely This creates legal uncertainties to the detriment of all involved. This applies to consumers just as much as to the creators of novel, innovative and important offerings, including in the rapidly growing market for the so-called legal tech sector. The present bill therefore rightly addresses precisely this issue and transfers supervision and control to a central supervisory authority at federal level. federal level. I and the SPD Group believe that it is important and right for this to be located at the Federal Office of Justice, given the specialist and expert knowledge available there. important and correct, ladies and gentlemen. In my view, the centralization envisaged in the bill has two advantages, among others, which I would like to emphasize: Firstly, in such a dynamically growing area, it is not desirable for a multitude of state authorities to create a patchwork of decisions that paralyze court proceedings, consume resources and make reliable nationwide enforcement of the law impossible. makes. On the other hand, in this way competencies and know-how are bundled at a central and technically experienced office. This is elementary. The The area of non-lawyer legal support is developing at a rapid pace and requires not only effective control, but also targeted management in order to prevent a decline. This requires not only effective monitoring, but also targeted control in order to effectively prevent a decline in the quality of legal services to the detriment of consumers. Good, reliable legal assistance must be accessible to everyone, but it must also not be sold off. We owe this to our constitutional state. This bill is also necessary because it finally implements the planned amendments to the provisions on fines in Sections 3 and 20 of the Legal Services Act. Legal Services Act. This will ensure that in future all unauthorized legal services, insofar as they are provided independently and on a and provided on a businesslike basis, are uniformly defined as administrative offenses and thus subject to fines. This creates a comprehensive system of sanctions under sanctions for any form of unauthorized legal services provided on a businesslike basis. Moreover, ladies and gentlemen, this bill is not only well thought out, but also well done. The draft closes existing gaps gaps, for example in dealing with lawyers who have come into contact with both sides of a legal dispute in the course of their studies or preparatory legal dispute, or in the area of the establishment of foreign lawyers in Germany. The planned new regulation effectively combats the weaknesses of the current Legal Services Act and places Germany in a progressive position with regard to the future field of legal services. legal services in a progressive and future-proof manner over the next few years. Many thanks.

Die Ergebnisse sind wie folgt zu deuten:

- **philschmidt/bart-large-cnn-samsum**[8] gibt eine lesbare Zusammenfassung aus, die immernoch sehr extrahierend ist. Es gibt formtechnisch kaum eine Veränderung zu den Ergebnissen aus Tabelle 4.2.

[8]https://huggingface.co/philschmid/bart-large-cnn-samsum

Tabelle 4.3.: Beispiel-Ergebnisse Testdurchlauf (2) auf vollständigen Reden

Modell	Zusammenfassung
philschmid/bart-large-cnn-samsum	The Act to Strengthen Supervision of legal services is first and foremost urgently needed and therefore well thought out. At present, the area of non-lawyer legal services suffers from an enormous suffers from enormous fragmentation in our country. This applies to consumers just as much as to the creators of novel, innovative and important offerings, including in the so-called legal tech sector.
transformersbook/ pegasus-samsum	The bill to strengthen the supervision of legal services is urgently needed. It transfers the supervision and control of a central supervisory authority to the federal level. The bill also implements the planned amendments to the fine provisions in sections 3 and 20 of the Legal Services Act. It creates a comprehensive criminal penalty regime for any form of commercial legal services.

- **google/pegasus-samsum**[9] zeigt starke Verbesserungen im Vergleich zu Tabelle 4.2 und bietet eine abstraktere Zusammenfassung als philschmidt/bart-large-cnn-samsum.

4.3.9. Ablauf

Mit diesen Erkenntnissen modelliere ich den Versuch mit folgenden Schritten:

1. Zusammenfassung der deutschen Rede mit TextRank (Extractive)
2. Übersetzen der Rede ins Englische
3. Berechnung der Übersetzungsmetrik aus Unterabschnitt 4.3.5
4. Generierung zwei weiterer abstrakter Zusammenfassungen mit:
   - philschmidt/bart-large-cnn-samsum (Abstractive)
   - google/pegasus-samsum (Abstractive)
5. Übersetzung der englischen Zusammenfassung ins Deutsche
6. Auswertung der Resultate

[9] https://huggingface.co/transformersbook/pegasus-samsum

5. Resultat

Ein wichtiger Punkt, den es noch zu klären gilt, ist die Frage nach dem Bewertungsschema. Es wurde der besagte Datensatz aus deutschen Bundestagsreden mit den erläuterten Methoden/Modellen zusammengefasst. Wie sind diese zu bewerten?

5.1. Bewertung

C.-Y. Lin (2004) stellte den bis heute weit verbreiteten ROUGE-Score vor. Dieser basiert jedoch auf dem Vergleich der maschinell erstellten Zusammenfassung mit einer menschlich geschriebenen Referenz. Letztere existiert für meinen Versuch nicht. Des Weiteren haben Tay u. a. (2019, S. 8) gezeigt, dass ROUGE nicht ausreichend ist, um meinungsbasierte Zusammenfassungen korrekt zu bewerten. Bundestagsreden sind ebenfalls sehr subjektiv und fallen daher in diese Kategorie. Somit habe ich für meinen Versuch eine eigene Metrik formuliert, welche die verschiedenen Zusammenfassungen einstufen soll.

5.1.1. Kriterien für eine gute Zusammenfassung

Nach Chen und Su (2011, S. 1) muss der Schreiber folgende Kriterien erfüllen, um eine gute Zusammenfassung zu generieren:

- Den Text verstehen
- Die wichtigsten Informationen selektieren
- Sekundäre Informationen sowie Duplikate entfernen
- Ähnliche Ideen zusammenführen
- In eigenen Worten schreiben

Für meine Auswertung habe ich ein Punktesystem auf Basis dieser Kriterien eingeführt. Jede Zusammenfassung startet mit zehn Punkten. Danach wird sie verschiedenen Tests unterzogen, die, je nach Ergebnis, der Zusammenfassung Punkte abziehen. Eine Zusammenfassung kann demnach maximal zehn Punkte erreichen, wenn sie alle Tests ohne Weiteres besteht (sehr gut) und minimal auf null Punkte fallen (sehr schlecht). Im Folgenden werde ich die Tests erläutern, welche die Zusammenfassungen durchlaufen.

5.1.2. Fehlerfreier Output

Bei diesem Test wird geprüft, ob die Maschine überhaupt eine Zusammenfassung generiert hat. Ist dies nicht der Fall, so werden zehn Punkte abgezogen.

Punktabzug bei Nichtbestehen: 10

5.1.3. Textlänge

Eine Zusammenfassung muss den Inhalt des Textes in wenigen Worten wiedergeben können. Eine Zusammenfassung, die länger als 30% des originalen Textes ist, führt deshalb zu Punktabzug. Je länger der Text, desto mehr Punkte werden abgezogen.

Es gilt für den Abzug:

Algorithm 1 Abzug Textlänge oben

compression_rate $\rightarrow (100.0/len(\text{full\_Text}) * len(\text{summary})$
if compression_rate ≥ 30 **then**
  loss $\rightarrow$ compression_rate$//15$
end if

Weiterhin darf die Zusammenfassung auch nicht zu kurz sein. Wurde der originale Text um 90% oder mehr gekürzt, ist es sehr unwahrscheinlich, dass alle wichtigen Punkte zusammengefasst wurden. In diesem Fall gibt es einen festen Abzug von 2 Punkten.

