Rechercheaufgabe

July 28, 2020

Inhaltsverzeichnis

[1]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

1 Einführung

TODOS:

- Übersicht über alle Verfahren geben (Tabelle?!)
- einführende Worte
- überprüfen Formulierung

Das **Natural Language Processing** (kurz: NLP) befasst sich mit Methoden und Verfahren zur maschinellen Verarbeitung von natürlicher Sprache in Form von Worten, Texten oder ganzen Korpora. Bevor jedoch NLP Verfahren wie die Textklassifikation oder das Topic Modelling auf die Textdaten angewendet werden können, müssen diese in eine Darstellungsweise umgewandelt werden, mit der die Verfahren arbeiten können. Die rohen Textdaten werden daher in **Vektoren**, die aus Zahlen bestehen, umgewandelt. Dieser Vorgang nennt sich **Vektorisierung**. Ein Wort wie "Baum" kann dadurch als Vektor aufgefasst werden. Natürlich können auch andere Features aus den Texten als Vektoren dargestellt werden; so ist es auch möglich, einzelne Buchstaben, Phrasen, Sätze, Segmente oder ganze Texte als Features aus den Textdaten zu extrahieren und diese zu vektorisieren. In der folgenden Übersicht werden jedoch vorwiegend Wörter als Features verwendet.

1.1 Das Bag-of-Words Modell

Das wohl einfachste Verfahren zur Darstellung von Wörtern als Vektoren ist das **Bag-of-Words** Modell. Wörter werden hier als eindimensionale Vektoren (= einfache Zahl) dargestellt, wobei jedes individuelle Wort einen individuellen eindimensionalen Vektor (auch: **Index**) zugeordnet bekommt. Die Zuordnungen jedes einzigartigen Wortes zu seinem Vektor werden in einem *Vokabular* gespeichert. Nun können mithilfe dieses Vokabulars auch ganze Sätze oder sogar Texte dargestellt werden. Dafür wird für jeden Satz/Text ein Vektor gebildet, der die gleiche Länge wie das Vokabular hat. Jedem Eintrag des Vektors wird anhand des Vokabulars ein Wort zugeordnet. Der Satz/Text wird dann als Vektor aus **absoluten Termhäufigkeiten** dargestellt, wo an jeder Stelle, an dem ein Wort aus dem Vokabular in dem Text vorkommt, die Häufigkeit des Wortes in dem jeweiligen Satz/Text steht und an jeder anderen Stelle eine 0, da es kein einziges Mal vorkommt.Section **??** Dies soll im Folgenden anhand eines Code-Beispiels erläutert werden. Zuerst wird das Vokabular aller Texte dargestellt, bei dem die Wörter einem Index zugeordnet werden (es wird ab 0 gezählt). Danach werden die vektorisierten Sätze/Texte angezeigt.

1 Dies ist nur eine Möglichkeit, die Häufigkeit eines Wortes beim Bag-of-Words Modell darzustellen. Eine weitere Möglichkeit wären binäre Häufigkeiten, bei denen das Vorkommen eines Wortes mit einer 1 und die Abwesenheit eines Wortes mit einer 0 gekennzeichnet werden. Um häufigen Wörtern in den Dokumenten weniger Gewicht zu geben, da dies oft einen geringeren Informationsgehalt besitzen, ist es auch möglich, das Bag-of-Words Modell in der Kombination mit dem TF-IDF Maß aus dem Bereich des Information Retrievals zu verwenden, bei dem die Häufigkeit von Worten skaliert wird.

