FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Textzusammenfassung von QM-Dokumenten mithilfe von BERT

Lucas Dohmen, 3283474

Erstbetreuer: Prof. Dr. Terstegge

Zweitbetreuer: Elena Schweicher, M. Sc.

16. Januar 2023

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die Seminararbeit mit dem Thema

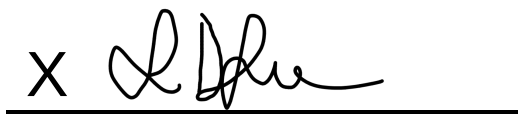
Textzusammenfassung von QM Dokumenten mithilfe von BERT

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Ich verpflichte mich, ein Exemplar der Seminararbeit fünf Jahre aufzubewahren und auf Verlangen dem Prüfungsamt des Fachbereiches Medizintechnik und Technomathematik auszuhändigen.

Name: Lucas Dohmen

Aachen, den 16. Januar 2023



Lucas Dohmen

Unterschrift des Studierenden

Zusammenfassung

Eine kurze prägnante Zusammenfassung eine langen Dokuments, kann einen Nutzer schnell über das Thema informieren und hilft beim gezielten Suchen von Informationen. Seit den 1950er Jahren wird versucht einer Maschine das Verständnis von natürlicher Sprache beizubringen und die Errungenschaften der letzten Jahre haben uns deutlich näher an dieses Ziel gebracht. Modelle wie BERT und GPT werden bereits von vielen im Alltag genutzt und mit der öffentlichen Testphase von ChatGPT wurde ein neuer Meilenstein gesetzt. Diese Seminararbeit befasst sich mit den Methoden und dem heutigen Stand der Technik zum Thema Computerlinguistik mit Schwerpunkt Textzusammenfassung.

Inhaltsverzeichnis

1 Einleitung

Der Einsatz von künstlicher Intelligenz (KI) zur Textzusammenfassung kann für Unternehmen eine Vielzahl von Vorteilen bieten. Eine KI-basierte Textzusammenfassung ermöglicht es, große Mengen an Text schnell und effizient zu analysieren und wichtige Informationen hervorzuheben. Dies kann besonders nützlich sein, wenn Unternehmen viel Zeit damit verbringen, Berichte, Memos oder andere Dokumente zu lesen und zu analysieren. Die neuesten Modelle, wie Bert und GPT-3 sind in der Lage wichtige Informationen aus Texten zu extrahieren und diese in wenigen Sätzen wiederzugeben.

Diese Seminararbeit befasst sich mit der Recherche zum Thema Computerlinguistik im Themenbereich Textzusammenfassung, den aktuellen Stand der Technik und die Nutzung von Modellen wie BERT für die ConSense GmbH. Des Weiteren wird ein lokal laufendes Programm zur Textzusammenfassung implementiert, um Dokumente zusammenzufassen.

2 Aufgabe

Die Produkte der ConSense GmbH verwalten eine Vielzahl von Qualitätsmanagement-Dokumente nach ISO 9001. Diese Dokumente haben oft viel Inhalt und eine kurze Zusammenfassung des Textes soll zu der Benutzerfreundlichkeit des Produkts beitragen. Es sollen verschiedene Anbieter und deren Modelle verglichen werden, mit Beachtung des Umgangs mit den Daten.

3 Theoretische Grundlagen

3.1 Die Geschichte der Computerlinguistik (NLP)

Computerlinguistik (engl. *Natural Language Processing (NLP)*) entstand in den 1950er Jahren als Schnittstelle zwischen künstlicher Intelligenz und Linguistik. Die Computerlinguistik untersucht, wie natürliche Sprache in Form von Text oder Audio algorithmisch verarbeitet werden kann.
In den 1950er Jahren wurde Sie erstmals verwendet, um russische Sätze auf Englisch zu übersetzen. Nach diesem Georgetown-Experiment wurde ein drei bis fünf-Jahre Zeitfenster gesetzt, um das Problem der maschinellen Übersetzung zu lösen. Dieses Ziel wurde bis zu den späten 1980er Jahren nicht erreicht. [1]
In der 1970er Jahren wurde das Computerprogramm SHRDLU entwickelt. SHRDLU konnte mit englischen Begriffen in einer "Blockwelt" verschiedene Formen bewegen.
Eine Beispielkonversation zwischen dem Nutzer und dem Programm sah folgend aus:

```
Person: PICK UP A BIG RED BLOCK.
Computer: OK. (does it)
Person: GRASP THE PYRAMID.
Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.
Person: FIND A BLOCK WHICH IS TALLER THAN THE ONE YOU ARE HOLDING AND
PUT IT INTO THE BOX.
Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE
ONE I AM HOLDING.
Computer: OK. (does it)
Person: WHAT DOES THE BOX CONTAIN?
Computer: THE BLUE PYRAMID AND THE BLUE BLOCK.
```
[2]