Es gilt für den Abzug:

Algorithm 2 Abzug Textlänge unten

compression_rate $\rightarrow (100.0/len(\text{full\_Text}) * len(\text{summary})$
if compression_rate ≤ 10 **then**
  loss $\rightarrow 2$
end if

Punktabzug bei Nichtbestehen: [2, 10]

5.1.4. Satzlänge

Sätze werden kompliziert und schwer zu lesen, wenn diese zu lang sind; vor allem in einer Zusammenfassung. Daher gibt es Punktabzug, wenn der durchschnittliche Satz mehr als zwanzig Wörter beträgt.

Es gilt für den Abzug:

Algorithm 3 Abzug Satzlänge

avg_words $\rightarrow sum(len(\text{all\_words})/len(\text{all\_sent}))$
if avg_words ≥ 20 **then**
  loss $\rightarrow$ avg_words$//10$
end if

Punktabzug bei Nichtbestehen: [2, 10]

5.1.5. Wiederholungen

Eine Zusammenfassung, vor allem eine abstrakte, soll flüssig und in eigener Sprache geschrieben sein. Wiederholen sich ähnliche Sätze, stört dies erstens den Lesefluss und zweitens ist dies ein Indiz dafür, dass Informationen wiederholt werden. Deshalb gibt es Punktabzug, wenn sich die Sätze in der Zusammenfassung untereinander zu sehr überschneiden. Dies wird mit der Levenshtein-Distanz berechnet.

Es gilt für den Abzug:

Algorithm 4 Abzug Wiederholungen

```
loss → 0
for i ← 0 to len(sentences) do
  for j ← i + 1 to len(sentences) do
    lv → levenshtein(sentences[i], sentences[j])
    similarity → 1 − lv/max(len(sentences[i]), len(sentences[j])) // In percentage
    if similarity = 1 then
      loss +5
    else if similarity ≥ 0.75 then
      loss +3
    else if similarity ≥ 0.4 then
      loss +2
    end if
  end for
end for
```

Punktabzug bei Nichtbestehen: [2,10]

5.1.6. Inhalt

Der Inhalt einer Zusammenfassung ist das oberste Gut, anhand dessen man diese bewerten kann. Wenn die Zusammenfassung nicht den Kern des Textes einfängt und wiedergibt, ist diese nutzlos und bei Falsch-Aussagen sogar gefährlich. Für diesen Versuch gibt es keine Referenzdaten. Trotzdem versuche ich anhand der Named-Entity-Relation die Zusammenfassung auf ihren Inhalt zu prüfen.

Die Idee ist Folgende:
Dem Input-Text werden die Named-Entities entzogen. Diese werden anhand ihrer Häufigkeit in einem Vektor hochgezählt und dieser durch eine Softmax-Funktion geführt. Der resultierende Vektor gibt die Named-Entity-Verteilung des Input-Texts wieder. Das Gleiche passiert mit der generierten Zusammenfassung. Wenn diese den Inhalt des Input-Texts korrekt abbildet, ähneln sich die Named-Entity-Verteilungen von Text und Zusammenfassung - so die These. Abbildung 5.1 visualisiert die Idee und zeigt für jede Zusammenfassung den Distanz-Wert d der Verteilungen.

Algorithm 5 Abzug NE-Verteilung

total_distance $\rightarrow 0$
ne_text_vector $= softmax(count(ne(\text{text})))$ // ordered
ne_sum_vector $= softmax(count(ne(\text{text})))$ // ordered
for $i \leftarrow 0$ to $len(\text{ne\_text\_vector})$ **do**
  total_distance $+sqrt(pow((\text{ne\_text\_vector}[i], \text{ne\_sum\_vector}[i]), 2))$
end for
if total_distance ≥ 3 **then**
  loss $\rightarrow 7$
else if total_distance ≥ 2 **then**
  loss $\rightarrow 5$
else if total_distance ≥ 1 **then**
  loss $\rightarrow 4$
else if total_distance ≥ 0.75 **then**
  loss $\rightarrow 2$
end if

Es gilt für den Abzug:

Punktabzug bei Nichtbestehen: [2,7]

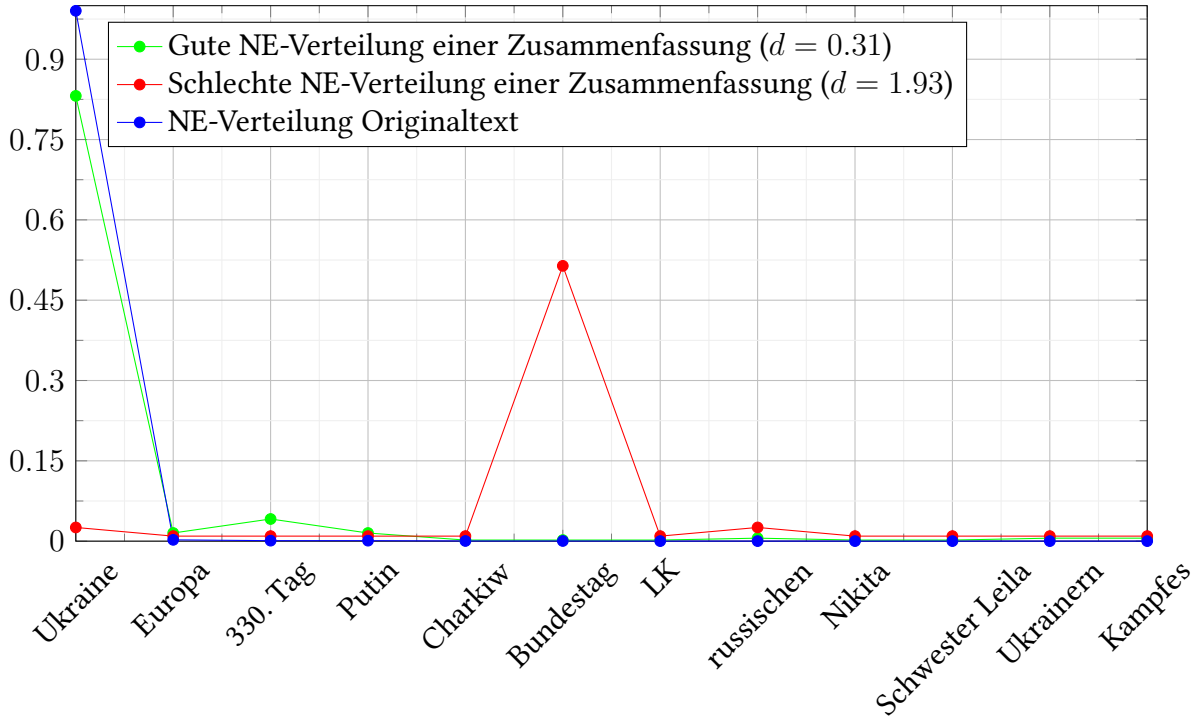
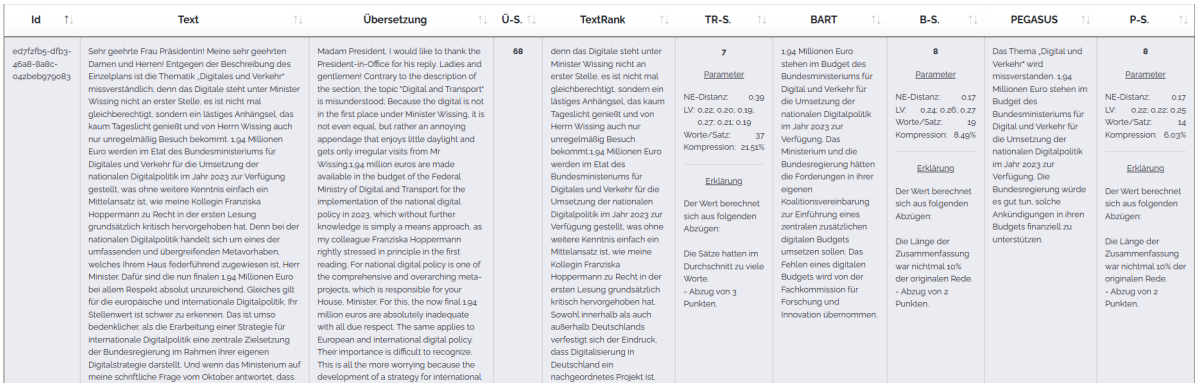


