DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

Fakultät Wirtschaft

Studiengang Wirtschaftsinformatik

Test

2. Projektarbeit

Im Rahmen der Prüfung zum Bachelor of Science (B. Sc.)

Sperrvermerk

31. August 2020

VerfasserIn:	Test
Kurs:	WWI22B5
Dualer Partner:	Musterfrau AG, Karlsruhe
Betreuer der Ausbildungsfirma:	Leonie Musterfrau
Wissenschaftlicher BetreuerIn:	Prof. Dr. Tina Mustermann
Abgabedatum:	31. August 2020

Selbstständigkeitserklärung

Ich versichere hiermit, dass ich die vorliegende 2. Projektarbeitmit dem Thema:

Test

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, 31. August 2020, _____

Test

Sperrvermerk

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung der Dualen Partners vorliegt.

Kurzfassung

Hier beginnt die Kurzfassung ihrer wissenschaftlichen Arbeit...

Inhaltsverzeichnis

Abkürzungsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Kontext und Relevanz des Themas

Keine Entwicklung der Welt der IT ist aktuell so viel besprochen wie die der künstlichen Intelligenz. Speziell durch den Aufstieg von generativer KI hat sich das Thema zu einer geradezu gesamtgesellschaftlich relevanten Entwicklung herangebildet. Maßgeblich angestoßen durch die Veröffentlichung von OpenAI's GPT-3 Modell, welches in der Lage ist, Texte zu generieren, die von menschlichen Texten nur noch schwer zu unterscheiden sind, hat sich die öffentliche Aufmerksamkeit auf die Möglichkeiten von generativer KI gerichtet. Die Konzepte und Technologien, die hinter diesen Entwicklungen stehen, sind dabei nicht unbedingt neu, eine breitere Verfügbarkeit von Rechenleistung und Trainingsdaten haben jedoch den entscheidenen Anstoß für die neue Leistungsfähigkeit dieser Modelle gegeben. KI ist also das das Thema der Stunde und als strategisch relevantes Thema für Unternehmen und Organisationen nicht mehr wegzudenken.

1.2 Ziel der Arbeit

Die vorliegende Arbeit beschäftigt sich mit einem konkreten Anwendungsfall von KI in der Praxis. Untersucht wird ein Beispiel, in dem eine semantische Datenbanksuche auf einem Materialstammdatensatz durchgeführt wird. Der Mehrwert dieses Systems liegt dabei in der Möglichkeit für Anwender, die Datenbank auf natürlichsprachliche Weise zu durchsuchen, ohne dabei auf die spezifischen Suchbegriffe und -syntaxen achten zu müssen, die in traditionellen Datenbanksystemen notwendig sind und gleichzeit von einem gewissen semantischen Verständnis des Suchsystems profitieren zu können. Die Technik aus dem Feld der KI, die für dieses System zum Tragen kommt sind sogenannte *word embeddings*, die es ermöglichen, Worte in einem Vektorraum abzubilden und so semantische Ähnlichkeiten zwischen Wörtern zu berechnen. Dieses Konzept wird in der Arbeit genauer erläutert die Effektivität verschiedener Techniken zur Erstellung von embeddings im konkreten Anwendungs-

fall beleuchtet. Das Ziel der Arbeit ist es, die Technik im Anwedungsfall zu erläutern, verschiedene Methoden und Modelle zu beleuchten und eine datengestützte Entscheidungsgrundlage für die Bewertung von word embeddings in semantischen Suchsystemen zu schaffen.