Die 1980er und frühen 1990er Jahre markieren die Blütezeit der Computerlinguistik. Mit der Einführung von maschinellem Lernen und eigene Algorithmen für Computerlinguistik begann eine Revolution. Mit der Entwicklung der Grammatiktheorie Head-Driven Phrase Structure Grammar (HPSG) wurde die Möglichkeit geschaffen Beschränkungen zu formulieren, um Sätze und Satzglieder auf Korrektheit zu prüfen. IBM nutzte zu dieser Zeit große Textkorpora

von dem Parlament von Kanada und der europäischen Union für maschinelles Übersetzen. Diese Textkorpora wurden wegen einer Gesetzesänderung geschrieben, um alle behördlichen Verfahren in den jeweiligen offiziellen Sprachen der Einsatzgebiete verfügbar zu machen. Dies führte dazu, dass die Programme stark abhängig von Informationen spezifisch in diesem Themengebiet geschrieben wurden. Dieses Problem besteht in der Computerlinguistik bis heute. [3]
Mit dem wachsenden "World Wide Web" wurden in den 2000ern viele rohe Sprachdaten gesammelt. Diese Daten wurden mit unüberwachtem Lernen und semiüberwachten Lernen ausgewertet. [3]

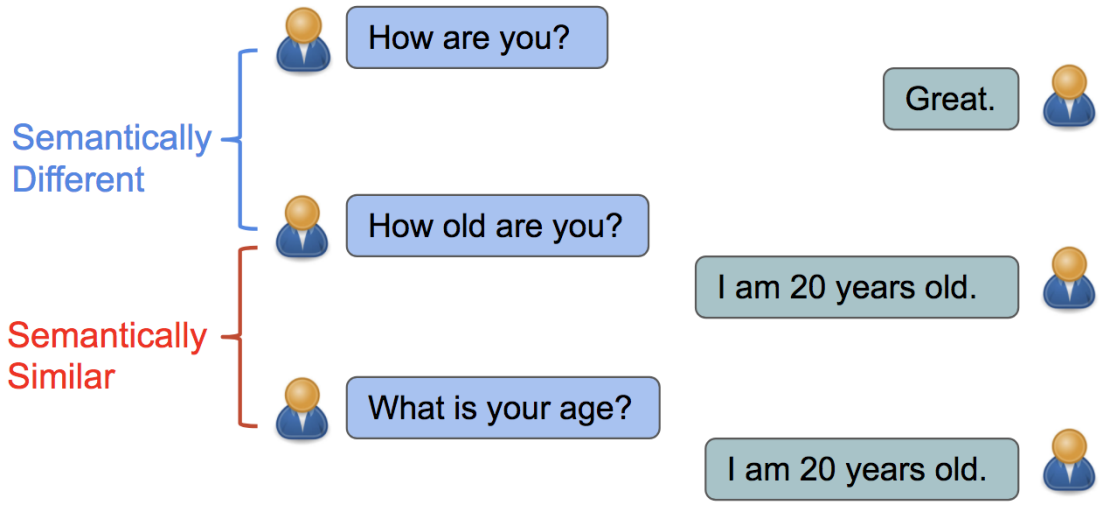
In den 2010er wurden mit Representation Learning und neuronalen Netzwerken die Computerlinguistik weit vorangebracht. Representation Learning beschäftigt sich mit der Frage wie man Daten am besten darstellen kann. 2017 stellte Google Transformer vor. Diese lösen die vorher verwenden rekurrenten Neuronale Netze ab. Später in 2018 wurde BERT , welcher auf Transformern basiert, von Google veröffentlicht und dieser wird seitdem in der Google Suche benutzt, um mögliche Suchanfragen vorherzusagen. Auch im Jahr 2018 wurde GPT (generative pre-training) von OpenAI vorgestellt. Beide Modelle benutzen Transformer und werden bis heute weiterentwickelt. [4]
Ende 2022 veröffentlichte OpenAI ChatGPT, ein Chatbot, der mit Daten bis 2020 vortrainiert wurde. ChatGPT ist seit dem 30. November 2022 in einer kostenlosen Testphase und bietet die Möglichkeit zu einem dialogischen Austausch. [5]

3.2 Sprachwissenschaft

Natürliche Sprache

Natürliche Sprache ist die Form der Kommunikation, die von Menschen genutzt wird, um untereinander Informationen auszutauschen. Sie kann auf verschiedenen Ebenen, wie semantisch und syntaktisch, analysiert werden. Die Erforschung von Natürlicher Sprache stellt ein interdisziplinarisches Forschungsgebiet dar, welches durch die Schnittstellen zwischen Informatik und Sprachwissenschaften erforscht wird.
Bei der semantischen Analyse werden Wörter über ihre eigentliche Bedeutung hinaus interpretiert. Es wird Wert auf den Kontext und inhaltliche Zusammenhänge gesetzt, die nicht aus einzelnen Wörtern hervorgehen. [6]

Die Syntaktische Analyse befasst sich mit der logischen Bedeutung von bestimmten Sätzen oder Teile von Sätzen. Hier wird auch die Grammatik berücksichtigt, welche die logische Bedeutung und Korrektheit der Sätze definiert. [7]



