Woche 12 Daten: Textrepräsentation

Skript

Erarbeitet von
Ann-Kathrin Selker

Lernziele

- Wörterbücher erklären können
- Worttaschenmodell am Beispiel erläutern und anwenden können
- word2vec- Idee und Nutzen erläutern können
- Idee des tf-idf-Maß erklären können

Inhalt

Wie erreiche ich eigentlich, dass mein Text von einem Machine-Learning-Modell verarbeitet werden kann?

Projekt „KI für alle"
HeiCAD – Heine Center for Artificial Intelligence and Data Science

Einstieg

In diesem Video betrachten wir Repräsentationen von Texten. Bei strukturierten Daten - zum Beispiel in Tabellen - ist jede Spalte ein Feature und der Eintrag für diese Spalte der Wert, den dieses Feature für das entsprechende Datenobjekt annimmt. Doch was sind die Features eines Textes, genannt Dokument? Intuitiv gesehen handelt es sich bei den Features eines Dokuments um dessen Sätze oder um dessen Wörter. In diesem Video beschränken wir uns auf die wortbasierte Repräsentation. Allerdings benötigen Machine-Learning-Algorithmen Eingaben in numerischer Form. Bei Daten in Tabellenform sind die Features entweder schon numerisch, oder können leicht in numerische Werte umgewandelt werden. Bei kategorischen Daten können z. B. die verschiedenen Kategorien durchnummeriert werden und der Kategoriename durch den entsprechenden Zahlenwert ersetzt werden. Damit Texte für das Machine-Learning-Modell lesbar und verarbeitbar sind, muss der Text ebenfalls numerisch repräsentiert werden. Das Umwandeln in numerische Form bezeichnet man als *feature extraction*.

Wörterbücher

Zuerst erstellen wir ein Wörterbuch, auch Vokabular genannt, bei dem jedes vorkommende Wort einen Eintrag im Wörterbuch darstellt. Da wir wie gewohnt nicht nur mit einem einzelnen Datenobjekt, hier Dokument, sondern mit einer Sammlung an Dokumenten (genannt Korpus) arbeiten, erstellen wir dieses Wörterbuch natürlich auch über alle Dokumente unseres Korpus. Neben den Wörtern des Korpus sollte das Wörterbuch auch einen Eintrag für unbekannte Wörter enthalten. Damit die Beispiele für dich als Mensch verständlicher sind, tue ich im Folgenden jetzt so, als ob bei unseren Dokumenten Vorverarbeitungsschritte wie z. B. Stoppwortentfernung und Stemming NICHT passiert sind. Angenommen, die einzigen beiden Dokumente unseres Korpus haben den Inhalt „KI für alle macht Spaß für alle!“ bzw. „KI macht immer Spaß.“. Das Wörterbuch sieht dann wie folgt aus:

(KI, für, alle, macht, Spaß, immer, UNBEKANNT)

Am Ende steht das Spezialwort UNBEKANNT. Ich habe die Wörter jetzt nach Erscheinen im Korpus zum Wörterbuch hinzugefügt, aber die Reihenfolge des Hinzufügens ist nicht wichtig. Für den weiteren Verlauf des Videos nehmen wir aber an, dass sich das Wörterbuch nicht mehr ändert.

Worttaschen-Modell

Eine Möglichkeit, Dokumente mithilfe eines Vektors zu repräsentieren, ist das Bag-of-Words-Modell, also das Worttaschen-Modell. Für ein gegebenes Dokument erstellen wir einen Vektor mit derselben Länge wie unser Wörterbuch und geben für jeden Eintrag an, wie häufig das entsprechende Wort in unserem Dokument vorkommt. Angenommen, wir betrachten das Dokument „KI macht mir Spaß, Spielen macht mir auch Spaß.“ Dann entsteht daraus diese Worttasche. Die Wörter *für*, *alle* und *immer* kommen null mal im Dokument

vor, *KI* einmal, *macht* und *Spaß* zweimal, und unbekannte Wörter wie z. B. *mir* kommen insgesamt viermal vor. Die Reihenfolge der Häufigkeiten entspricht dabei der Reihenfolge der Wörter im Wörterbuch. Wir haben also unser Dokument erfolgreich in numerischer Form angegeben und dafür einen Vektor in Länge des Wörterbuchs gebraucht.