2 Theoretischer Hintergrund

2.1 Maschinelles Lernen

Maschinelles Lernen beschreibt das Konzept, auf Basis von einer großen Menge von Daten Algorithmen zu approximieren, die auf anderem Wege nicht erschlossen werden können. Man nehme beispielsweise die klassische Aufgabe, ein Programm zu schreiben, das in der Lage ist, Bilder von Hunden und Katzen zu unterscheiden. Wenn wir als Menschen uns dieser Aufgabe stellen, müssen wir nicht lange überlegen, wir lösen sie intuitiv. Wenn wir uns aber fragen, nach welchen Regeln wir diese Entscheidung treffen, wird es schon schwieriger. Wir könnten uns auf die Form der Ohren, die Farbe des Fells oder die Größe des Tieres konzentrieren. Aber wie genau wir Regelmäßigkeiten definieren, ist nicht so einfach. Maschinelles Lernen verfolgt den Ansatz, genau solche Regeln nicht mehr fest zu definieren, sondern sie anhand von einer großen Menge von Daten zu lernen (Vgl. Bishop, 2006, S. 2f.).

2.2 Neuronale Netze

Ein Mittel der Wahl um das Konzept des maschinellen Lernens umzusetzen, sind sogenannte künstliche Neuronale Netze. Neuronale Netze, lose inspiriert von der Struktur des menschlichen Gehirns, bestehen aus einer Vielzahl von einfacher Einheiten, sogenannte Knoten, die in Schichten angeordnet sind und über unterschiedlich gewichtete Verbindungen verknüpft sind. Diese Struktur ermöglicht es, komplexe statistische Zusammenhänge in einem Datensatz zu modellieren, indem für einen gegebenen Datensatz mithilfe von Techniken des maschinellen Lernens die Parameter, also beispielsweise die Gewichte der Verbindungen des Netzes, so angepasst werden, dass sie die gegeben Daten möglichst genau abbilden. Wurde dieser Prozess erfolgreich durchlaufen, so kann das Modell im Anschluss dazu genutzt werden, Aussagen über Daten, die es im Lernprozess noch nie gesehen hat, zu treffen oder Vorhersagen abzugeben. Das interessante an diesem Ansatz ist es, dass durch diesen Ansatz, gerade bei großen neuronalen Netzen auch nicht triviale, subtile Muster im Daten-

satz erkannt werden können und so, wie oben bereits angedeutet, Approximationen für Probleme getroffen werden können, die formal nur schwer beschrieben werden können (Vgl. Bishop, 2006, S. 225f.).

2.3 Word Embeddings

Eine weitere, für diese Arbeit relevante Entwicklung der jüngeren Forschung sind die Fortschritte der Computerlinguistik. Ein Kernproblem dieses Feldes ist die Forschung an der Repräsentationen von Sprache. Hierbei geht es nicht einfach darum, einzelne Wörter in ihrer Schriftform zu speichern, sondern vielmehr den Wort*sinn* festzuhalten. Man betrachte zum Beispiel die Wörter *Couch* und *Sofa*, die in Schriftform, mit Ausnahme des zweiten Buchstabens vollkommen unterschiedlich sind, in ihrer Bedeutung aber nahezu Synonym verwendet werden. Weiterhin möchten wir Aussagen über die Beziehung von Wörtern treffen können. *Heiß* und *kalt* haben in ihrer Wortbedeutung einen klaren Zusammenhang (Es handelt sich um Gegensätze), den wir eventuell darstellen möchten, genauso wie *Replika* und *Fälschung* im Grunde dasselbe meinen, aber einen klaren Unterschied in ihrer Konnotation aufweisen (Vgl. Jurafsky und Martin, 2020, S. 106ff.). Eine Form der semantisch reichen Repräsentationen zu finden, die diesen Anforderungen genügt ist nicht trivial, es handelt sich aber wieder um ein solches Problem, das, wie oben beschrieben, intuitiv einfach zu lösen, formal jedoch schwer zu beschreiben ist. Und genau wie oben beschrieben, können die Techniken aus dem Feld des maschinellen Lernens auf dieses Problem angewandt werden, um es zu lösen.