Abbildung 5.1.: NE-Verteilung zweier Zusammenfassungen im Vergleich zum Input-Text.



Id	Text	Übersetzung	Ü-S.	TextRank	TR-S.	BART	B-S.	PEGASUS	P-S.
ed7fafb5-dfb3-46a8-8a8c-042beb979083	Sehr geehrte Frau Präsidentin! Meine sehr geehrten Damen und Herren! Entgegen der Beschreibung des Einzelplans ist die Thematik „Digitales und Verkehr" missverständlich; denn das Digitale steht unter Minister Wissing nicht an erster Stelle, es ist nicht mal gleichberechtigt, sondern ein lästiges Anhängsel, das kaum Tageslicht genießt und von Herrn Wissing auch nur unregelmäßig Besuch bekommt. 1,94 Millionen Euro werden im Etat des Bundesministeriums für Digitales und Verkehr für die Umsetzung der nationalen Digitalpolitik im Jahr 2023 zur Verfügung gestellt, was ohne weitere Kenntnis einfach ein Mittelansatz ist, wie meine Kollegin Franziska Hoppermann zu Recht in der ersten Lesung grundsätzlich kritisch hervorgehoben hat. Denn bei der nationalen Digitalpolitik handelt sich um eines der umfassenden und übergreifenden Metavorhaben, welches Ihrem Haus federführend zugewiesen ist, Herr Minister. Dafür sind die nun finalen 1,94 Millionen Euro bei allem Respekt absolut unzureichend. Gleiches gilt für die europäische und internationale Digitalpolitik. Ihr Stellenwert ist schwer zu erkennen. Das ist umso bedenklicher, als die Erarbeitung einer Strategie für internationale Digitalpolitik eine zentrale Zielsetzung der Bundesregierung im Rahmen ihrer eigenen Digitalstrategie darstellt. Und wenn das Ministerium auf meine schriftliche Frage vom Oktober antwortet, dass	Madam President, I would like to thank the President-in-Office for his reply. Ladies and gentlemen! Contrary to the description of the section, the topic "Digital and Transport" is misunderstood. Because the digital is not in the first place under Minister Wissing, it is not even equal, but rather an annoying appendage that enjoys little daylight and gets only irregular visits from Mr Wissing.1.94 million euros are made available in the budget of the Federal Ministry of Digital and Transport for the implementation of the national digital policy in 2023, which without further knowledge is simply a means approach, as my colleague Franziska Hoppermann rightly stressed in principle in the first reading. For national digital policy is one of the comprehensive and overarching meta-projects, which is responsible for your House, Minister. For this, the now final 1.94 million euros are absolutely inadequate with all due respect. The same applies to European and international digital policy. Their importance is difficult to recognize. This is all the more worrying because the development of a strategy for international	68	denn das Digitale steht unter Minister Wissing nicht an erster Stelle, es ist nicht mal gleichberechtigt, sondern ein lästiges Anhängsel, das kaum Tageslicht genießt und von Herrn Wissing auch nur unregelmäßig Besuch bekommt.1,94 Millionen Euro werden im Etat des Bundesministeriums für Digitales und Verkehr für die Umsetzung der nationalen Digitalpolitik im Jahr 2023 zur Verfügung gestellt, was ohne weitere Kenntnis einfach ein Mittelansatz ist, wie meine Kollegin Franziska Hoppermann zu Recht in der ersten Lesung grundsätzlich kritisch hervorgehoben hat. Sowohl innerhalb als auch außerhalb Deutschlands verfestigt sich der Eindruck, dass Digitalisierung in Deutschland ein nachgeordnetes Projekt ist.	7 Parameter NE-Distanz: 0.39 LV: 0.22, 0.20, 0.19, 0.27, 0.21, 0.19 Worte/Satz: 37 Kompression: 21.51% Erklärung Der Wert berechnet sich aus folgenden Abzügen: Die Sätze hatten im Durchschnitt zu viele Worte. - Abzug von 3 Punkten.	1,94 Millionen Euro stehen im Budget des Bundesministeriums für Digital und Verkehr für die Umsetzung der nationalen Digitalpolitik im Jahr 2023 zur Verfügung. Das Ministerium und die Bundesregierung hätten die Forderungen in ihrer eigenen Koalitionsvereinbarung zur Einführung eines zentralen zusätzlichen digitalen Budgets umsetzen sollen. Das Fehlen eines digitalen Budgets wird von der Fachkommission für Forschung und Innovation übernommen.	8 Parameter NE-Distanz: 0.17 LV: 0.24, 0.26, 0.27 Worte/Satz: 19 Kompression: 8.49% Erklärung Der Wert berechnet sich aus folgenden Abzügen: Die Länge der Zusammenfassung war nichtmal 10% der originalen Rede. - Abzug von 2 Punkten.	Das Thema „Digital und Verkehr" wird missverstanden. 1,94 Millionen Euro stehen im Budget des Bundesministeriums für Digital und Verkehr für die Umsetzung der nationalen Digitalpolitik im Jahr 2023 zur Verfügung. Die Bundesregierung würde es gut tun, solche Ankündigungen in ihren Budgets finanziell zu unterstützen.	8 Parameter NE-Distanz: 0.17 LV: 0.22, 0.22, 0.25 Worte/Satz: 14 Kompression: 6.03% Erklärung Der Wert berechnet sich aus folgenden Abzügen: Die Länge der Zusammenfassung war nichtmal 10% der originalen Rede. - Abzug von 2 Punkten.