(1, 0, 0, 2, 2, 0, 4)

Alternativ kann für jedes Dokument auch nur angegeben werden, ob ein Wort des Wörterbuchs enthalten ist oder nicht. Vektoren enthalten dann für jedes Wort entweder eine 0 für „nicht enthalten“ oder eine 1 für „enthalten“.

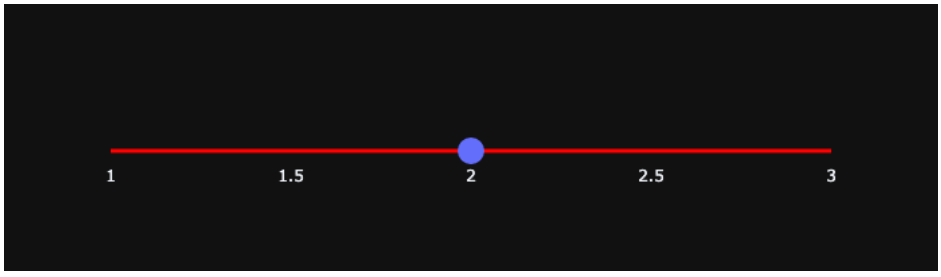
In der vorgestellten Version des Worttaschen-Modells haben wir mit absoluten Häufigkeiten von Wörtern gearbeitet. Diese sogenannte Vorkommenshäufigkeit, im englischen term frequency, kann auch als Maß verwendet werden, wie relevant ein Wort für ein Dokument ist. Je häufiger ein gewisses Wort in dem Dokument vorkommt, desto relevanter ist es für dieses Dokument. Die absoluten Vorkommenshäufigkeiten sind aber irreführend. Angenommen, ich möchte die Relevanz des Wortes *KI* für ein Dokument ermitteln. Im schon genannten Beispieldokument „KI für alle macht Spaß für alle“ kommt genau einmal *KI* vor, sodass das Wort *KI* als irrelevanter eingestuft wird als in einem Dokument, in dem bei 100 Millionen Wörtern zweimal das Wort *KI* vorkommt. Allerdings kann man hier natürlich gut argumentieren, dass im kürzeren Dokument KI ein Hauptfokus ist, während im extrem langen Dokument KI wohl nur am Rande erwähnt wird. Daher wird die Vorkommenshäufigkeit normalisiert angegeben. Normalisierung bedeutet hier, dass Werte auf eine gemeinsame Skala gebracht werden, um sie besser zu vergleichen. Normalisierungen kennst du aus dem Alltag. Beim Vergleichen von Lebensmittelpreisen hilft es nicht, nur die Gesamtpreise zu kennen, daher gibt es einen Preis pro 100 g, bei dem durch Gewicht geteilt wird. Und bei der Vorkommenshäufigkeit wird der Wert für ein Wort in einem Dokument normalisiert, indem durch die maximale Vorkommenshäufigkeit eines Wortes in demselben Dokument geteilt wird. In unserem Beispiel kommt im kurzen Dokument jedes Wort maximal zweimal vor, sodass sich die sogenannte relative Vorkommenshäufigkeit von *KI* auf 1/2, also 0.5 beläuft. Kommt im extrem langen Dokument das Wort *KI* zweimal und das häufigste Wort im Dokument 2000 vor, dann ergibt das eine relative Vorkommenshäufigkeit von 2/2000, also 0.001.
Allerdings erhalten bei den Vorkommenshäufigkeiten Wörter wie "und", "haben" und andere häufig benutzte Wörter automatisch höhere Werte als Wörter, die im Korpus selten auftreten. Die selteneren Wörter sind aber relevanter für die Dokumente, in denen sie vorkommen. Dies wird durch die sogenannte inverse Dokumenthäufigkeit gemessen, die die Relevanz eines Worts in Bezug auf den Korpus misst und dabei seltenere Wörter höher bewertet. Beide Konzepte werden im Tf-idf-Maß vereint, was für term frequency inverse document frequency steht. Dabei erhalten die Wörter einen hohen Wert, die im aktuell betrachteten Dokument häufig und im gesamten Korpus selten vorkommen. Die Idee des tf-idf-Maß kann im Worttaschen-Modell verwendet werden, indem jedem Wort anstelle der absoluten Häufigkeit der tf-idf-Wert zugeordnet wird, um so einen repräsentierenden Vektor für ein Dokument zu erzeugen.