Die Grundlage für die nun folgenden Überlegungen bildet die 1950 erstmals formulierte Verteilungshypothese der Linguistik. Im Grunde besagt sie, dass Wörter, die in ähnlichen Kontexten auftauchen, eine ähnliche Bedeutung haben. Wenn die Wörter *Pizza* und *Burger* beispielsweise beide häufig im Zusammenhang mit den Wörtern *Essen* und *geniessen* auftauchen, kann daraus geschlossen werden, dass sie ihr Wortsinn eine Ähnlichkeit hat, in diesem Fall, dass es sich bei beiden Wörtern um Essen handelt. (Vgl. Jurafsky und Martin, 2020, S. 109)

	Essen	italienisch	Auto
Pizza	150	122	11
Burger	136	3	13
Porsche	0	6	350
Ferrari	1	199	475

Tabelle 1: Wort-Wort-Matrix auf Basis des Wikipedia Corpus und ausgewählten Worten (Davies, 2015)

Auf Basis dieser Erkenntnis kann eine erste simple Repräsentationen des Wortsinns gefunden werden. Gegeben sei ein Corpus C auf Basis dessen wir einen Wortsinn für jedes Wort im Vokabular V des Corpus finden wollen. Auf Basis der Verteilungshypothese kann nun eine Wort-Wort-Matrix aufgestellt werden, die abbildet, wie häufig Worte im Kontext anderer Worte auftauchen. Dafür muss ein Kontext definiert werden, häufig ist dieser Kontext ein Bereich um das Wort, kann aber auch beliebig definiert werden, beispielsweise als eine Menge Dokumente im Corpus. Als Ergebnis erhält man eine Matrix mit der Dimension $|V| \times |V|$, beziehungsweise einen $|V|$-dimensionalen Spaltenvektor für jedes Wort (Vgl. Jurafsky und Martin, 2020, S. 113).

Die so gewonnen Vektoren haben bereits einige der gewünschte Eigenschaften um den Wortsinn zu encodieren. Die in Tabelle 2.3 dargestellte Wort-Wort-Matrix basiert auf dem Corpus der englischsprachigen Wikipedia und illustriert eine Eigenschaft, die auch noch bei der Reduktion auf wenige Dimensionen sichtbar wird: Wörter, die sich ähnlich sind, tauchen in ähnlichen Kontexten auf. Die Vektoren für Automarken tauchen beide ähnlich häufig im Kontext des Wortes Auto auf, genauso wie Essbares ähnlich häufig im Kontext von dem Wort Essen auftauchen. Auch eine 1970 entwicklelte Methode zur Beschreibung von Analogieproblemen lässt sich anwenden. (Vgl. Rumelhart und Abrahamson, 1973) Die Idee dieser Methode ist es Fragen nach folgendem Schema zu stellen: *Deutschland gehört zu Berlin. Was gehört zu Paris?* Auch bei diesem sehr simplifizierten Beispiel lässt sich ein solches Sinn-Parallelogram aufstellen. Anhand der in Abbildung 1 getroffenen Visualisierung des