[8]

Sprachmodelle

Sprachmodelle sind Computersysteme, die die Wahrscheinlichkeiten von bestimmten Wortfolgen in einer Sprache berechnen können. Sie finden Nutzen in z.B. automatischer Sprachübersetzung, Spracherkennung oder Generierung von Text. [9]

Ein bekanntes Sprachmodell ist das sogenannte N-Gramm-Modell. Hier wird die Wahrscheinlichkeit von einem Wort anhand der Häufigkeit von diesem mit dem vorhergehenden Wort berechnet. Ein Beispiel dafür wäre: „Ich gehe“ und „zur Arbeit“. Dabei steht das n für die Anzahl Wörter, die in Betracht gezogen werden. In dem Beispiel wird 2-Gramm-Modell benutzt und als Einheit wird ein Wort gesehen. Bei der Wortvervollständigung wird als Einheit einzelne Buchstaben genommen. [10]

Das neuronale Sprachmodell, welches von dem Generative Pre-trained (GPT) Transformer Modell genutzt wird, verwendet neuronale Netze, um die Wahrscheinlichkeit einer bestimmten Wortfolge zu berechnen. Hier werden Wörter in Vektoren umgewandelt und diese Vektoren werden von einem neuronalen Netz verarbeitet.

Beide Sprachmodelle besitzen verschiedene Vor- und Nachteile. Um die Nachteile auszugleichen können beide Sprachmodelle kombiniert werden.

Neuronale Sprachmodelle sind in den letzten Jahren immer bedeutender geworden, weil sie in der Lage sind große Textmengen zu lernen und eine hohe Genauigkeit aufweisen.

Der Vorteil eines N-Gramm-Modells gegenüber des neuronalen Sprachmodells ist der geringe Aufwand, der für das Trainieren benötigt wird. Beide Sprachmodelle profitieren von einem größeren Textkorpus beim Training. Ein Vorteil von dem neuronalen Sprachmodell ist, dass die Wahrscheinlichkeit für ein Wort nicht null betragen kann, da die Einheiten am Anfang des Trainings mit zufälligen kleinen Zahlen initialisiert worden sind. Beim N-Gramm hingegen kann die Wahrscheinlichkeit null auftreten, wenn ein Wort im Korpus kein einziges mal in einem bestimmten Kontext auftritt. [11]

3.3 Stand der Technik

Automatic Summarization

Automatic Summarization ist ein Themengebiet der Computerlinguistik, bei der eine menge Text automatisch auf die wichtigsten Informationen reduziert wird. Hier gibt es zwei verschiedene Techniken: extractive und abstractive Summarization.

Die Extractive Summarization wählt bereits vorhandene Sätze oder Textabschnitte aus einem Text aus, um eine Zusammenfassung zu erstellen. Hier werden hauptsächlich statistische Methoden als Ansatz genutzt, um die Relevanz von Sätzen oder Textabschnitten zu bewerten. Eine Methode ist das Nutzen von Text-Ranking-Algorithmen, bei dem der Text nach Schlagwörter in dem Themenbereich abgesucht wird. Auch werden Wörter, die in einer Sprache oft vorkommen ausgefiltert und es werden die einzelnen Wortarten identifiziert. [9]

Beim Textclustering geht es darum, eine Menge unmarkierter Texte oder Sätze so zu gruppieren, dass die Texte oder Sätze im selben Cluster einander ähnlicher sind als die in anderen Clustern. Text-Clustering-Algorithmen verarbeiten Texte oder Sätze und stellen fest, ob natürliche Cluster (Gruppen) in den Daten vorhanden sind. [12]

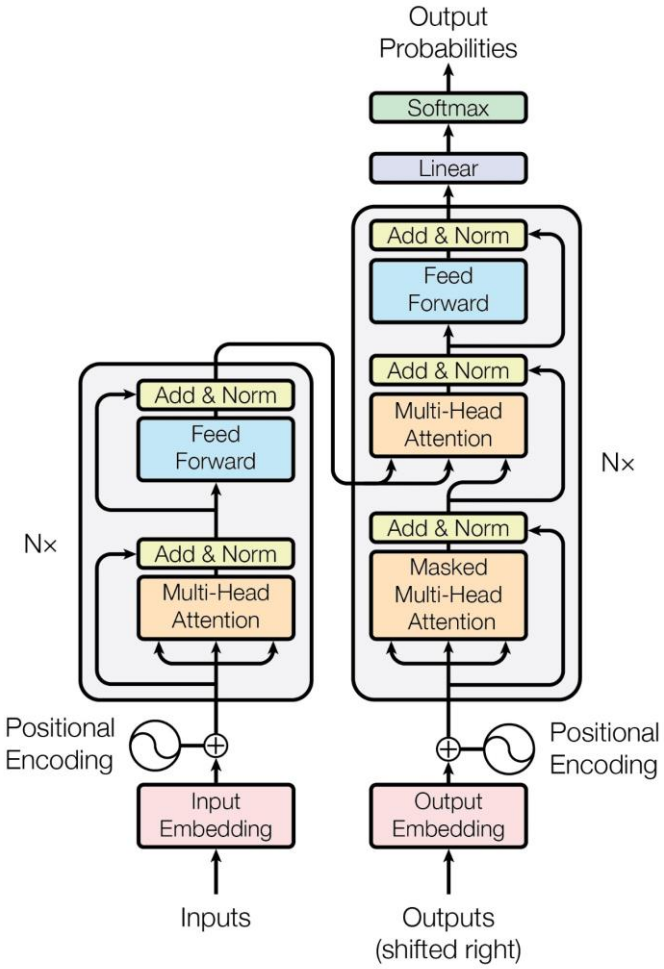
Bei der Abstractive Text Summarization werden Texte verstanden und dann wird eine kurze Zusammenfassung generiert. Hier besteht der Unterschied zur Extractive Summarization darin, dass neuer Text generiert wird und nur die Informationen aus dem Text genutzt werden. Hier werden hauptsächlich Kodier-Dekodier Modelle und Attention- Modelle genutzt, wie zum Beispiel BERT.