[20]:
```
text = ["ich gehe nicht mehr zur schule, ich gehe jetzt arbeiten",
        "in der schule habe ich viel gelernt",
        "zu hause habe ich nichts gelernt"]

vectorizer = CountVectorizer()
vector = vectorizer.fit_transform(text)
print(vectorizer.vocabulary_)
```

```
{'ich': 6, 'gehe': 2, 'nicht': 10, 'mehr': 9, 'zur': 15, 'schule': 12, 'jetzt':
8, 'arbeiten': 0, 'in': 7, 'der': 1, 'habe': 4, 'viel': 13, 'gelernt': 3, 'zu':
14, 'hause': 5, 'nichts': 11}
```

[21]:
```
print(vector.toarray())
```

```
[[1 0 2 0 0 0 2 0 1 1 1 0 1 0 0 1]
 [0 1 0 1 1 0 1 1 0 0 0 0 1 1 0 0]
 [0 0 0 1 1 1 1 0 0 0 0 1 0 0 1 0]]
```

1.2 Grenzen des Bag-of-Words Modells

Aufgrund seiner Einfachheit ist das Bag-of-Words Modell gut verständlich und sehr schnell umsetzbar. Es hat jedoch eine Reihe an Nachteilen, von deinen einige im Folgenden kurz erläutert werden:

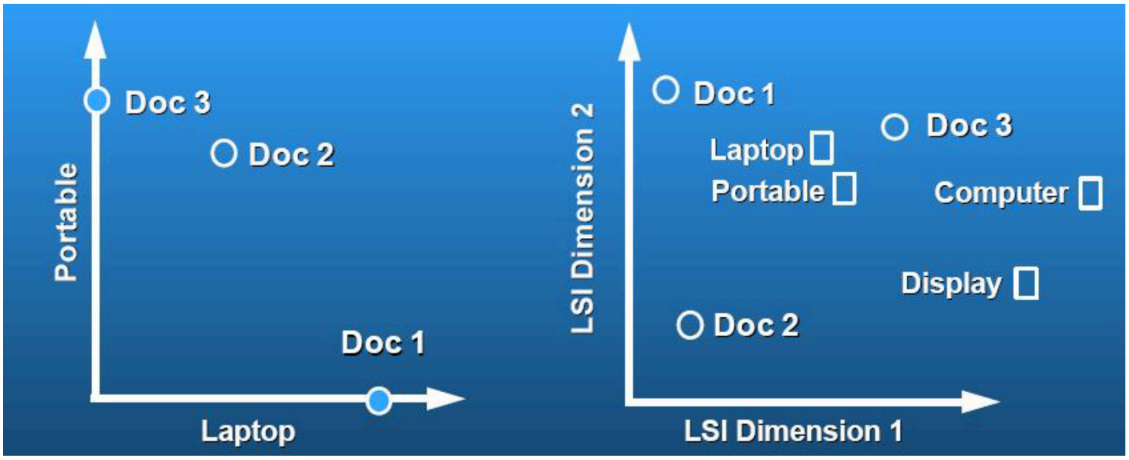
- **Keine Informationen über Reihenfolge der Wörter.** Beim Bag-of-Words Modell wird jegliche Information über die Reihenfolge der Wörter verworfen, der Kontext eines Wortes

bleibt unberücksichtigt. Dies wird auch durch den Namen dieses Modells deutlich: Die Bezeichnung "bag" (deutsch: Sack) soll darauf hinweisen, dass alle Informationen über die Struktur oder Reihenfolge der Wörter im Dokument verworfen werden, da sie metaphorisch in einen "Sack" geworfen werden. Die Reihenfolge lässt sich auch nicht im Nachhinein rekonstruieren. Insgesamt gehen somit sehr viele semantische Informationen verloren. Eine Lösung, bei der die Reihenfolge der Worte berücksichtigt werden kann, ist die Verwendung von **N-Grammen**.

- **Spärlichkeit von Wortvektoren.** Umso mehr verschiedene Worte in den verwendeten Texten vorkommen, umso größer wird das Vokabular. Dies kann oft zu sehr spärlichen (engl. *sparse*) Wortvektoren führen. Besteht das Vokabular aus 500000 Worten, ein Text aber nur aus 50 verschiedenen Worten, sind nur 0.01% der Stellen des 500000 langen Wortvektors mit Einsen besetzt, der Rest nur mit Nullen. Dies führt dazu, dass eine große Menge an Rechenspeicher für die Verarbeitung der riesigen Matrizen benötigt wird. Weiterhin werden wenige Informationen in sehr großen Repräsentationsräumen benutzt, wodurch es für einige NLP Verfahren und Modelle problematisch ist, diese wenigen Information effizient zu nutzen. Eine Lösung bieten dichtbesetzte **Word Embeddings**, die in den nächsten Kapiteln behandelt werden.
- **Abbildung der Mehrdeutigkeit von Worten**. Wörter können trotz gleicher Schreibweise mehrere Bedeutungen haben, welche sich durch den Kontext des Wortes zeigen können. Dies wird durch das Bag-of-Words Modell nicht abgebildet. Eine mögliche Lösung wäre die Verwendung von **kontextabhängigen Word Embeddings** wie die **BERT-Embeddings** in Kapitel 7 (TODO).