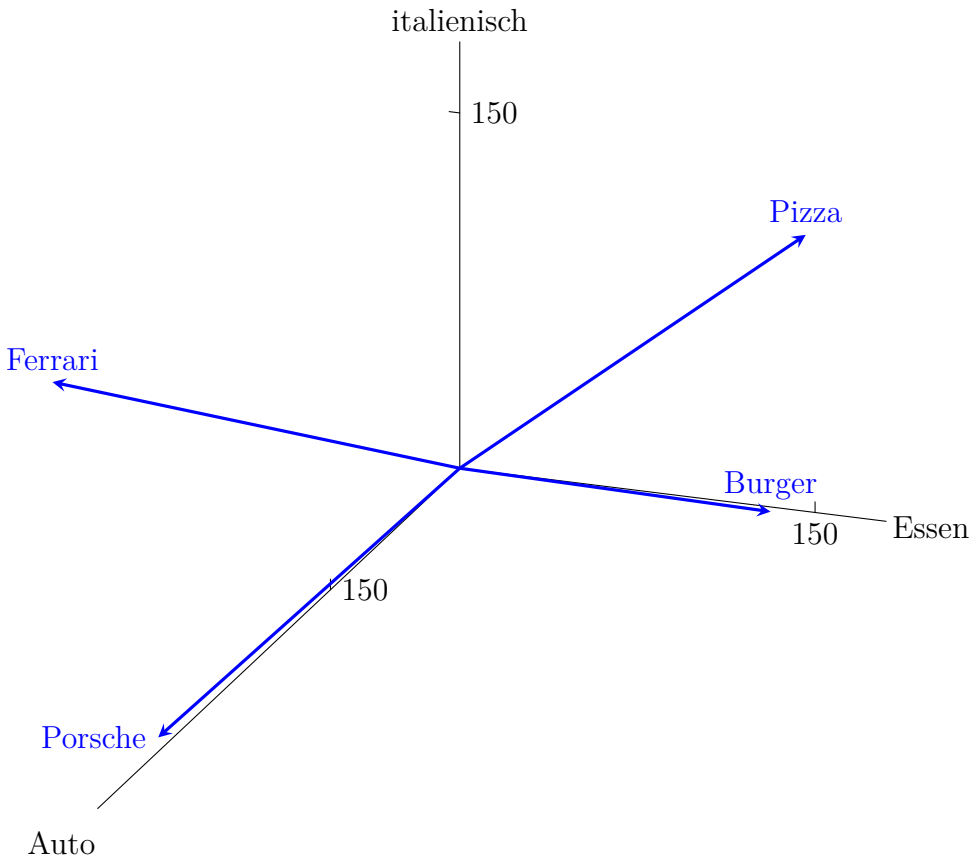

Abbildung 1: Visualisierung der in Tabelle 2.3 berechneten Vektoren in dreidimsionalem Raum

Beispiels lässt sich leicht erkennen, dass dieses Problem durch einfach Vektorrechnung lösen lässt. *Pizza gehört zu Burger. Was gehört zu Porsche?*. Die Antwort innerhalb dieses Beispiels ist Ferrari und lässt sich bestimmen, indem der Vektor für Pizza von Burger subtrahiert wird und das Ergebnis dieser Operation auf den Vektor für Porsche aufaddiert wird. Das Ergebnis ist ein Vektor der in die Nähe von Ferrari zeigt. Dieses Beispiel ist selbstverständlich enorm simplifiziert, es lässt sich aber genau diese Encodierung von Wortsinn auch bei größeren Vokabularien feststellen[1] (Vgl. Jurafsky und Martin, 2020, S. 128f.).

Vorgestellt wurde die hiermit die einfachste Form von statischen, also kontextunabhängige, Embeddings, mittlerweile gibt es eine Vielzahl von Techniken um dieses stumpfe Zählen von Worten mit verschiedenen Verfahren zu optimieren und schlussendlich bessere und sinnhaltigere Ergebnisse zu erzielen. Ein Problem dieser Methode ist beispielsweise, dass die so gewonnen Vektoren eine sehr hohe Dimensionalität haben ($|V|$), dafür aber größtenteils leer sind, man spricht von dünnbesetzten Vektoren. Die Empirik hat gezeigt, dass sich wesentlich bessere Ergebnisse erzielen lassen, wenn Wörter als Vektoren mit niedrigerer Dimensionalität dargestellt werden, also "dicht"sind. Die Intuition hinter dieser Erkenntnis ist es, dass dadurch eine gewisse "Abstraktion"des Wortsinnes stattfindet und so konzeptuelle Zusammenhänge besser abgebildet werden können. (Vgl. Jurafsky und Martin, 2020, S. 113f.)