Der Nachteil von Abstractive Summarization ist, dass einer größerer Ressourcenaufwand entsteht und es im Gegensatz zur Extractive Summarization eine höhere Fehlerrate auftreten kann. Bei der Extractive Summarization besteht hauptsächlich die Fehlerquelle darin, die Sätze oder Abschnitte falsch zu wiegen, jedoch ist der Inhalt der Sätze und Abschnitte immer noch

gleich. Außerdem benötigt die Abstractive Summarization größere Trainingsdaten, um die Informationen vernünftig formulieren zu können. Ein großer Vorteil der Abstractive Summarization ist, dass der Inhalt in natürlicher Sprache ausgegeben wird, im Gegensatz zu der Extractive Summarization. Hier kann die Bedeutung des Originaltextes verfälscht werden, da lediglich bestimmte Textabschnitte ausgewählt werden. [13]

Transformer

Ein Transformer ist ein neuronales Netzwerk, welches in der NLP verwendet wird und für maschinelles Übersetzen entwickelt wurde. Es wurde erstmals 2017 von Google mit dem Paper "Attention is All You Need" [14] vorgestellt. Der Transformer baut auf dem Attention-Mechanismus auf und setzt die Eingaben in Beziehung zueinander. Dabei trägt das jeweilige Wort durch Kombinationen mit den restlichen Wörtern zum Verständnis der Gesamtdaten bei. Im Gegensatz zu früheren Ansätzen wird hier keine Rekursion oder Konvolution benutzt und so wird bei einem geringeren Rechenaufwand ähnliche oder bessere Ergebnisse erzielt.
Ein Transformer besteht auf in Serie geschalteten Kodierern und in Serie geschaltete Dekodierer. Die Eingabesequenz wird in der Embedding-Schicht in einen Vektor überführt. Ein Kodierer und Dekodierer bestehen aus seinem Self-Attention-Modul und einem Feedforward-Neuronales-Netzwerk, zusätzlich besitzt der Dekodierer ein Kodierer-Dekodierer-Attention-Modul.
Das Self-Attention-Modul akzeptiert Kodierungen von vorherigen Kodierern und wiegt die Relevanz zueinander ab, um Ausgabesequenzen zu generieren. [15]
Die Aufgabe des Feedforward-Neuronale-Netzwerks ist die individuelle Verarbeitung der Ausgabesequenzen und die Weitergabe dieser an die Dekodierer und weitere Kodierer als Eingabesequenzen Dann kommt eine zusätzliche Positionskodierung zum Einsatz, um zum Beispiel einem Wort am Anfang eines Satzes eine andere Repräsentation zu geben. [15]



[14]

Attention

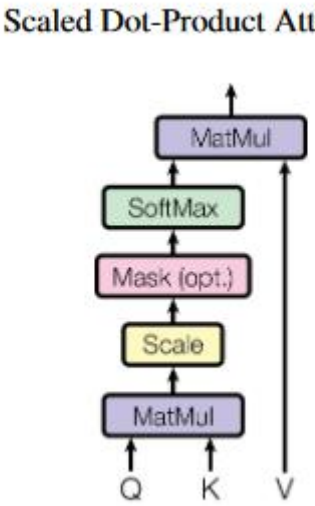
Der Attention-Mechanismus berechnet eine Gewichtung der tokenisierten Eingabe anhand ihrer Relevanz für die aktuelle Aufgabe. Diese Gewichtungen werden dann verwendet, um die Verarbeitung der Eingabe zu steuern und zu modulieren.

Scaled Dot-Product Attention

Bei der Scaled Dot-Product Attention werden insgesamt drei Eingaben benötigt: Query, Key und Value. Es werden die relevanten Token der Eingabe aus den Keys zu einer Query bestimmt. Dabei wird das Skalarprodukt genutzt um die Ähnlichkeit, also wie nah die Vektoren im Raum liegen, zu bestimmen.