Abbildung 5.2.: Ergebnisse im Research-Center

5.2. Ergebnisse

Die Ergebnisse dieses Versuchs sind öffentlich einsehbar im Research-Center der Bundestags-Mine[1] (siehe Abbildung 5.2). Beispiel-Ergebnisse aus der folgenden Aufarbeitung lassen sich über deren ID dort immer finden. Dafür gilt:

- **Id**: Id der Rede in der Datenbank
- **Text**: Originaltext der Rede
- **Übersetzung**: Deutsche Rede ins Englische übersetzt
- **Ü-S.**: Übersetzungs-Score nach Unterabschnitt 4.3.5
- **TextRank**: Zusammenfassung generiert durch TextRank
- **TR-S.**: Der Score für die Zusammenfassung von TextRank
- **BART**: Zusammenfassung generiert durch BART
- **B-S.**: Der Score für die Zusammenfassung von BART
- **PEGASUS**: Zusammenfassung generiert durch PEGASUS
- **P-S.**: Der Score für die Zusammenfassung von PEGASUS

Es wurden insgesamt 11.919 Zusammenfassungen generiert.

[1] https://bundestag-mine.de/ResearchCenter/AutomaticTextSummarization

5.3. Auswertung

Tabelle 5.1.: Auswertung der erstellten Zusammenfassungen

Statistik (Alles im Durchschnitt)	TextRank	BART	PEGASUS	Gesamt
Score	6	7	5	6
Textlänge (In Worten)	93	48	52	65
Satzlänge (In Worten)	24	14	20	19
Named-Entity-Distanz	0.67	0.63	0.61	0.64
Kompressions-Rate	73%	82%	75%	77%
Satz-Überschneidungen	23, 56%	24, 09%	23, 02%	23, 56%
Fehlerhaft	0%	0%	10, 36%	3, 45%

Abschnitt 5.3 zeigt die Ergebnisse aller eingeführten Parameter.

- **Fehlerhaft**: Nur PEGASUS hat bei manchen Reden einen Fehler geworfen und diese nicht zusammengefasst. Hier lag es (nach Logdateien) an einer zu großen Token-Anzahl, weshalb der Transformer abbrach. Es wurden die gleichen Einstellungen auf den jeweiligen Modellen gewählt.
- **Satz-Überschneidungen**: Hier gibt es nur marginale Unterschiede zwischen den verschiedenen Methoden. Dies liegt wahrscheinlich daran, dass BART und PEGASUS trotzdem viel Text extrahieren, und damit auch TextRank ähneln, obwohl diese abstrakt zusammenfassen sollen.
- **Kompressions-Rate**: BART hat im Schnitt die kürzesten Zusammenfassungen generiert, gefolgt von PEGASUS und TextRank. Diese Beobachtungen stimmen mit den Vor-und Nachteilen von Extractive Text Summarization aus Abschnitt 2.4 überein, jedoch war meine Annahme, dass sich TextRank deutlich mehr von PEGASUS und BART absetzt, als es die Zahlen zeigen.
- **Named-Entity-Distanz**: Hier gibt es auch kaum Unterschiede. Meine Vermutung ist auch hier, dass PEGASUS und BART extrahierender gearbeitet haben, als man es annehmen durfte. Dadurch ähneln sich die Werte. Trotzdem ist überraschend, dass TextRank die größte Named-Entity-Distanz hat und damit, nach meinem Kriterium, den wahrscheinlich schlechtesten Informationsgehalt generiert hat.
- **Satzlänge**: Die Ergebnisse stimmen mit den Annahmen aus Abschnitt 2.4 überein.
- **Textlänge**: Hier ist der Unterschied zwischen Extractive/Abstractive am besten zu erkennen. TextRank generiert im Schnitt fast doppelt so viele Wörter pro Zusammenfassung wie BART oder PEGASUS.

- **Score**: Laut dem eingeführten Bewertungsschema hat BART die besten Zusammenfassungen generiert, gefolgt von TextRank und abschließend PEGASUS. Mit einem Schnitt von $6/10$ Punkten sind die Zusammenfassungen nur mittelmäßig gut ausgefallen.

5.3.1. Übersetzung

Wie sehr beeinträchtigt das Übersetzen der deutschen Reden die Zusammenfassung? Nach dem eingeführten Übersetzungs-Score aus Unterabschnitt 4.3.5 lagen alle Übersetzungen im Schnitt bei einer Überlagerung von 89% und damit bei einem sehr guten Wert. Die Zusammenfassung hat im Allgemeinen also nicht unter der vorherigen Übersetzung gelitten. Ausnahmen sind vereinzelte Eigennamen, die wörtlich übersetzt wurden. So wurden zum Beispiel die "Freien Demokraten" nach "Free Democrats" übersetzt, nach der Zusammenfassung jedoch nicht wieder zurück ins Deutsche überführt. Die resultierende Zusammenfassung[2] (von BART) bestand dann unter anderem aus dem Satz:

> Die Free Democrats wollen den gesamten Prozess durch die Genehmigung von Beimischungen für die Anwendung öffnen.

Das Übersetzen hat in der Regel keine Auswirkungen auf die Qualität der Zusammenfassungen, jedoch findet sich das Problem oft in den Eigennamen.

5.3.2. Bewertungsschema

Wie gut funktioniert mein, für diesen Versuch eingeführtes, Bewertungsschema? Dies wird anhand von Stichproben versucht zu erörtern.

Positiv-Beispiele

Tabelle 5.2 zeigt drei Zusammenfassungen zur gleichen Rede. Auch ohne die originale Rede gelesen zu haben, lässt sich der Score gut einordnen.

TextRank hat folgende Abzüge bekommen:

- -3: Die Zusammenfassung war zu kurz (Kompression von über 90%)
- -4: Die Named-Entity-Dichte lag bei $1,29$
- -2: Die Sätze waren sich zu ähnlich

TextRank hat sich hier sehr auf den Ausdruck "Liebe Kolleginnen und Kollegen" fokussiert; vermutlich aufgrund der hohen Frequenz. Leider ist dies nur eine Ansprache und in einer Zusammenfassung unbrauchbar (Named-Entity-Dichte war schlecht). Der Kern der Rede wird durch Wiederholung von sehr ähnlichen Sätzen ersetzt. Alle Punktabzüge sehe ich hier als adäquat und berechtigt an. Die Zusammenfassung ist nicht gut.

[2] Rede: f1f524a3-27c5-4cfc-e7a1-08daea4d024d

BART hat keinen Abzug bekommen. Die Zusammenfassung lässt sich flüssig lesen, die Sätze sind nicht zu lang, es gibt keine Wiederholung von Information und der Kern der Rede ist ersichtlich. Auch hier sehe ich die Bewertung als korrekt an.

PEGASUS hat folgende Abzüge bekommen:

- -3: Die Zusammenfassung war zu kurz (Kompression von über 90%)
- -2: Die Named-Entity-Dichte lag bei 0.87

PEGASUS hat eine ähnliche Zusammenfassung wie BART generiert. Der Abzug bzgl. der zu kurzen Zusammenfassung war unglücklich, da diese um nur $0,58\%$ zu kurz war. BART hingegen ist mit einer Länge von $10,25\%$ unter dem Radar geblieben. Was diesen Punkt angeht, sollten optimalerweise beide Zusammenfassungen gleich behandelt werden. Den Abzug durch die mangelnde Named-Entity-Dichte sehe ich jedoch als korrekt an. Vergleicht man die Zusammenfassungen von PEGASUS und BART genauer, so wird ersichtlich, dass BART zwei Informationen mehr generiert als PEGASUS:
- *"Irans Parlament hat Patenschaften für einige der über 18 000 politischen Gefangenen Irans übernommen."*
- *"Mehrere EU-Sanktionspakete wurden bereits verabschiedet."*
PEGASUS hingegen nahm noch die Information auf, dass *"[d]er Krieg der Mullah gegen Gott [...] der liberal-demokratischen Bewegung vorgeworfen [wird]"*, jedoch wirkt dieser Satz sehr fehl am Platz und aus dem Kontext gerissen, auch wenn er generell nicht unnötig ist. Die Zusammenfassungen liegen nahe beieinander, das zeigen auch die Werte, jedoch sehe ich das Ergebnis von BART als akkurater an - vor allem mit Blick auf die ganze Rede.