Ein großer Durchbruch bei diesem Problem wurde beispielsweise durch den Ansatz der Autoren des Word2Vec Papers erzielt, die ihrem Algorithmus *Skip-Gram with negative sampling* ein statisches, aber dichtes Embedding berechnen. SGNS ist eine Methode, um Wort-Embeddings mithilfe von Techniken des maschinellen Lernens zu trainieren, indem ein Modell trainiert wird, das die Kontextwörter um ein Zielwort innerhalb eines Fensters vorhersagen soll, während gleichzeitig echte Kontextwörter von zufällig ausgewählten negativen Beispielen unterschieden wer-

[1] Einer der wohl bekanntesten Beispiele aus diesem Feld ist der Erfolg aus dem word2vec paper, indem das Modell den Zusammenhang zwischen König minus Mann plus Frau ist gleich Königin herstellen konnte (Vgl. Mikolov u. a., 2013).

den. Das zu Grunde liegende statistische Modell wird dabei immer besser darin, gegeben ein Wort, zu bestimmen, ob ein anderes Wort wahrscheinlich ist in dessen Kontext aufzutauchen. Die eigentlichen Embeddings, die das Ziel dieses Vorgehens sind, sind dabei die Eingabe in den Algorithmus und werden solange angepasst, bis das Modell mit einer gewünschten Verlässlichkeit vorhersagen kann, ob die Eingabe ein echtes Kontextwort ist, oder nicht. (Vgl. Mikolov u. a., 2013).

2.4 Text Embeddings und die Transformer Architektur

Den bisher vorgestellen Methoden, um semantisch reiche Repräsentationen von Worten zu erstellen, fehlt ein entscheidendes Merkmal, um sie für praktisch nutzen zu können: Ihnen fehlt der Kontext. Bis jetzt haben wir immer ausschließlich versucht, einzelne Wörter in ihrer Bedeutung zu erfassen, ohne dabei auf den Kontext einzugehen, in dem sie auftreten. Ein Wort wie *Bank* kann beispielsweise sowohl ein Möbelstück, als auch eine Finanzinstitution beschreiben. Die Bedeutung des Wortes hängt also stark vom Kontext ab. Für die letztendliche Anwendung ist das insofern relevant, als dass bei der Analyse von Geschäftsdaten immer gesamte Dokumente von Bedeutung sind. Was bringt es mir, wenn ich die Bedeutung einzelner Wörter in der Beschreibung eines Bauteils kenne, nicht aber ihren Zusammenhang untereinander?

Es muss also eine Methode gefunden werden, Embeddings nicht auf einzelnen Wörtern zu erstellen, sondern ganze Texte zu verarbeiten. Der aktuelle Stand der Technik ist dabei die Transformer Architektur. Hierbei handelt sich auch um die Technik, die Anfang 2021 von OpenAI in Form von ChatGPT zu weltweiter Bekanntheit gelangt ist. Die Grundidee dieser Architektur ist es, über eine Reihe von sogenannten Transformer-Blöcken graduell die Eingabewörter mit ihrem Kontext anzureichern.

Die in Abbildung 2 dargestellte Architektur zeigt den Aufbau eines solchen Transformers. An Anfang des Modells stehen die Eingabe-Embeddings. Ähnlich der Word-