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

Um die Genauigkeit zu verbessern wird mit $\sqrt{d}$, der Dimension im Raum, skaliert. Dieses Produkt wird dann in eine Softmax-Funktion gegeben und liefert den Attention-Score. Je wichtiger der Token ist, desto höher ist der Attention-Score. Der resultierende Vektor ist die Ausgabe der Scaled Dot-Product Attention. [14]

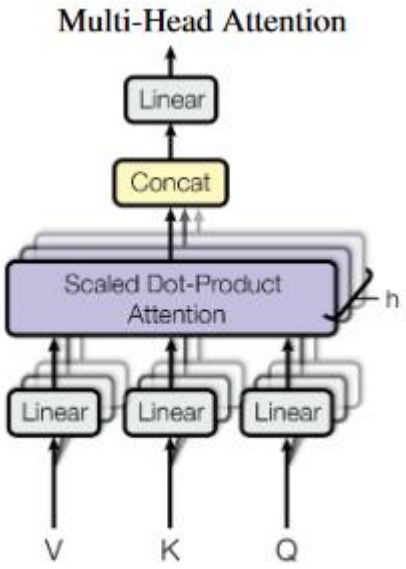


Multi-Head Attention

Anstelle des einfachen Ausführens der Attention Funktion auf dem kompletten Raum wird beim Multi-Head Attention verschiedene Gewichtungen für Keys, Query und Value genutzt, um diese auf kleinere Räume zu projizieren. Diese Projektion, Attention-Head genannt, setzt den Kontext, wie zum Beispiel ein Verb zu einem Objekt zuzuweisen, das nächstmögliche Wort zu generieren oder auch eine Verbindung zu Satzzeichen herzustellen.

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head}_1,...,\mathrm{head}_\mathrm{h})W^O$$
$$\text{where } \mathrm{head}_\mathrm{i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

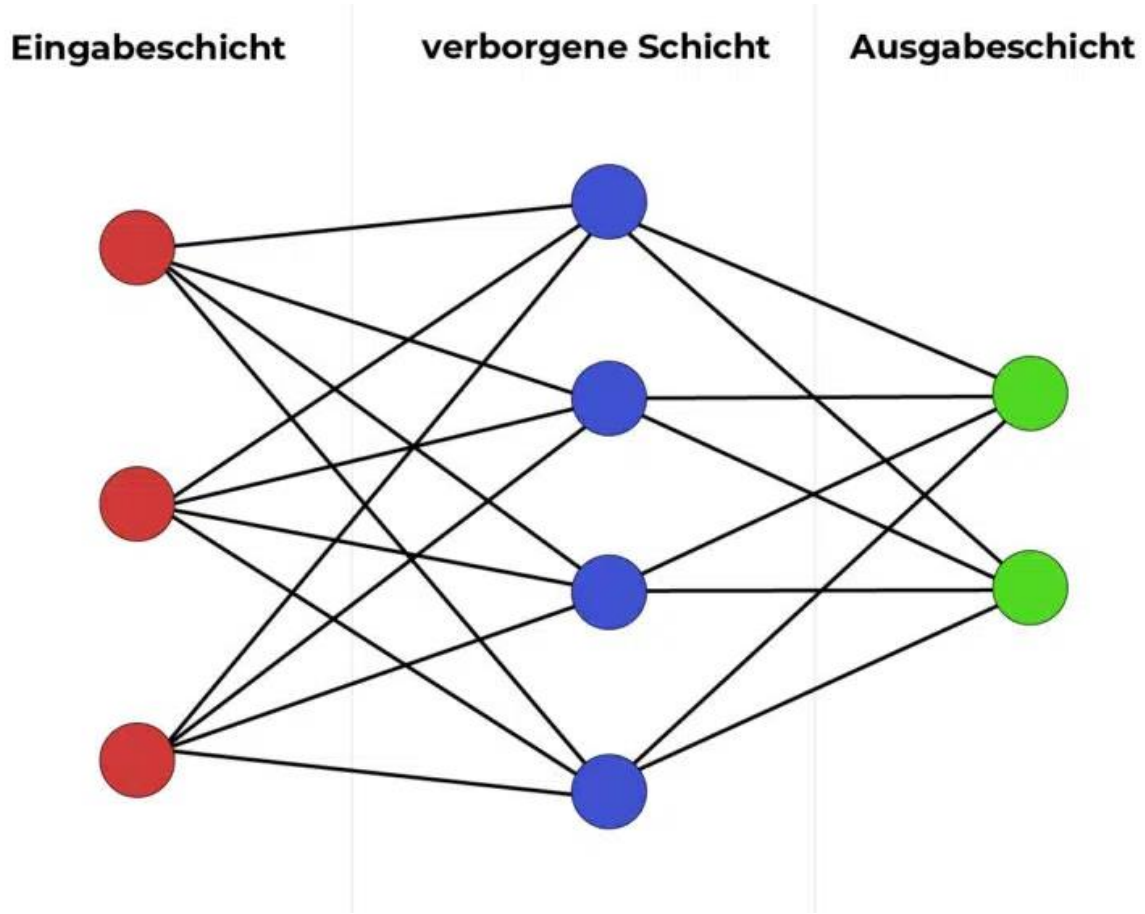
Die Ergebnisse der einzelnen Attention-Heads werden dann konkateniert und nochmal mit den einzelnen Gewichten projiziert. [14]