Negativ-Beispiele

Tabelle 5.3 zeigt drei weitere Zusammenfassungen samt Score. Alle drei Zusammenfassungen haben gut bis sehr gut abgeschnitten, wobei TextRank die schlechteste generiert haben soll. Dies lag vor allem an den langen Sätzen. BART und PEGASUS haben, nach Bewertungsschema, sehr gute Zusammenfassungen generiert. Bei genauer Betrachtung fällt jedoch auf, dass beide Zusammenfassungen davon ausgehen, dass Frau Baerbock die Rede gehalten hat. Die Rede wurde jedoch von Jürgen Braun gehalten. In seinem zweiten Satz referenziert er Frau Baerbock sarkastisch:

> Tja, das ist das Ergebnis dieser Rede der Bundesaußenministerin Annalena Baerbock: [...]

Dadurch gehen BART und PEGASUS irrtümlicherweise davon aus, dass Frau Baerbock spricht. Hier sollten optimalerweise BART und PEGASUS null Punkte anerkannt bekommen, während TextRank am besten abschneiden sollte.

Tabelle 5.2.: Zusammenfassungen Rede (siehe Anhang A)

Methode	Zusammenfassung	Score
TextRank	Das hier ist ein Haus der freien Rede, liebe Kolleginnen und Kollegen! Und wichtig ist, dass wir mit dem Druck nicht nachlassen, liebe Kolleginnen und Kollegen. Am 3. November sind sie nach der Trauerfeier für eine junge, durch das Regime ermordete Frau aus Sicht des Mullah-Regimes zur falschen Zeit am falschen Ort. Kolleginnen und Kollegen!	2
BART	Farzaneh und ihr Mann Hamid sind politische Gefangene im Iran. Sie wurden von Sicherheitskräften zu Hause angegriffen, schwer missbraucht und vor ihren Kindern entführt. Sie werden gefoltert und erpresst, um falsche Geständnisse zu machen. Hamid wird ohne Beweise zum Tode verurteilt, Farzana zu 25 Jahren Gefängnis ohne Besuchsrechte. Irans Parlament hat Patenschaften für einige der über 18 000 politischen Gefangenen Irans übernommen. Mehrere EU-Sanktionspakete wurden bereits verabschiedet.	10
PEGASUS	Farzaneh und ihr Mann Hamid sind politische Gefangene im Iran, für die ich die Patenschaft übernommen habe. Hamid wird ohne Beweise und unter Missachtung der Mindeststandards der Rechtsstaatlichkeit zum Tode verurteilt. Farzaneh wird zu 25 Jahren Gefängnis ohne Besuchsrechte verurteilt. Der Krieg der Mullah gegen Gott wird der liberal-demokratischen Bewegung vorgeworfen. Das Volk Irans rebelliert nicht gegen Gott. Sie rebellieren gegen dieses Regime.	6

Tabelle 5.3.: Zusammenfassungen Rede (siehe Abbildung A)

Methode	Zusammenfassung	Score
TextRank	Ein solcher Staat darf hier Handlangervereine gründen, die auch noch als gemeinnützig anerkannt werden und deutsche Steuergelder aus Staatsverträgen bekommen, wie im Fall der Blauen Moschee im rot-grünen Hamburg. Nicht die Iraner leiden unter dem Mullah-Regime, sondern die Bundesaußenministerin leidet im Auswärtigen Amt und entblättert hier den entsetzlichen Zustand in Sachen Außenpolitik. Jene Parteien, die sich niemals kritisch zum Kopftuchzwang in Deutschland geäußert haben, inszenieren sich jetzt als Unterstützer der Proteste im Iran – ausgerechnet jener Proteste, die sich am Kopftuchzwang entzündet haben;am Kopftuchzwang, das ihnen ein Regime aufgezwungen hat, das von Anfang an von den europäischen Linken hofiert und gestützt worden ist. Das führt dazu, dass der Iran nicht nur im Verhältnis zur Bevölkerungszahl, sondern sogar in absoluten Zahlen das Land mit den meisten Geschlechtsumwandlungen weltweit ist.	7
BART	Annalena Baerbock, die deutsche Außenministerin, hat eine Rede gehalten, in der sie die iranische Regierung und ihre Außenpolitik kritisiert. Annalena wirft den linken Grünen Steinmeier, Joschka Fischer, Claudia Roth und Co. vor, die Proteste im Iran zu unterstützen.	8
PEGASUS	Annalena Baerbock wirft dem Auswärtigen Amt einen entsetzlichen Zustand der Außenpolitik vor. Die linken Grünen versuchen seit Jahrzehnten zu überzeugen, dass die sogenannten Reformer unsere Partner sind und humaner sind als Ahmadinejad. Die lokale Bevölkerung hat schon lange genug vom Mullah-Regime. Die Islamische Republik ist ein Staat, in dem Homosexuellen die Wahl gegeben wird, entweder Hurerei oder Sterilisation oder Re-Operation in Frauen, alles im Namen des Islams, hingerichtet zu werden.	10

6. Zusammenfassung und Ausblick

6.1. Zusammenfassung

Automatic Text Summarization ist ein wachsendes Feld, welchem vor allem durch die neue Transformer-Technologie neue Möglichkeiten eröffnet wurden (so wie der Künstlichen Intelligenz im Allgemeinen). Die Anwendungsbereiche sind groß und die Wertschöpfung im Besten Fall enorm. Während die Sprachmodelle im englisch-sprachigen Raum bereits sehr fundierte Ergebnisse liefern, ist die Anwendung im deutschen Raum vergleichsweise noch unterentwickelt. Es gibt zu wenige qualitativ hochwertige deutsche Datensätze, welche die Sprachmodelle verfeinern könnten. Nur über Umwege und Abstriche konnten im Kontext dieses Versuchs deutsche Zusammenfassungen überhaupt generiert werden. Des Weiteren hat der Versuch gezeigt, dass ein Modell zwar auf einer Domain funktionieren kann (News-Artikeln), gleichzeitig aber auf einer anderen (Bundestagsreden) starke Defizite aufweist. Das eingeführte Bewertungsschema hat gezeigt, dass die generierten Zusammenfassungen im Schnitt nur mittelmäßig waren. Dabei wurde vor allem auf die Form geschaut. Katastrophale Fehler, wie die Falschinterpretation von Aussagen durch abstrakte Zusammenfassungen (siehe Unterunterabschnitt 5.3.2), sind im Bewertungsschema nicht aufgefallen, was die Glaubwürdigkeit dieses anzweifeln lässt. Es zeigt jedoch, dass extrahierende Methoden, wie der TextRank, zuverlässiger sind, da diese nichts falsch interpretieren und dadurch weitergeben können. Dafür leidet hier, neben anderen Dingen aus Abschnitt 2.4, die Form und allgemeine Ästhetik.