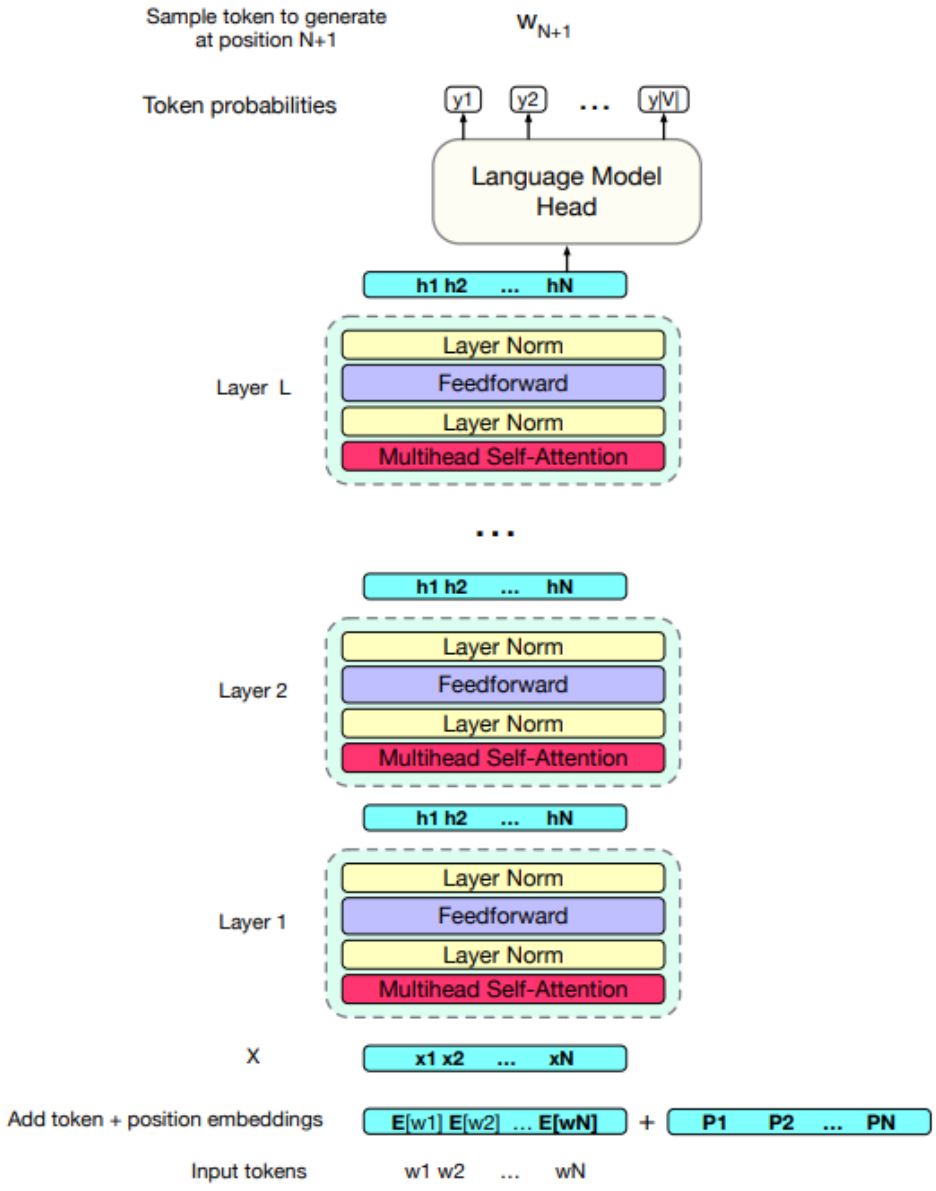

Abbildung 2: Die Architektur eines Transformers für die Anwendung als Sprachmodell (Jurafsky und Martin, 2020)

Embeddings der Word2Vec Algorithmus handelt es sich hierbei um dichte Vektoren, die im Rahmen des Training des Modells erlernt werden und einzelne Wörter darstellen.[2] Das interessante an der Transformer-Architektur ist es nun aber, dass sie, ähnlich wie Word2Vec, ein sogenanntes Kontext-Fenster betrachtet, also eine geordnete Menge von Worten, um ein Wort eben nicht statisch, sondern im Kontext seiner direkten Nachbarn zu betrachten. Der entscheidende Unterschied zum Word2Vec Algorithmus ist dabei aber, dass beim Transformer auch die Position und nicht einfach nur die Häufigkeit des nebeneinander Auftretens einbezogen wird. Um das zu erreichen, werden die Word Embeddings des Kontextfensters mit sogenannten Positions-Embeddings angereichert. Ein Positions-Embedding kann dabei wieder ein gelernter Vektor für jede Position im Kontextfenster sein, oder auch eine simple Funktion, die beispielsweise durch eine Kombination von Sinus- und Kosinusfunktionen einer Position einen Vektor zuordnet.[3]