[16]

Deep Learning

Deep Learning ist eine Technik des maschinellen Lernens, die darauf abzielt, künstliche neuronale Netze zu entwickeln, die in der Lage sind, selbstständig aus großen Mengen an Daten zu lernen und daraus Muster und Verhaltensregeln abzuleiten. Dies geschieht mithilfe von Neuronen, die in Schichten angeordnet sind und miteinander verbunden sind. Die erste Schicht besteht aus Input-Neuronen, die die Eingangsdaten repräsentieren, während die letzte Schicht Output-Neuronen enthält, die das Ergebnis des Lernvorgangs darstellen. Die Schichten dazwischen werden als versteckte Schichten bezeichnet und sind für das tatsächliche Lernen verantwortlich. [17]



Ein Künstliches Neuronales Netzwerk mit einzelnen Schichten [17]

BERT

BERT (Bidirectional Encoder Representations from Transformers) ist eine auf Transformern basierende Technik für das Pretraining von NLP Modellen. Es wurde 2018 von Google als Open Source veröffentlicht und wird hauptsächlich für Frage und Antwort Systeme genutzt. Anders als bei vorherigen Modellen wir bei BERT eine bidirektionale Struktur verwendet, so dass die vorherigen als auch die nachfolgenden Wörter eines jeden Satzes betrachtet werden. BERT hat dadurch seit Veröffentlichung in vielen Sprachmodellen Anwendung gefunden und ist durch seine Performance oft die erste Wahl in der NLP. 2019 hat Google BERT in die englische Google-Suche implementiert und Ende 2020 wurde BERT bei fast jeder englischen Anfrage genutzt. Das englische BERT hat im Ursprung zwei verschiedene Modelle. BERT-Base besteht aus 12 Kodierern und 12 bidirektionalen Self-Attention-Modulen und der BERT-Large aus 24 Kodierern und 16 bidirektionalen Self-Attention-Modulen. Beide BERT-Modelle wurden mit dem englischen Wikipedia mit 2,5 Milliarden Wörtern und dem BookCorpus [18] Datensatz mit ungefähr 980 Millionen Wörtern vortrainiert. [19]

ChatGPT

ChatGPT ist ein Chatbot, der von OpenAI am 30. November 2022 veröffentlicht wurde. ChatGPT befindet sich momentan in einer kostenlosen Testphase (Stand Januar 2023) mit einer voraussichtlichen Monetarisierung in der Zukunft. ChatGPT hat in der ersten fünf Tagen nach Veröffentlichung mehr als eine Millionen Nutzer erreicht.

ChatGPT basiert auf GPT-3.5 und wurde mit überwachtem und bestärktem Lernen angelernt.

ChatGPT hat in den Medien gemischte Reaktionen erhalten. Die New York Times berichtete am 5. Dezember über den Chatbot. Der Artikel berichtet über die verschiedenen Meinungen zu dem Chatbot. Elon Musk Co-gründete 2015 das Unternehmen OpenAI und erklärte, dass die Forschung „die digitale Intelligenz auf eine Art und Weise vorantreiben würde, die der Menschheit am ehesten zugutekommt“, wie es in einem damaligen Blogeintrag hieß

Der Zeitungsartikel geht auch auf die negativpunkte des Chatbots ein, vorangehend den Fakt, dass ChatGPT, wenn mit Falschinformationen angelernt wird, diese als Fakten ansieht. OpenAI sagte, es sei schwierig dieses Problem zu beheben, „weil die ideale Antwort davon abhängt, was das Modell weiß und nicht davon, was der menschliche Nutzer weiß“. [20]

Im Januar 2023 wurden bereits von unter anderem dem New York City Department of Education die Nutzung von ChatGPT verboten oder eingeschränkt. OpenAI arbeitet bereits an einem kryptischen Wasserzeichen, mit dem der von dem Chatbot ausgegebene Text versehen werden soll. Mit diesem Wasserzeichen sollen Plagiate aufgedeckt werden. Dieses Wasserzeichen kann man jedoch umgehen, indem man den von ChatGPT ausgegebenen Text von einem anderen Modell umformulieren zu lassen. [5]

4 Vorgehen

4.1 Vorbereitung

Verwendete Umgebungen

Die Produkte der Consense GmbH nutzen die Programmiersprache Delphi in der Entwicklungsumgebung Embacadero Delphi.

Delphi basiert auch der objektorientierten Sprache Object Pascal und bietet eine Ausführung von Python-Skripten in der Entwicklungsumgebung an.

Python4Delphi bietet keine Syntaxhervorhebung, deswegen wurde für die initiale Implementierung Visual Studio Code verwendet.