Beim Worttaschen-Modell ist die Länge des Wörterbuchs auch gleichzeitig die Länge des Vektors, der das Dokument repräsentiert. Es ist also besonders wichtig, dass ein Wörterbuch

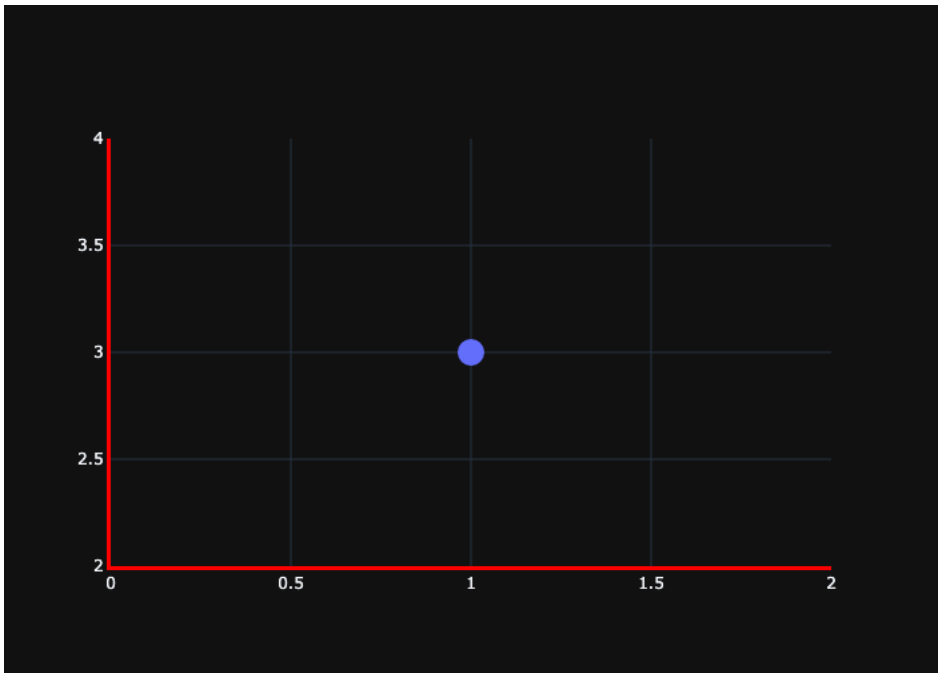
nicht übermäßig lang wird (Stichwort Fluch der Dimensionalität). Es bietet sich also nicht nur an, vorher Textvorverarbeitungsschritte wie das Entfernen von Stoppwörtern und Stemming durchzuführen, um die Länge des Wörterbuchs zu reduzieren, sondern auch, nicht jedes vorkommende Wort in das Wörterbuch mit aufzunehmen. Zum Beispiel kann mithilfe des tf-idf-Wertes die Größe des Wörterbuchs verringert werden, indem Worte mit höheren Werten bevorzugt aufgenommen werden. Aber selbst mit verkürzten Wörterbüchern handelt es sich bei den daraus entstehenden Dokumentvektoren noch um Vektoren, die zum größten Teil aus Nullen bestehen, da ein Wörterbuch hunderttausende oder sogar mehrere Millionen Einträge haben kann.
Das Worttaschen-Modell kann zum Beispiel bei der Spam-Erkennung angewendet werden. E-Mails werden daraufhin untersucht, ob sie gewisse Wörter enthalten, die häufig in Spam-E-Mails auftauchen, wie z. B. nigerianischer Prinz oder Bitcoin, und daraus eine Wahrscheinlichkeit berechnet, ob es sich bei der E-Mail um Spam handelt. Das ist auch der Grund, weshalb du bei Spam-E-Mails häufig seltsame Zeichen in den relevanten Wörtern findest, wie hier: Bítc0iɳ. Dadurch erhofft sich der Absender, dass der Spamfilter das Wort nicht als typisches Spamwort identifiziert.