Formal gesprochen ergeben sich also eine Embedding-Matrix E der Form $|V| \times d$ wobei d die Dimension des Word-Embedding ist, die für jedes Wort im Vokabular einen gelernten Vektor enthält, und eine Positions-Matrix P der Form $N \times d$, wobei N die Größe des Kontextfensters ist und d wieder die gewählte Dimension. Die Eingabe in den Transformer soll nun jedes Wort im Kontext-Fenster, angereichter mit seiner Position sein. Diese Matrix X berechnet sich nun folgendermaßen:

$$X_i = e_i \in E + p_i \in P \forall i \in N$$

Die Eingabe ist also eine Matrix der Form $N \times d$. Diese Matrix wird im folgenden durch das Kernstück des Transformers, die Transformer-Blöcke, gegeben. Interessanterweise wird dabei die Form der Matrix beibehalten, das gibt dem Transformer auch

[2] Es bleibt anzumerken, dass diese Word-Embeddings sich in den meisten Fällen nicht mehr auf Wörter im Sinne der Sprache beziehen, sondern der zugrundeliegende Corpus algorithmisch in kleinste Einheiten zerlegt wird. In der Literatur werden diese Einheiten häufig auch als Tokens bezeichnet.

[3] Es gibt durchaus noch weitere Techniken, die Position eines Wortes als Embedding darzustellen, beispielsweise als relative Position zum betrachteten Wort, das soll in dieser Arbeit aber kein Thema sein.

seinen Namen.

Im folgenden sollen die Blöcke näher betrachtet werden. Jeder dieser Blöcke folgt dabei demselben Muster und in modernen Modellen sind teilweise über 100 dieser Blöcke übereinander angeordnet. Die Intuition hinter diesem Aufbau ist es, mit jedem Block einen kleinen Schritt weiter in der Abstraktion zu gehen und so auch komplexe Zusammenhänge abbilden zu können. Ein Block selbst besteht dabei aus verschiedenen, in Abbildung 2 dargestellten Komponenten. Den Anfang macht die Multihead-Self-Attention-Schicht.[4] Sie bildet im Grunde das Herzstück des Blockes, weil sie diejenige Schicht ist, die den Zusammenhang zwischen den verschiedenen Wörtern des Kontext-Fensters herstellt. Multihead-Self-Attention vereint in sich einigermaßen viele Funktionen, deshalb soll an dieser Stelle nur kurz auf das Prinzip des Mechanismus eingegangen werden. Für jedes Word-Embedding wird das Skalarprodukt zu jedem anderen Wort-Embedding gebildet. Der daraus entstandene Wert wird als Gewicht verwendet, um schließlich alle Word-Embeddings des Kontext-Fensters gewichtet aufzusummieren. Zudem gibt es für jedes Wort in seiner Funktion als Zielwort

[4]Diese Komponente bildet auch die Kern-Innovation, die ChatGPT zu seinem massiven Qualitätssprung verholfen hat und den Namen des zugehörigen Papers begründet: Attention is all you need –vgl

3 Ist- und Problemanalyse

4 Optimierung des Systems

5 Fazit

Quellenverzeichnis

Bücher

Bishop, Christopher (2006). *Pattern Recoginition and Machine Learning*. Hrsg. von Kleinberg, Schölkopf und Jordan. Springer. URL: https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf.

Jurafsky, Daniel und James Martin (2020). *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition*. draft. URL: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf.

Artikel

Mikolov, Tomas u. a. (2013). „Efficient Estimation of Word Representations in Vector Space“. In: URL: https://arxiv.org/pdf/1301.3781.

Rumelhart, David E und Adele A Abrahamson (1973). „A model for analogical reasoning“. In: *Cognitive Psychology* 5.1, S. 1–28. ISSN: 0010-0285. DOI: https://doi.org/10.1016/0010-0285(73)90023-6. URL: https://www.sciencedirect.com/science/article/pii/0010028573900236.

Internetquellen

Davies, Mark (2015). *English-Corpora: Wikipedia*. URL: https://www.english-corpora.org/wiki.

Anhang

1. Digitale Version der Arbeit
2. Interviews
    2.1. Expertmann 2018