Anbieter

OpenAI

OpenAI bietet Ihre cloudbasierte GPT-3 API in für verschiedene Anwendungszwecke an. GPT-3 wird für folgende Anwendungen angeboten: Verfassen von Werbetexten, Textzusammenfassung, Analyse von unstrukturiertem Text mit Fragenbeantwortung, Klassifizierung von Inhalten und Textübersetzung. Außerdem wird Codex angeboten, welcher als Programmierhilfe verwendet werden kann. Die Kosten werden pro 1000 Token verrechnet, Wobei 1000 Token ungefähr 750 Wörtern entsprechen. Die Sprachmodelle kosten von 0,0004$ bis zu 0,02$ pro 1000 Token. [21]

Azure OpenAI Service

Azure OpenAI Service bietet REST-API-Zugriff auf die Sprachmodelle von OpenAI. Zusätzlich wird hier ein Fokus auf Sicherheit gelegt und anders als bei OpenAI kann hier in privaten Netzwerken gearbeitet werden. [22]

Open Source BERT

Der BERT-Extractive Summarizer ist eine progammschnittstelle zur extraktiven Textzusammenfassung mit MIT-Lizenz. Die Schnittstelle beinhaltet verschiedene Modelle, die sich bei der Genauigkeit und dem davon abhängigen Ressourcenaufwand unterscheiden. Der Summarizer kann über ein Docker-File ausgeführt werden und kann GPUs zur berechnung benutzen. [23]

4.2 Implementierung

Als ersten Ansatz wurde mit einem Codebeispiel gearbeitet.

```
import openai
import wget
import pathlib
import pdfplumber
import numpy as np


def getPaper(paper_url, filename="https://arxiv.org/pdf/1808.04295.pdf"):
    downloadedPaper = wget.download(paper_url, filename)
    downloadedPaperFilePath = pathlib.Path(downloadedPaper)

    return downloadedPaperFilePath


def showPaperSummary(paperContent):
    openai.organization = 'org-K95H9B7j3gAZkvoB--------'
    openai.api_key = "sk-5v0jHWYgmkcDJRspJ5ziT3BlbkFJ9IhgBIeodHO5--------"
    engine_list = openai.Engine.list()

    for page in paperContent:
        text = page.extract_text() + tldr_tag
        response = openai.Completion
                        .create(engine="davinci", prompt=text,
                                                temperature=0.3,
                                                max_tokens=140,
                                                top_p=1,
                                                frequency_penalty=0,
                                                presence_penalty=0,
                                                stop=["\n"]
                                                )
        print(response["choices"][0]["text"])


paperFilePath = "./random_paper.pdf"
paperContent = pdfplumber.open(paperFilePath).pages
showPaperSummary(paperContent) [24]
```

In diesem Programm wird eine PDF Datei mithilfe von PDFPLumber geöffnet und dann als Text mithilfe von „DaVinci“ von OpenAI zusammengefasst. Die Anfrage wird in der Cloud bearbeitet.

Das Gegenstück dazu bietet der lokal arbeitende BERT-Extractive-Summarizer. Die Einbindung dieser API sieht wie folgt aus.

```
from summarizer import Summarizer

body = 'Text body that you want to summarize with BERT'
model = Summarizer()
model(body)
```

Es muss lediglich der Summarizer importiert werden, dieser hat eine Reihe an optionalen Parametern:

```
super(Summarizer,self).__init__(model,custom_model,custom_tokenizer,hidden,re-
duce_option,sentence_handler,random_state,hidden_concat,gpu_id)
```

Die Parameter bedeuten: Das Standardmodell welches genutzt wird, es sind auch andere Modelle neben BERT möglich. Des Weiteren kann man einen eigenen Tokenizer nutzen. Der „hidden“ Parameter beschreibt die hidden output layers, die für die Zusammenfassung genutzt werden sollen. Die Reduce-Option konfiguriert Bedingungen , die die Zusammenfassung erfüllen muss, zum Beispiel „min“ für die Minimale Ausgabe. Randomstate beschreibt den RandomState der genutzt werden soll. Hidden_Concat wird als depracted angesehen und schlussendlich kann man eine GPU ID eingeben, falls diese zur Verfügung steht.

4.3 Nachbereitung

Bei der Implementierung von dem Summarizer via Python4Delphi kam es zu Komplikationen mit den Bibliotheken und konnte deswegen noch nicht genutzt werden.

Während der lokalen Nutzung von dem BERT-Extractive Summarizer war schnell zu erkennen, dass sehr große Texte und hohe Genauigkeit einen großen Ressourcenaufwand mit sich bringen. Zu diesem Zeitpunkt war ein Zugriff auf ein Gerät mit GPU nicht möglich, wodurch die Performance des Programms profitiert.