Worteinbettungen

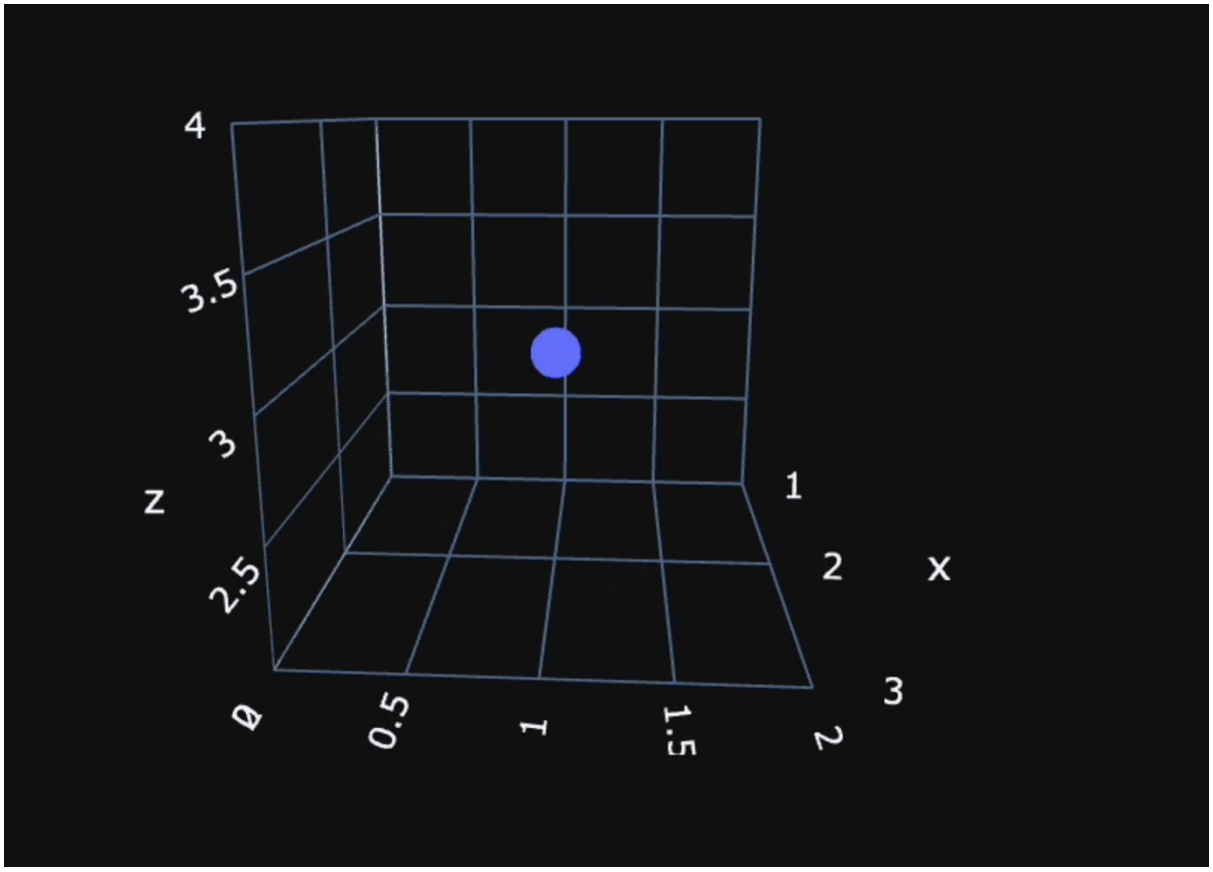
Der Nachteil an dem Worttaschen-Modell ist der Verlust von Kontext. So kann nicht erkannt werden, dass es sich bei den Dokumenten „KI macht Spaß“ und „KI bereitet Freude“ um zwei Dokumente mit derselben Aussage handelt, da sich die verwendeten Wörter unterscheiden. Um auch den Kontext von Wörtern zu erhalten, lässt sich zum Beispiel eine etwas kompliziertere Repräsentation eines Wörterbuchs erstellen. Wir berechnen für jedes Wort im Wörterbuch eine Einbettung (engl. Embedding), also einen Vektor, der den Standort dieses Worts im Raum angibt. Zur Erinnerung: Jeder Vektor beschreibt einen Punkt im Raum. Im eindimensionalen Raum beschreibt der Vektor (2) diesen Punkt,