6.2. Ausblick

Eine Zusammenfassung ohne Referenz zu bewerten, ist ein kompliziertes Unterfangen. Das Bewertungsschema aus Abschnitt 5.1 hat Probleme mit der Überprüfung auf allgemeine ”Richtigkeit“, besitzt mit der Named-Entity-Distanz jedoch einen Ansatz dafür. Dieser könnte durch einen weiteren Sentiment-Distanz-Test flankiert werden. Hier würde der Sentiment des Input-Texts mit dem Sentiment der Zusammenfassung verglichen werden. Lägen diese beiden weit auseinander, könnte das ein Indiz dafür sein, dass die emotional wichtigsten Aussagen des Redenden nicht beachtet wurden. Hier müsste vor allem auf die Extrema geschaut werden und nicht einfach der simple Durchschnitt berechnet werden. Außerdem ließen sich die Parameter in den vorgestellten Tests noch verbessern.

Weiterhin bedarf es mehr systematischer Datenakquirierung im deutschen Raum. Eine Lösung dafür könnte es sein, eine Plattform zu bieten, auf der menschliche Referenz-Zusammenfassungen geschrieben und eingereicht werden können, um einen beschrifteten Datensatz, bestehend aus Reden samt derer Zusammenfassungen, zu generieren und für die Öffentlichkeit bereitzustellen.

A. Anhang

Frau Präsidentin! Kolleginnen und Kollegen! Liebe Zuschauende! Ich möchte Ihnen von Farzaneh und ihrem Ehemann Hamid erzählen, zwei politische Gefangene im Iran, für die ich die Patenschaft übernommen habe. Es handelt sich um ein Ehepaar, das für seinen Einsatz und seine Unterstützung in Krisengebieten, bei der medizinischen Versorgung und Unterstützung von Kindern bekannt ist. Am 3. November sind sie nach der Trauerfeier für eine junge, durch das Regime ermordete Frau aus Sicht des Mullah-Regimes zur falschen Zeit am falschen Ort. In der Nacht wird das Ehepaar von Sicherheitskräften zu Hause überfallen, vor den Augen ihrer Kinder schwer misshandelt und verschleppt. Im Gefängnis werden sie gefoltert und erpresst, falsche Geständnisse abzulegen. Hamid erleidet innere Blutungen und Rippenfrakturen. Das Scheingerichtsverfahren gegen ihn und seine Frau wird ohne rechtlichen Beistand ihrer Wahl durchgeführt. Farzaneh wird eine Aussage verweigert. Hamid wird ohne Beweise und unter Missachtung rechtsstaatlicher Mindeststandards zu Tode verurteilt, seine Frau Farzaneh zu 25 Jahren Haft ohne Besuchsrecht. Das Ziel ist, den Anschein von Rechtsstaatlichkeit zu wahren und die Menschen zum Schweigen zu bringen. Aber sie können nicht die ganze Welt zum Schweigen bringen. Das hier ist ein Haus der freien Rede, liebe Kolleginnen und Kollegen! Der freiheitlich-demokratischen Bewegung werfen die Mullahs Krieg gegen Gott vor. Doch die Menschen im Iran rebellieren nicht gegen Gott. Sie rebellieren gegen dieses Regime. Dieses Regime verwechselt sich selbst mit Gott. Sie brechen die elementarsten Regeln menschlicher Zivilisation. Fritz Bauer bezeichnete Zivilisation als eine sehr dünne Decke, die schnell abblättert. Mit Schauprozessen, Mord an Kindern und Massenhinrichtungen beweist das Regime, wie recht er hatte. Deswegen ist es auch international isoliert, und das ist auch gut so. Die Opfer der Revolution haben Gesichter. Sie haben Geschichten und Botschaften. Diese lassen sich in einer freien Welt nicht auslöschen. Viele in diesem Parlament haben Patenschaften für einzelne der über 18 000 politischen Gefangenen im Iran übernommen. Lassen Sie die Welt hören, wer die Menschen sind, die von den Mullahs in Folterkammern gesteckt werden. Es ist wichtig, dass wir laut sind für sie. Denn die Menschen im Iran können uns hören, und sie sollen wissen: Es ist ihre Revolution. Aber im Kampf für Freiheit und Selbstbestimmung sind sie nicht allein. Wir stehen an ihrer Seite. Mehrere EU-Sanktionspakete sind bereits verabschiedet. Diese Woche ist ein weiteres hinzugekommen. Unser Parlament hat dem Regime vor Wochen jegliche Legitimation abgesprochen. Im UN-Menschenrechtsrat ist auch dank der Außenministerin eine Mehrheit gegen das iranische Regime zustande gekommen, die die Gewalt im Land unabhängig untersuchen will. Aus der UN-Kommission für Frauenrechte wurde das Land nun ausgeschlossen. Es zeigt: Der internationale Druck hilft. Und wichtig ist, dass wir mit dem Druck nicht nachlassen, liebe Kolleginnen und Kollegen. Mit einer Regierung, die Mordanschläge in Deutschland auch schon hat durchführen lassen und offenbar weitere plante, werden wir nicht auf Augenhöhe zusammenarbeiten. Die Spirale der Gewalt braucht eine unmissverständliche Antwort: maximalen Druck gegen dieses Regime. Auch deswegen ist es wichtig, dass eine Terrororganisation wie die Revolutionswächter auf eine Terrorliste kommt. Auch deswegen ist wichtig, dass sämtliche Abgeordnete, die ihre Hand für die Massenhinrichtungen gehoben haben, ebenfalls persönlich sanktioniert werden, liebe Kolleginnen und Kollegen. Zugleich braucht es Unterstützung für die Zivilbevölkerung, die seit Jahren unter den internationalen Sanktionen am stärksten leidet. Keiner von uns weiß, wie lange der Kampf bis zur Befreiung des Irans dauern wird. Und weil das so ist, dürfen die Menschen im Land nicht auf sich selbst gestellt sein. Freie Kommunikation, Austausch mit Kulturschaffenden, Presse und Wissenschaft, Versorgung mit Hilfsgütern – das ist, wozu wir einen konkreten Beitrag leisten können und auch leisten müssen. Bei alldem wird auch entscheidend sein, dass dieses Parlament in den zentralen Punkten mit einer Stimme spricht. Im Iran fragt aktuell keiner, ob du Perser, Kurde, Afghane, Araber, Belutsche oder Aserbaidschaner bist. Da wird nur gefragt: Bist du für einen freien Iran? – Das ist eine Frage, die auch unser Parlament einen sollte, liebe Kolleginnen und Kollegen. Die Tage der Mullahs sind gezählt. Für ihre Taten werden sie sich verantworten. Aber das kann ein langer Weg werden. Menschen, denen der Tod droht, gehen auf die Straße, weil sie an den Sieg von Freiheit, Gerechtigkeit und Selbstbestimmung glauben. Und wenn sie das tun, dann sollten wir das auch. Setzen wir auf den Erfolg der Revolution! Ich danke Ihnen.