5 Ergebnis

Der heutige Markt bietet viele Möglichkeiten an, Texte in seiner Software zusammenfassen zu lassen. Ein wichtiger Punkt, um sich für einen Anbieter zu entscheiden, ist der Datenschutz. Das direkte Angebot von OpenAI operiert in der Cloud, deswegen besteht die Möglichkeit eines Datenlecks. Der Azure Service bietet im Gegensatz dazu private Netzwerke an, mit denen die Sicherheit erhöht ist. Die lokale Version von BERT bietet die größte Sicherheit in facto Datenschutz und ist deswegen auch der gewählte Lösungsansatz. Der BERT-Extractive Summarizer kann auf einem AWS-System mit GPU implementiert werden, um die Genauigkeit der Zusammenfassungen zu erhöhen.

6 Fazit

Der BERT-Extractive Summarizer bietet noch vielseitige Nutzungsmöglichkeiten. Man könnte die API auf einem System integrieren, welches eine GPU besitzt. Des Weiteren ist zu Testen welche Einstellung und welcher Ressourcenaufwand am Besten für den Kundeneinsatz sind.
Der Azure-OpenAI-Service ist außerdem eine sinnvolle Alternative, da dort auch eine Abstractive Summarization möglich ist.

7 Literaturverzeichnis

[1] J. Hutchins, „The history of machine translation in a nutshell,“ 2014.

[2] S. Edu, "SHRDLU," Stanford Edu, [Online]. Available: http://hci.stanford.edu/winograd/shrdlu/.

[3] Wikipedia, „Natural Language Processing,“ Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Natural_language_processing.

[4] Wikipedia, "GPT-3," Wkipedia, [Online]. Available: https://en.wikipedia.org/wiki/GPT-3.

[5] Wikipedia, „ChatGPT,“ [Online]. Available: https://de.wikipedia.org/wiki/ChatGPT.

[6] Seiten-Werk, „Semantische Analyse,“ Seiten-Werk, 15 Juni 2021. [Online]. Available: https://seiten-werk.com/lexikon/semantische-analyse/.

[7] J. Thanaki, "Python Natural Language Processing by Jalaj Thanaki," O'Reilly, [Online]. Available: https://www.oreilly.com/library/view/python-natural-language/9781787121423/273cc324-4cfa-47bd-8238-7c0875d71039.xhtml.

[8] C. T. Yinfei Yang, "Advances in Semantic Textual Similarity," Google Research, 17 Mai 2018. [Online]. Available: https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html.

[9] J. Wijffels, „Textrank for summarizing text,“ 10 Dezember 2020. [Online]. Available: https://cran.r-project.org/web/packages/textrank/vignettes/textrank.html.

[10] Wikipedia, "N-gram," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/N-gram.

[11] R. Labryga, „https://nats-www.informatik.uni-hamburg.de/pub/VGS20/WebHome/labryga-lms-draft.pdf,“ Uni-Hamburg, August 2020. [Online]. Available: https://nats-www.informatik.uni-hamburg.de/pub/VGS20/WebHome/labryga-lms-draft.pdf.

[12] a. Sudha-Nadchal, "Text Clustering," 19 Januar 2022. [Online]. Available: https://devopedia.org/text-clustering.

[13] M. E. B. M. Nouf Ibrahim Altmami, „Automatic summarization of scientific articles: A survey,“ ScienceDirect, April 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157820303554.

[14] N. S. P. U. J. N. G. K. P. Ashish Vaswani, „Attention Is All You Need,“ arxiv, 6 Dezember 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762v5.pdf.

[15] Wikipedia, "Transformer (machine learning model)," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Transformer_(machine_learning_model).

[16] U. K. L. D. M. Kevin Clark, „What does BERT look at?,“ Computer Science Department, Stanford University,Facebook AI Research, 1 August 2019. [Online]. Available: https://aclanthology.org/W19-4828.pdf.

[17] L. Wuttke, „Was ist Deep Learning ?,“ datasolu, [Online]. Available: https://datasolut.com/was-ist-deep-learning/.

[18] R. K. Z. S. U. T. F. Yukun Zhu, „Aligning Books and Movies: Towards Story´-like Visual Explanations by Watching Movies and Reading Books,“ University of Toronto,Massachusetts Institute of Technology, 22 Juni 2015. [Online]. Available: https://arxiv.org/pdf/1506.06724.pdf.

[19] M.-W. C. L. T. Jacob Devlin, „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,“ Google AI Language, 24 Mai 2019. [Online]. Available: https://arxiv.org/pdf/1810.04805.pdf.

[20] S. Lock, "The Guardian," 5 Dezember 2022. [Online]. Available: https://www.theguardian.com/technology/2022/dec/05/what-is-ai-chatbot-phenomenon-chatgpt-and-could-it-replace-humans.

[21] OpenAI, „OpenAI Pricing,“ OpenAI, [Online]. Available: https://openai.com/api/pricing/.

[22] Azure, „Azure OpenAI Service,“ Azure, [Online]. Available: https://azure.microsoft.com/en-us/products/cognitive-services/openai-service/#layout-container-uid189e.

[23] D. Miller, „Leveraging BERT for Extractive Text Summarization on Lectures,“ Georgia Institute of Technology, [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1906/1906.04165.pdf.

[24] J. H. M. Dan Jurafsky, "Speech and Language Processing," Stanford, 7 Januar 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/.