KI für Alle 2: Verstehen, Bewerten, Reflektieren

Themenblock Clustering: vom Sortieren bis zum Explorieren:
06_04Modellauswahl_TopicModeling

Topic Modeling

Erarbeitet von
Dr. Katarina Boland

Lernziele

- Du kannst erklären, was Topic Modeling ist
- Du kannst Verfahren nennen, mit denen Texte thematisch geclustert werden können
- Du kannst nachvollziehen, wie semantische Embeddings für Clustering verwendet werden können
- Du kannst die Funktionsweise des BERTopic Verfahrens erklären

Projekt ai4all „KI für Alle"
HeiCAD –Heine Center for Artificial Intelligence and Data Science

Inhalt

Einstieg

Topic Modeling ist ein seit über 30 Jahren bestehendes Forschungsfeld. Ziel ist die Detektion von semantischen Konzepten, oder einfacher: Themen, in Texten. Input ist eine Sammlung von Dokumenten, aus denen die Topic Modeling Verfahren unüberwacht Themen extrahieren sowie repräsentative Wörter zur Beschreibung von Dokumenten, die diese Themen adressieren.

Topic Models sind geeignet, große Mengen von Textdokumenten zu explorieren, zusammenzufassen und zu organisieren.

Zu den bekanntesten Verfahren gehört LDA, Latent Dirichlet Allocation, das zur Gruppe der probabilistischen Topic Models gehört.

Quelle [1]

Wir wollen uns hier auf eine andere, neuere Art von Topic Models konzentrieren: die der neuronalen Topic Models. Stellvertretend betrachten wir dazu das BERTopic Framework.

Quelle [2]

BERTopic geht im Wesentlichen in vier Schritten vor:

1. Zuerst werden Dokumente durch semantische Embeddings repräsentiert. Zwischen Embeddings können Ähnlichkeiten berechnet werden, was den Einsatz von Clusteringverfahren ermöglicht.
2. In einem zweiten Schritt findet eine Dimensionsreduktion statt, um die Effizienz zu steigern und irrelevante Merkmale zu entfernen.
3. Der dritte Schritt ist das Clustering an sich, bei dem semantisch ähnliche Dokumente gruppiert werden.
4. In einem letzten Schritt werden repräsentative Wörter auf Grundlage der Cluster und der enthaltenen Dokumente generiert.

Wir betrachten diese Schritte nun im Detail.