Abbildung A.1.: Rede *b19d3875-15e1-4a8c-a7ea-04fad143c136*

Herr Präsident! Meine sehr verehrten Damen und Herren! Tja, das ist das Ergebnis dieser Rede der Bundesaußenministerin Annalena Baerbock: Nicht die Iraner leiden unter dem Mullah-Regime, sondern die Bundesaußenministerin leidet im Auswärtigen Amt und entblättert hier den entsetzlichen Zustand in Sachen Außenpolitik. Es ist wirklich beschämend! Es ist beschämend, was Sie hier abgeliefert haben in den letzten Minuten. So eine Rede – unglaublich! Jene Parteien, die sich niemals kritisch zum Kopftuchzwang in Deutschland geäußert haben, inszenieren sich jetzt als Unterstützer der Proteste im Iran – ausgerechnet jener Proteste, die sich am Kopftuchzwang entzündet haben; am Kopftuchzwang, das ihnen ein Regime aufgezwungen hat, das von Anfang an von den europäischen Linken hofiert und gestützt worden ist. Die Links-Grünen Steinmeier, Joschka Fischer, Claudia Roth und Co haben uns jahrzehntelang weismachen wollen, die sogenannten Reformer seien ja unsere Partner und ach so viel humaner als Typen wie Ahmadinejad. Glückwunschtelegramme zum Jahrestag der Islamischen Revolution, unwürdige Atomabkommen und sogar devote Delegationsreisen mit Kopftuch waren die Folge. Am Sonntag hat das iranische Scheinparlament mit fast 80-prozentiger Mehrheit, also auch mit Unterstützung der angeblichen Reformer, beschlossen, dass die Teilnahme an den Demonstrationen fortan als „Krieg gegen Allah“ gelten soll – ein Tatbestand, auf den im Iran die Todesstrafe steht. Die Leben von rund 14 000 inhaftierten Demonstranten befinden sich in Gefahr, allein weil sie von ihrem Recht auf Versammlungsfreiheit Gebrauch gemacht haben. Seit dem gewaltsamen Tod der 22-jährigen Mahsa Amini sind den islamistischen Schlägertrupps der Mullahs unzählige weitere junge Frauen und Mädchen zum Opfer gefallen. Insgesamt geht es mittlerweile um mehr als 300 Demonstranten, Männer wie Frauen, die vom Regime innerhalb von zwei Monaten ermordet wurden. Aber anders, als wohl insgeheim im Auswärtigen Amt erhofft, reißen die Proteste keineswegs ab. Nein, die Proteste gehen weiter, und zwar keineswegs nur in Teheran, sondern ebenso in der Provinz. Auch die dortige Bevölkerung hat seit Langem genug vom Mullah-Regime. Die Islamische Republik ist ein Staat, in dem Schwule vor die Wahl gestellt werden, entweder wegen Unzucht hingerichtet zu werden oder sich sterilisieren und zu Frauen umoperieren zu lassen, und das alles im Namen des Islam. Denn es gibt eine Fatwa des Ajatollah Chomeini, die das vorschreibt. Das führt dazu, dass der Iran nicht nur im Verhältnis zur Bevölkerungszahl, sondern sogar in absoluten Zahlen das Land mit den meisten Geschlechtsumwandlungen weltweit ist. Vielleicht haben unsere Gender-Gaga-Fanatiker deshalb immer so viel für die Mullahs übriggehabt. Ein solcher Staat darf hierzulande Interessensvertretungen haben. Ein solcher Staat darf hier Handlangervereine gründen, die auch noch als gemeinnützig anerkannt werden und deutsche Steuergelder aus Staatsverträgen bekommen, wie im Fall der Blauen Moschee im rot-grünen Hamburg. Jahrelanges deutsches Zaudern, jahrelange Halbheiten haben nur der Islamischen Revolutionsgarde genutzt. Sie unterhält einen Schlägertrupp al-Basidsch, der an der Niederschlagung der Demonstrationen beteiligt ist. Aber die Revolutionsgarde ist nicht nur kriminell-islamistisch, sondern auch kriminell-kleptokratisch. Sie hat sich etliche Fabriken und Raffinerien unter den Nagel gerissen. Die Drohnenproduktion und nicht zuletzt das iranische Streben nach der Atombombe erfolgen unter Aufsicht von Revolutionsgardisten. Wenn Sie wirklich Druck auf das Regime in Teheran ausüben wollen, dann beenden Sie Ihre heuchlerische Farce, und benennen Sie die Vorgänge als das, was sie sind: brutaler islamistischer Terror.

Abbildung A.2.: Rede *b9a5ff93-8adf-4a6b-9aba-033a69382b8d*

Abbildungsverzeichnis

Literatur

Abrami, Giuseppe, Mevlüt Bagci, Leon Hammerla und Alexander Mehler (Juni 2022). „German Parliamentary Corpus (GerParCor)“. In: *Proceedings of the Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, S. 1900–1906. URL: https://aclanthology.org/2022.lrec-1.202.

Alammar, Jay (2018). *The Illustrated Transformer*. URL: http://jalammar.github.io/illustrated-transformer/ (besucht am 22. 02. 2023).

Allahyari, Mehdi u. a. (2017). „Text summarization techniques: a brief survey“. In: *arXiv preprint arXiv:1707.02268*.

Altinok, D. (2021). „Mastering spaCy: An end-to-end practical guide to implementing NLP applications using the Python ecosystem.“ In: *Packt Publishing Ltd.*

Aumiller, Dennis, Jing Fan und Michael Gertz (2023). *On the State of German (Abstractive) Text Summarization*. DOI: 10.48550/ARXIV.2301.07095. URL: https://arxiv.org/abs/2301.07095.

Ba, Jimmy Lei, Jamie Ryan Kiros und Geoffrey E. Hinton (2016). *Layer Normalization*. DOI: 10.48550/ARXIV.1607.06450. URL: https://arxiv.org/abs/1607.06450.

Bloem, Peter (2020). *Lecture 12.1 Self-attention*. Youtube. URL: https://www.youtube.com/watch?v=KmAISyVvE1Y.

C, Bala Priya (2022). *Build Better Deep Learning Models with Batch and Layer Normalization*. URL: https://www.pinecone.io/learn/batch-layer-normalization/ (besucht am 24. 02. 2023).

Carenini, Giuseppe, Jackie Chi und Kit Cheung (Juni 2008). „Extractive vs. NLG-based Abstractive Summarization of Evaluative Text: The Effect of Corpus Controversiality“. In.

Chen, Yuan-Shan und Shao-Wen Su (Sep. 2011). „A genre-based approach to teaching EFL summary writing“. In: *ELT Journal* 66.2, S. 184–192. ISSN: 0951-0893. DOI: 10.1093/elt/ccr061. eprint: https://academic.oup.com/eltj/article-pdf/66/2/184/1037176/ccr061.pdf. URL: https://doi.org/10.1093/elt/ccr061.

Chitrakala, S., N. Moratanch, B. Ramya, C. G. Revanth Raaj und B. Divya (2018). „Concept-Based Extractive Text Summarization Using Graph Modelling and Weighted Iterative Ranking“. In: *Emerging Research in Computing, Information, Communication and Applications*. Hrsg. von N. R. Shetty, L. M. Patnaik, N. H. Prasad und N. Nalini. Singapore: Springer Singapore, S. 149–160. ISBN: 978-981-10-4741-1.

