Programmieraufgaben - Seminar zu Generativer KI

Im Rahmen dieses Seminars absolvieren Sie drei Programmieraufgaben in Kleingruppen, deren Bewertung am Ende des Seminars Ihre Note ausmachen wird. Abzugeben sind dabei neben dem Code der Aufgabe zusätzlich ein Bericht, in dem die Herangehensweise an die Aufgabe erläutert wird sowie die Fragen der Aufgabe beantwortet werden.
Der Bericht sollte auch die zentralen Erkenntnisse aus der Programmieraufgabe enthalten und damit nachweisen, dass Sie die Aufgabe bearbeitet haben.

01 - Programmieren und trainieren Sie einen BPE-Tokenizer und untersuchen Sie die Relevanz der Sprachrepräsentation in den Trainingsdaten

Die Tokenisierung stellt für alle sprachbasierten KI-Modelle eine zentrale Grundlage dar. Die Tokenisierung selbst erfolgt dabei durch einen simplen Trainingsalgorithmus - dessen Parameter können aber großen Einfluss auf die Performance von KI-Modellen haben.

a) Schreiben Sie ein Skript bzw. kleines Programm, mit dessen Hilfe Sie einen simplen BPE-Tokenizer auf der Basis eines oder mehrerer Textsammlungen trainieren können und anschließend neue Texte in tokens unterteilen können.
b) Trainieren Sie ihren Tokenizer auf Textsammlungen von verschiedenem Umfang, verschiedenen Sprachen (Englisch, Deutsch sowie Deutsch und Englisch) und vergleichen Sie die Effizienz des Tokenizers (wie viele Tokens werden für einen bestimmten Text/eine bestimmte Information benötigt). Eine geeignete Quelle für Textsammlungen ist hier: https://opus.nlpl.eu/results/en&de/corpus-result-table, z.B. auch ein Datensatz aus dem Auswärtigen Amt: https://opus.nlpl.eu/ELRC-642-Federal_Foreign_Berl/en&de/v1/ELRC-642-Federal_Foreign_Berl
c) Analysieren Sie die Tokenizer, die in verschiedenen Sprachen trainiert wurden, anhand von Sätzen in den jeweiligen Sprachen. Welche Effekte fallen Ihnen dabei auf? Welche Tokenizer können eine möglichst gute Repräsentation der Informationen darstellen?

02 - Programmieren Sie einen Taschenrechner, der anstelle von Zahlen mit Worten rechnet. Untersuchen Sie den Einfluss verschiedener Embedding-Methoden

Die Umwandlung von Worten bzw. Tokens natürlicher Sprache in einen multi-dimensionalen Vektorraum stellt eine wichtige Voraussetzung für die Verarbeitung der Sprache mit Hilfe von KI dar.
In dieser Aufgabe programmieren Sie einen Taschenrechner, der anstelle von Zahlen mit Worten rechnet.

a) Schreiben Sie ein Skript bzw. kleines Programm, in das einfache Rechenaufgaben aus Worten eingegeben werden können, z.B. “king - man + woman” (hier wäre die erwartete Antwort “queen”). Finden Sie weitere angemessene “Rechnungen” für Worte, mit denen Sie ihr Skript überprüfen.
b) Vergleichen Sie Rechnungen von traditionellen Embedding-Ansätzen (bspw. mit Hilfe der GloVe Embeddings der Universität Stanford) mit Embedding-Modellen (unter Zuhilfenahme eines Wörterbuchs), die auf der Transformer-Technologie basieren. Welche Unterschiede erkennen Sie in den Ergebnissen, wie sind diese möglicherweise zu erklären? Welche Vor- bzw. Nachteile haben die verschiedenen Methoden?
c) Welche Rechenoperationen lassen sich im Rahmen eines solchen Rechners sinnvoll umsetzen? Welche Rechenoperationen ergeben für das “Rechnen” von Wörtern keinen Sinn?

03 - Extraktion der relevantesten Sätze aus einem langen Dokument mithilfe des TextRank Algorithmus und verschiedener Embedding-Methoden

Die Extraktion zentraler Informationen aus umfangreichen Texten spielt in vielen Anwendungen im Bereich NLP eine wesentliche Rolle. Mit dem TextRank-Algorithmus können Sie mittels graph basierter Methoden die Relevanz einzelner Sätze ermitteln und so eine prägnante Zusammenfassung erstellen.

a) Schreiben Sie ein Programm, das zunächst einen Text in seine einzelnen Sätze zerlegt und anschließend zwei alternative Repräsentationen der Sätze erzeugt: einmal mit TF-IDF-Vektoren und einmal unter Verwendung verschiedener Embedding-Modelle
b) Entwickeln Sie einen Graph, der die Ähnlichkeit zwischen Sätzen darstellt und verwenden Sie den TextRank Algorithmus zur Extraktion der relevantesten Säze.
c) Untersuchen Sie dabei, wie sich die verschiedenen Repräsentationen auf die Graphstruktur, die Auswahl der zentralen Sätze und letztlich auf die Informationsdichte der Zusammenfassung auswirken. Diskutieren Sie, welche Methode eine möglichst aussagekräftige und konsistente Repräsentation der wesentlichen Inhalte liefert.