im zweidimensionalen beschreibt zum Beispiel der Vektor (1,3) diesen Punkt,



und im dreidimensionalen beschreibt der Vektor (2,1,3) diesen Punkt.



Einblendung der jeweiligen Punkte

Sinn der Vektorwerte für die einzelnen Wörter ist es, dass Wörter, die Ähnliches bedeuten, sich auch im Vektorraum nah beieinander befinden. So sollen z. B. Kaffee und Tee nah beieinander stehen, da es sich bei beiden um Heißgetränke handelt.
Algorithmen basierend auf neuronalen Netzen wie word2vec erstellen nach Eingabe eines Korpus eine Vektorrepräsentation aller Wörter im erstellten Wörterbuch. Diese Vektoren haben häufig eine Länge im dreistelligen Bereich, d. h. der entsprechende Vektorraum hat mehrere hundert Dimensionen. Mithilfe der Einbettungen ist es sogar möglich, Analogien zu bilden wie „Welpe ist zu Hund wie Kätzchen zu Katze“. Dabei muss eine Maschine die genaue Bedeutung dieser Wörter gar nicht kennen, sondern sie kann die Zusammenhänge aufgrund der Abstände der Vektoren berechnen. Wie nah gewisse Wörter aneinander liegen, kommt allerdings stark auf den Korpus an, mit dem trainiert wird. Gibt es eine Voreingenommenheit in den Originaldaten, z. B. dass Ärzt*innen häufig männlich und Pfleger*innen häufig weiblich sind, dann spiegelt sich das auch in der Entfernung dieser zwei Berufe zu den Worten Mann und Frau wider.
Es ist natürlich sehr aufwendig, selbst ein Modell für das Erstellen von Einbettungen zu trainieren, nur um damit das Training für das eigentliche Ziel durchzuführen, für das die Textrepräsentation benötigt wird. Daher gibt es bereits viele vortrainierte Einbettungen, die unter gewissen Lizenzen für die eigene Arbeit verwendet werden können. Bitte achte hier auf die rechtlichen Rahmenbedingungen bei der Nutzung!

Abschluss

In diesem Video hast du gelernt, was ein Wörterbuch ist und wie es bei der Repräsentation von Dokumenten helfen kann. Du kannst jetzt sowohl das Worttaschen-Modell als auch Einbettungen anhand von Beispielen veranschaulichen.

Weiterführendes Material

https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

https://machinelearningmastery.com/what-are-word-embeddings/

https://machinelearningmastery.com/develop-word-embedding-model-predicting-movie-review-sentiment/

Disclaimer

Transkript zu dem Video „Woche 12 Daten: Textrepräsentation“, Ann-Kathrin Selker.