Feng, Fangxiaoyu, Yinfei Yang, Daniel Cer, Naveen Arivazhagan und Wei Wang (2020). *Language-agnostic BERT Sentence Embedding*. DOI: 10.48550/ARXIV.2007.01852. URL: https://arxiv.org/abs/2007.01852.

Gliwa, Bogdan, Iwona Mochol, Maciej Biesek und Aleksander Wawer (Nov. 2019). „SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization“. In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Hong Kong, China:

Association for Computational Linguistics, S. 70–79. DOI: 10.18653/v1/D19-5409. URL: https://www.aclweb.org/anthology/D19-5409.

Gupta, Vishal und Gurpreet Lehal (Aug. 2010a). „A Survey of Text Summarization Extractive Techniques“. In: *Journal of Emerging Technologies in Web Intelligence* 2. DOI: 10.4304/jetwi.2.3.258-268.

– (Aug. 2010b). „A Survey of Text Summarization Extractive Techniques“. In: *Journal of Emerging Technologies in Web Intelligence* 2. DOI: 10.4304/jetwi.2.3.258-268.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren und Jian Sun (2015). *Deep Residual Learning for Image Recognition*. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

Hou, Liwei, Po Hu und Chao Bei (2018). „Abstractive Document Summarization via Neural Model with Joint Attention“. In: *Natural Language Processing and Chinese Computing*. Hrsg. von Xuanjing Huang, Jing Jiang, Dongyan Zhao, Yansong Feng und Yu Hong. Cham: Springer International Publishing, S. 329–338. ISBN: 978-3-319-73618-1.

Joshi, Monika, Hui Wang und Sally McClean (2018). „Dense Semantic Graph and Its Application in Single Document Summarisation“. In: *Emerging Ideas on Information Filtering and Retrieval: DART 2013: Revised and Invited Papers*. Hrsg. von Cristian Lai, Alessandro Giuliani und Giovanni Semeraro. Cham: Springer International Publishing, S. 55–67. ISBN: 978-3-319-68392-8. DOI: 10.1007/978-3-319-68392-8_4. URL: https://doi.org/10.1007/978-3-319-68392-8_4.

Junnan, Zhu u. a. (Jan. 2018). „Augmenting Neural Sentence Summarization Through Extractive Summarization“. In: S. 16–28. ISBN: 978-3-319-73617-4. DOI: 10.1007/978-3-319-73618-1_2.

El-Kassas, Wafaa S, Cherif R Salama, Ahmed A Rafea und Hoda K Mohamed (2021). „Automatic text summarization: A comprehensive survey“. In: *Expert Systems with Applications* 165, S. 113679.

Lewis, Mike u. a. (2019). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. DOI: 10.48550/ARXIV.1910.13461. URL: https://arxiv.org/abs/1910.13461.

Lin, Chin-Yew (Juli 2004). „ROUGE: A Package for Automatic Evaluation of Summaries“. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, S. 74–81. URL: https://aclanthology.org/W04-1013.

Lin, Jimmy (2009). „Summarization“. In: *Encyclopedia of Database Systems*. Hrsg. von LING LIU und M. TAMER ÖZSU. Boston, MA: Springer US, S. 2884–2889. ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_953. URL: https://doi.org/10.1007/978-0-387-39940-9_953.

Lipton, Zachary C, John Berkowitz und Charles Elkan (2015). „A critical review of recurrent neural networks for sequence learning“. In: *arXiv preprint arXiv:1506.00019*.

Luhn, H. P. (1958). „The Automatic Creation of Literature Abstracts“. In: *IBM Journal of Research and Development* 2.2, S. 159–165. DOI: 10.1147/rd.22.0159.

Mallick, Chirantana, Ajit Kumar Das, Madhurima Dutta, Asit Kumar Das und Apurba Sarkar (2019). „Graph-Based Text Summarization Using Modified TextRank“. In: *Soft Computing in Data Analytics*. Hrsg. von Janmenjoy Nayak, Ajith Abraham, B. Murali Krishna, G. T.

Chandra Sekhar und Asit Kumar Das. Singapore: Springer Singapore, S. 137–146. ISBN: 978-981-13-0514-6.

Mihalcea, Rada und Paul Tarau (2004). „Textrank: Bringing order into text“. In: *Proceedings of the 2004 conference on empirical methods in natural language processing*, S. 404–411.

Moratanch, N. und Chitrakala Gopalan (Jan. 2017). „A survey on extractive text summarization“. In: S. 1–6. DOI: 10.1109/ICCCSP.2017.7944061.

Nenkova, Ani und Kathleen McKeown (2012). „A Survey of Text Summarization Techniques“. In: *Mining Text Data*. Hrsg. von Charu C. Aggarwal und ChengXiang Zhai. Boston, MA: Springer US, S. 43–76. ISBN: 978-1-4614-3223-4. DOI: 10.1007/978-1-4614-3223-4_3. URL: https://doi.org/10.1007/978-1-4614-3223-4_3.

Page, Lawrence, Sergey Brin, Rajeev Motwani und Terry Winograd (Nov. 1999). *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. Previous number = SIDL-WP-1999-0120. Stanford InfoLab. URL: http://ilpubs.stanford.edu:8090/422/.

Tahseen, Rabia, Uzma Omer, Muhammad Shoaib Farooq und Faiqa Adnan (2022). „Text Summarization Techniques Using Natural Language Processing: A Systematic Literature Review“. In.

Tandel, Amol, Brijesh Modi, Priyasha Gupta, Shreya Wagle und Sujata Khedkar (2016). „Multi-document text summarization - a survey“. In: *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, S. 331–334. DOI: 10.1109/SAPIENCE.2016.7684115.

Tay, Wenyi, Aditya Joshi, Xiuzhen Zhang, Sarvnaz Karimi und Stephen Wan (Apr. 2019). „Red-faced ROUGE: Examining the Suitability of ROUGE for Opinion Summary Evaluation“. In: *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*. Sydney, Australia: Australasian Language Technology Association, S. 52–60. URL: https://aclanthology.org/U19-1008.

Tiedemann, Jörg und Santhosh Thottingal (Nov. 2020). „OPUS-MT – Building open translation services for the World“. In: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. Lisboa, Portugal: European Association for Machine Translation, S. 479–480. URL: https://aclanthology.org/2020.eamt-1.61.

Vaswani, Ashish u. a. (2017). „Attention is all you need“. In: *Advances in neural information processing systems* 30.

Wan, Xiaojun, Huiying Li und Jianguo Xiao (Juli 2010). „Cross-Language Document Summarization Based on Machine Translation Quality Prediction“. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, S. 917–926. URL: https://aclanthology.org/P10-1094.

Zhang, Jingqing, Yao Zhao, Mohammad Saleh und Peter J. Liu (2019). *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*. DOI: 10.48550/ARXIV.1912.08777. URL: https://arxiv.org/abs/1912.08777.

Zhu, Junnan, Yu Zhou, Jiajun Zhang und Chengqing Zong (Juli 2020). „Attend, Translate and Summarize: An Efficient Method for Neural Cross-Lingual Summarization“. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, S. 1309–1321. DOI: 10.18653/v1/2020.acl-main.121. URL: https://aclanthology.org/2020.acl-main.121.