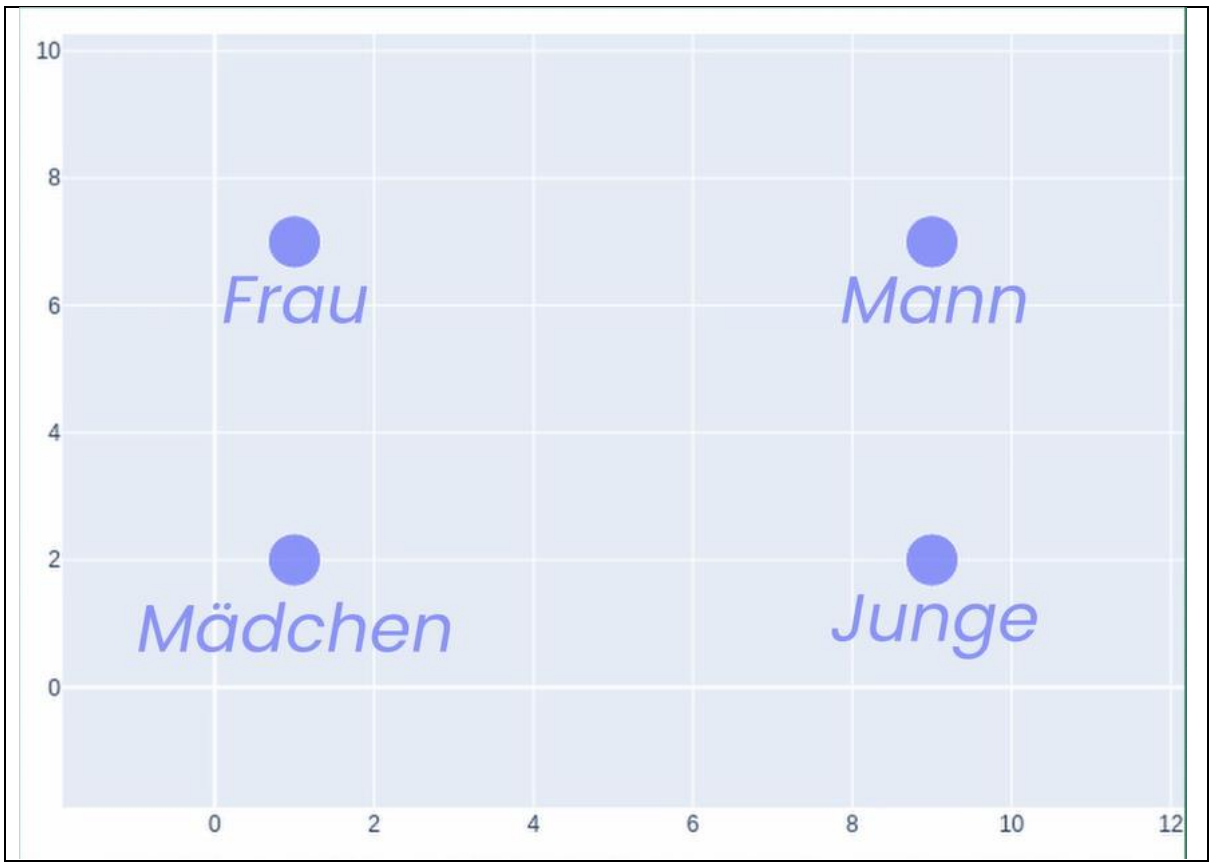
Generierung von Semantischen Embeddings

Der folgende Exkurs ist eine Adaption des Embedding-Tutorials von Mitarbeitenden der Carnegie Mellon University.

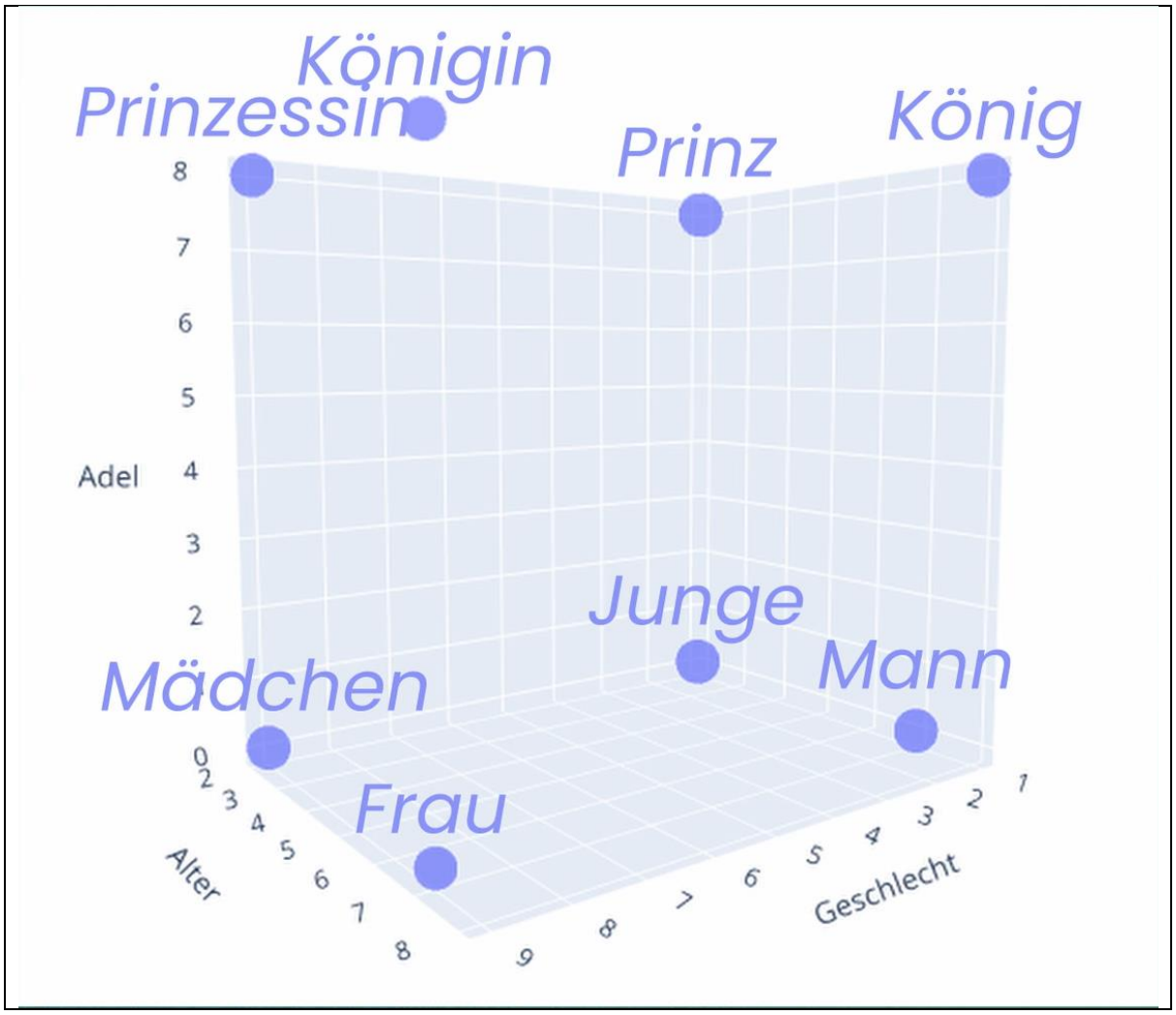
Quelle [3]

Zur Erinnerung: Embeddings (zu Deutsch: Worteinbettungen) sind Repräsentationen von Wörtern, Sätzen oder Dokumenten im sogenannten Semantic Feature Space (zu Deutsch etwa: semantischer Merkmalsraum).

Wir haben also eine Anzahl von semantischen Merkmalen, nehmen wir als Beispiel Alter und Geschlecht. Diese können wir im Merkmalsraum darstellen. Wenn wir zwei Merkmale haben, reichen dafür zwei Dimensionen. Wir könnten also das Alter auf der y-Achse und das Geschlecht auf der x- Achse anordnen. Wenn wir nun Wörter haben wie „Mädchen", „Frau", „Junge" und „Mann", so können wir sie gemäß ihrer Merkmale im Raum abbilden.



Angenommen, wir möchten nun die Wörter „König", „Königin", „Prinzessin" und „Prinz" in diesem Raum abbilden. Die zwei Dimensionen reichen nun nicht mehr aus, denn es gäbe keine Möglichkeit, den Unterschied zwischen „Mann" und „König", zwischen „Königin" und „Prinzessin" usw. darzustellen. Wir fügen also eine neue Dimension hinzu: die Dimension „Adel".



Wir können durch diese Repräsentation nicht nur Relationen erkennen, wie z. B., dass Frau und Mädchen auf dieselbe Weise mit einander in Beziehung stehen wie Mann und Junge, wir können nun auch die Wörter durch ihre Koordinaten im Merkmalsraum repräsentieren.

Wir haben also gerade Wörter durch 3-dimensionale Vektoren dargestellt.

Durch diese Darstellung lassen sich Relationen zwischen Wörtern mathematisch berechnen. Wenn wir z. B. die Analogie auflösen wollen „König verhält sich zu Mann wie Frau zu ____" können wir den Vektor des Worts Mann vom Vektor des Worts König subtrahieren. Dies gibt uns sozusagen den Weg, den wir von „König" gehen müssen um zu „Mann" zu gelangen. Diesen Weg gehen wir dann vom Wort „Frau", das heißt, wir addieren das Ergebnis der Subtraktion zum Vektor von „Frau". Das nächstgelegene Konzept zu dieser Koordinate ist „Königin", also ist Königin die gesuchte Antwort.

Vielleicht fragst du dich, wieso die Koordinate nur nahe an der Königin und nicht genau an ihren Koordinaten liegt. Das liegt daran, dass wir „König" ein etwas höheres Alter zugewiesen hatten als „Königin", wie du in der Tabelle sehen kannst.

Solche Effekte mögen auf den ersten Blick unlogisch erscheinen, sind aber erwartbar, wenn wir die Zuordnungen von Wörtern und Merkmalen von Beobachtungen, zum Beispiel aus biografischen Texten, lernen, und die Merkmale nicht gleich verteilt sind.

Offensichtlich kommen wir mit drei Dimensionen schnell an die Grenzen dessen, was wir an Wörtern im semantischen Merkmalsraum darstellen können. Wir können uns mehr als drei Dimensionen räumlich zwar nicht vorstellen, trotzdem können wir den Merkmalsraum um beliebig viele Dimensionen erweitern, um immer mehr Wörter durch Vektoren darstellen zu können. In der Praxis werden meist mehrere hunderte oder tausende Merkmale berücksichtigt, um Wörter zu repräsentieren.

Analog zur Darstellung einzelner Wörter können auch gesamte Sätze oder Dokumente durch Vektoren repräsentiert werden.

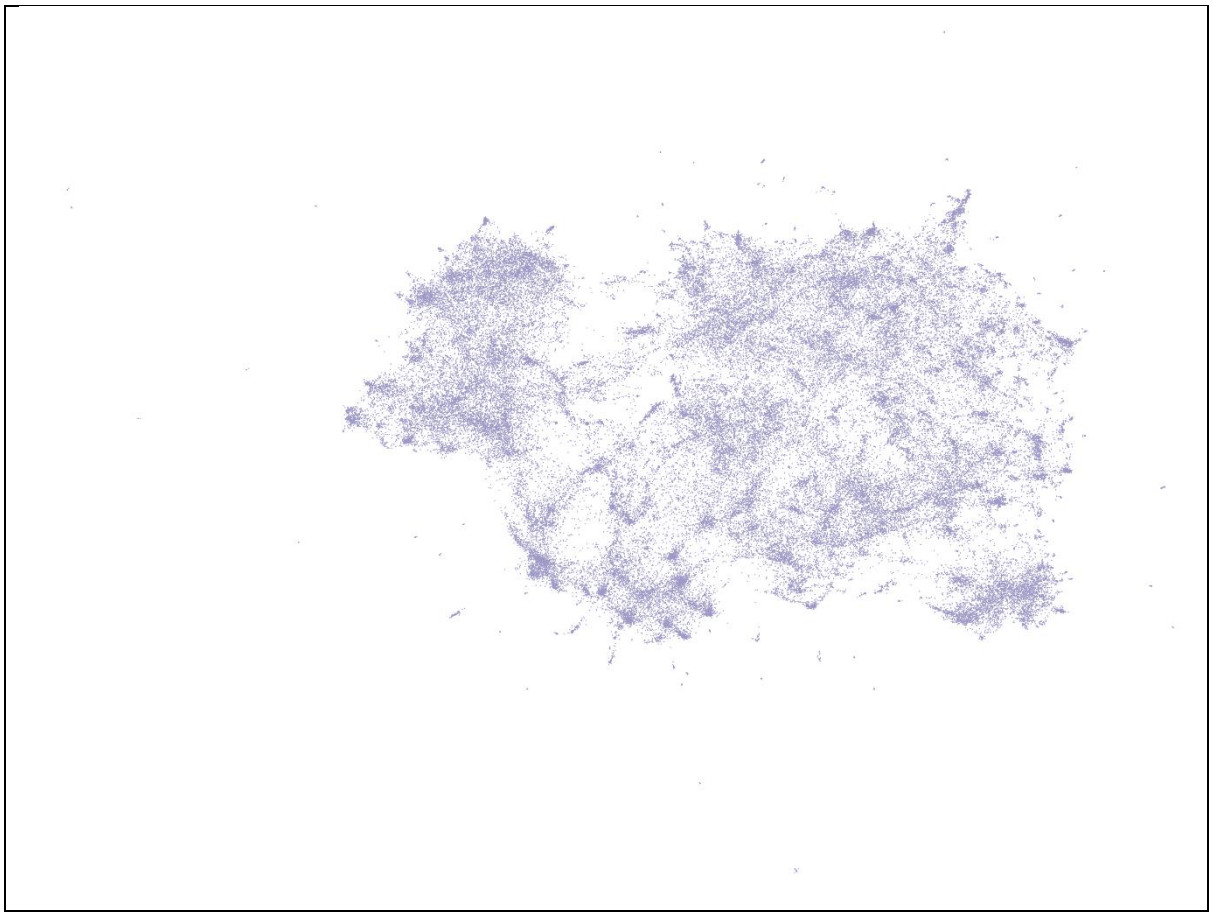
Genauso, wie wir die semantischen Ähnlichkeiten von Wörtern berechnen können, können wir nun mithilfe der Dokumentembeddings die Ähnlichkeiten von Dokumenten errechnen und sie nach ihrer Ähnlichkeit clustern.

Dimensionsreduktion

Da jedes Dokument durch eine riesige Anzahl von Merkmalen und hochdimensionalen Vektoren dargestellt wird, werden Berechnungen schnell komplex und ggf. verwässert die hohe Anzahl an Merkmalen den Blick auf das Wesentliche, wir haben also mit dem „Fluch der Dimensionalität" zu kämpfen. Daher empfiehlt es sich, eine Dimensionsreduktion durchzuführen. Hierfür kann beispielsweise die Methode UMAP eingesetzt werden.

Quelle [4]

Der Grundgedanke ist, dass die Datenpunkte derart in einen Raum mit weniger Dimensionen transformiert werden, dass die Abstände zwischen ihnen möglichst gut erhalten bleiben.



Dokumentvektoren in 2D. Quelle: [5]

Quelle [5]

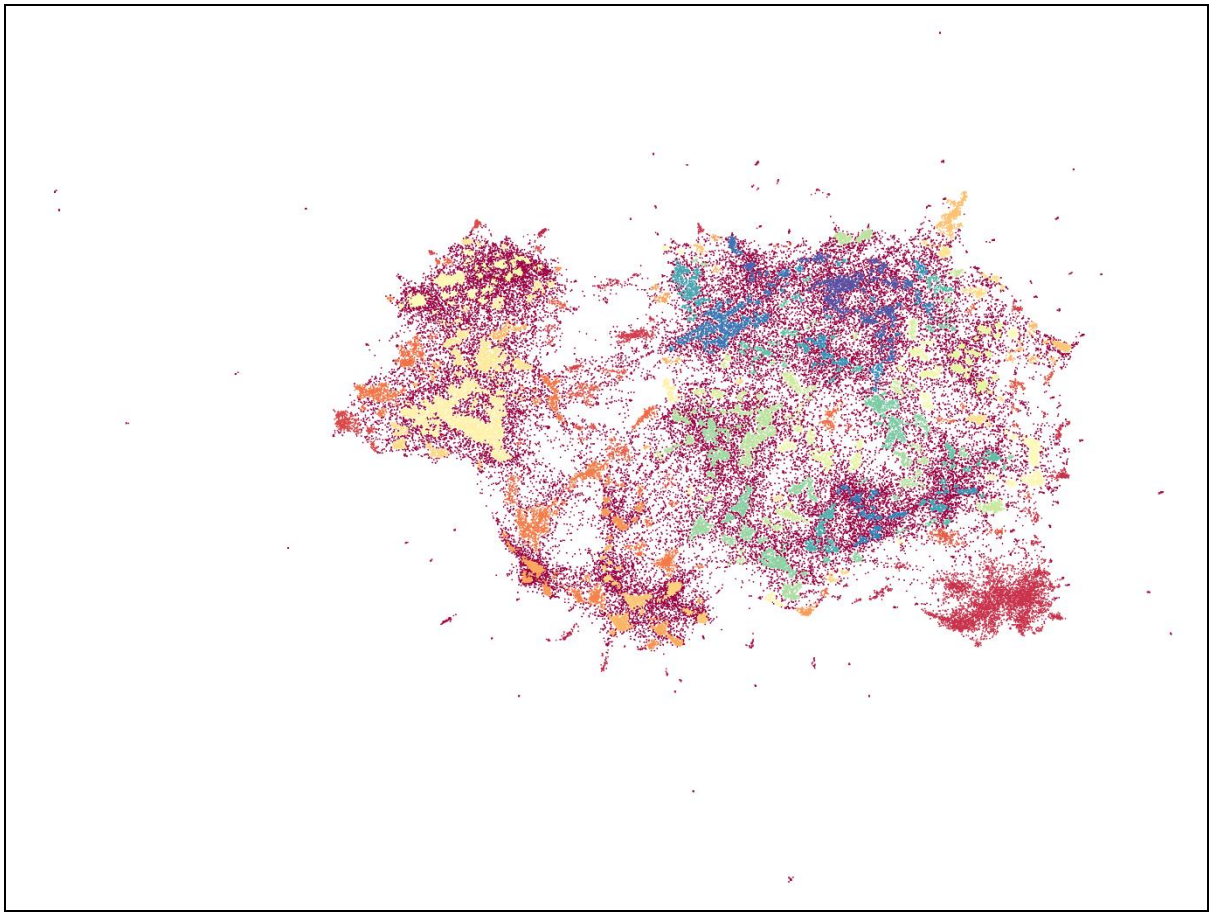
Clustering

Das Clustering der Vektoren kann mit verschiedenen Clusteringverfahren durchgeführt werden, z. B. mit k-Means oder dem hierarchischen HDBScan. Wir verwenden hier HDBScan.

Quelle [6]
Quelle [7]

Dieses Clusteringverfahren identifiziert jene Bereiche, die eine hohe Dichte aufweisen, also in denen Dokumente bezüglich ihrer semantischen Ähnlichkeit nahe beieinander liegen. Nahe beieinander liegende Dokumente werden zu einem Cluster vereinigt.

Diese sind die Themencluster, im Bild bunt eingefärbt. Einige Bereiche weisen keine hohe Dichte auf, d. h. viele Dokumente gehören zu keinem erkennbaren Cluster, sondern stehen eher für sich allein und sind zu weit von anderen Dokumenten entfernt, als dass sie vereinigt werden könnten, im Bild mit roter Farbe dargestellt. Diese Punkte werden keinem Cluster zugewiesen, sondern werden als Rauschen betrachtet und ignoriert. HDBScan benötigt keinen expliziten Parameter k für die Anzahl an zu bildenden Clustern.

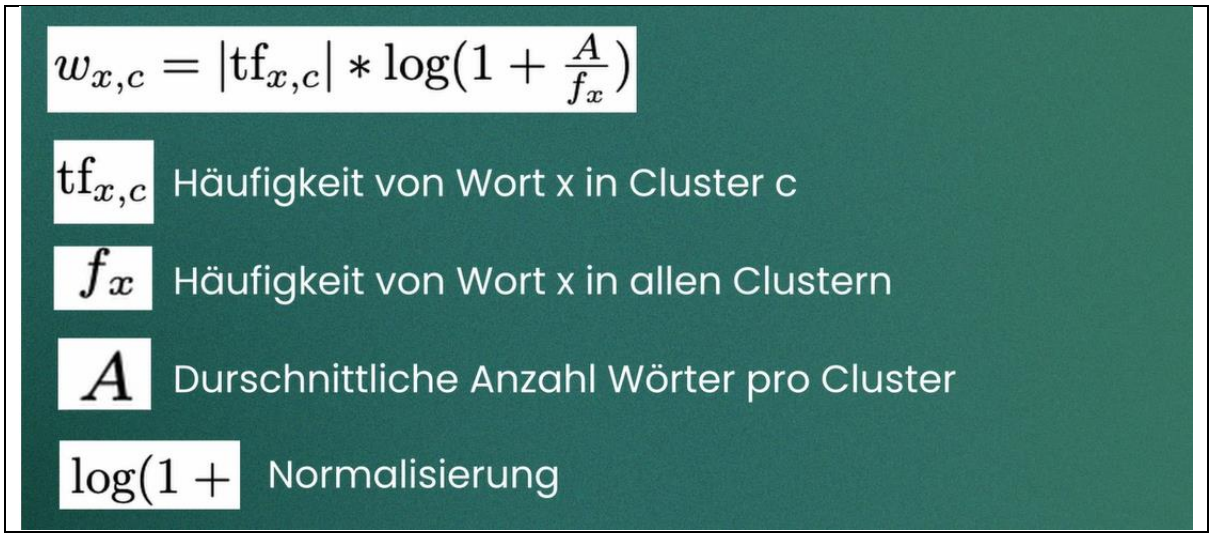


Themencluster. Quelle: [5]

Quelle [5]

Extraktion von repräsentativen Wörtern

Sind die Dokumente geclustert, ist es nun noch hilfreich, repräsentative Wörter für die thematischen Cluster zu finden um beurteilen zu können, welches Thema jeweils durch die Cluster repräsentiert wird. Für BERTopic wird hier c-TF-IDF, eine Abwandlung von TF-IDF, auf die tokenisierten Dokumente innerhalb eines Clusters angewendet.



c-TF-IDF steht für class-based Term Frequency-Inverse Document Frequency. Bei TF-IDF wird die Relevanz eines Terms für ein Dokument, also seine Vorkommenshäufigkeit, relativ zur Relevanz bzw. Vorkommenshäufigkeit des Terms in allen Dokumenten gemessen. Der Vergleich mit der Häufigkeit in anderen Dokumenten ist nötig, weil sonst wenig spezifische Wörter, wie beispielsweise Artikel, für alle Dokumente die höchsten Werte hätten. Wir interessieren uns aber für Wörter, die in einem gegebenen Dokument auffallend häufig sind.

Bei der c-TF-IDF Variante wird die Relevanz eines Terms für ein Cluster relativ zur Relevanz für alle Cluster gemessen. Alle Dokumente innerhalb eines Clusters werden hierfür konkateniert und die Häufigkeit des Terms wird pro Cluster gemessen und mit der relativen Häufigkeit des Terms über alle Klassen normalisiert. Wir erhalten so für jedes Themencluster eine Liste von repräsentativen Begriffen, die die Semantik des Clusters, also das Thema, gut beschreiben sollte.

Abschluss

Wir haben BERTopic als Stellvertreter für neuronale Topic Models kennengelernt. Mit diesem werden Dokumente basierend auf ihrer semantischen Ähnlichkeit geclustert und so eine Menge von Themenclustern erzeugt. Anschließend werden aus den Dokumenten innerhalb der Cluster repräsentative Begriffe ausgewählt. Wir erhalten so Themencluster

mit Dokumenten, die jeweils zu einem Thema gehören, und eine Liste von repräsentativen Begriffen. Beides zusammen bezeichnen wir als „Topic".

Quellen

Quelle [1] Abdelrazek, A., Eid, Y., Gawish, E., Medhat, W., & Hassan, A. (2023). Topic modeling algorithms and applications: A survey. Information Systems, 112, 102131. https://doi.org/10.1016/j.is.2022.102131

Quelle [2] Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure (arXiv:2203.05794). arXiv. https://doi.org/10.48550/arXiv.2203.05794

Quelle [3] Jason Xu, Saptarashmi Bandyopadhyay, Neel Pawar, and David S. Touretzky. (o. J.). Word Embedding Demo: Tutorial. Abgerufen 27. August 2024, von https://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

Quelle [4] McInnes, L., Healy, J., & Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction (arXiv:1802.03426). arXiv. https://doi.org/10.48550/arXiv.1802.03426

Quelle [5] Angelov, D. (2020). Top2Vec: Distributed Representations of Topics (arXiv:2008.09470). arXiv. https://doi.org/10.48550/arXiv.2008.09470

Quelle [6] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100–108. https://doi.org/10.2307/2346830

Quelle [7] McInnes, L., Healy, J., & Astels, S. (2017). hdbscan: Hierarchical density based clustering. Journal of Open Source Software, 2(11), 205. https://doi.org/10.21105/joss.00205

Disclaimer