2 LSA

Latent Semantic Analysis (kurz: LSA, auch: *Latent Semantic Indexing*) ist ein Verfahren aus dem Bereich des Information Retrievals aus dem Jahre 1990. Bei diesem Verfahren werden Dokumente in einem latenten Raum abgebildet, der aus **Konzepten** (oder **Hauptkomponenten**) besteht. Dokumente, die ähnlich zueinander sind, d.h. aus ähnlichen Konzepten bestehen, werden in diesem Raum näher beieinander platziert. Das Ziel der LSA ist es, diese Konzepte innerhalb der Dokumente zu finden. Dies wird durch die folgende Grafik deutlich, bei der Dokumente mit ähnlichen Konzepten näher beieinander platziert werden.



Grafik von Susan Dumais, siehe Präsentation.

Beim Latent Semantic Indexing wird eine Term-Dokument Matrix in einen latenten Raum überführt, in dem Terme und Dokumente, die einander ähnlich sind, näher beieinander platziert werden. Anfragen werden ebenfalls in diesen Raum transformiert, so dass die Ähnlichkeit zwischen Dokumenten und Anfragen im latenten Raum ermittelt werden können.

LSI greift auf eine Technik der linearen Algebra zu, der Singulärwertzerlegung (Singular Value Decomposition). Diese zerlegt die originale Matrix in drei Matrizen. Die beiden äußeren Matrizen bestehen aus den linken, bzw. rechten orthonormalen Eigenvektoren. Die mittlere Matrix ist eine Diagonalmatrix, die die singulären Werte der Originalmatrix enthält.

Mit Hilfe dieser Zerlegung kann eine Approximation der Originalmatrix mit einer kleiner dimensionierten Matrix von Rang r erreicht werden. Die singulären Werte in der Diagonalmatrix sind nach ihrer Größe absteigend geordnet. Singulärwerte, die unter einem bestimmten Schwellenwert liegen, werden entfernt. Auch in den anderen Matrizen werden entsprechende Zeilen oder Spalten entfernt. Mit Hilfe der reduzierten Matrizen erhält man durch Matrixmultiplikation die optimale Approximation der Originalmatrix. Optimal ist diese Approximation deswegen, weil es keine andere Matrix von Rang k gibt, die ähnlicher zur Originalmatrix ist. Der Fehler oder Unterschied zwischen den Matrizen wird dabei mit der Technik der kleinsten Quadrate bestimmt (Frobenius Norm).

[ ]:

[ ]:

3 Word Embeddings

TODO: Einführung geben, folgendes ist von textclf tutorial

Word Embeddings (deutsch: Worteinbettungen) sind die Sammelbezeichnung für eine Reihe von Sprachmodellierungstechniken. Sie bieten eine andere Form der Wortrepräsentation als das Bag-of-Words-Modell, bei der Wörter mit einer ähnlichen Bedeutung ähnlich dargestellt werden, wobei der Kontext der Wörter berücksichtigt wird. Word Embeddings repräsentieren Wörter als Vektoren in einem multidimensionalen semantischen Raum. In diesem Raum werden Wörter, die ähnlich zueinander sind, näher beieinander platziert. Diesen Raum kann man sich etwa folgendermaßen vorstellen:

TODO: ausfüllen

LSA

word2vec

Glove

FastText

ELMo

BERT

Entstehungsjahr

4 word2vec

TODO

5 Glove

TODO

6 FastText

TODO

7 ELMo

TODO

8 BERT

TODO

[ ]: