# Technischer Bericht
# Tech4Germany

16. Oktober 2019

**Zusammenfassung**

Zusammenfassung ...

# 1 Einleitung

# 2 Datenbestand

# 3 Orientierung

Die gezeige Anwendung beinhaltet verschiedene Aspekte zur Orientierung von Nutzer*innen. Insbesondere ist diese Funktionalität für unsere ermittelten Personas der ambitionierten Aufsteigerin und des unsicheren Umsteigers hilfreich. Da allerdings diese Personas verschiedene Ziele verfolgen, haben wir dies auch in der Anwendung berücksichtigt. So kann sich die ambitionierte Aufsteigerin nach Angabe ihres aktuellen Tätigkeitsfelds innerhalb eines Raumes Kurse, die nahe an ihrer derzeitigen Position liegen, explorieren. Der unsichere Umsteiger dagegen kann ohne Vorgabe der aktuellen Tätigkeit, sondern ausschließlich basierend auf einer Vorauswahl an Branchen, verschiedene Berufe explorieren. Diese Auftrennung ist sinnvoll, da sich die Aufsteigerin innerhalb eines gegebenen Feldes weiterbilden möchte, wobei der Umsteiger verschiedene Bereiche erkunden möchte. Diese Erkenntnisse stammen aus unseren Interviews mit Nutzer*innen. Im Folgenden wird die räumliche Exploration des Umsteigers eingegangen, wobei zunächst ein Ähnlichkeitsmaß für Berufe definiert werden muss.

## 3.1 Ähnlichkeit von Berufen

Die Bundesagentur für Arbeit stellt in ihrem Portal BERUFENET Definitionen von Berufen zur Verfügung. Um eine Ähnlichkeit zwischen diesen zu ermitteln, muss die textuelle Form in einen Merkmals-Vektor überführt werden, der genauere Informationen zu den einzelnen Berufen liefert. Zwar könnte man die Levenshtein-Distanz[1] verwenden, allerdings verwendet diese keinerlei Informationen über die eigentlichen Berufsinhalte.

Anhand der Berufs-Bezeichnungen können durch vortrainierte Modelle wie bspw. Word2Vec[2] oder GloVe[3] bereits Merkmals-Vektoren erzeugt werden. Diese Modelle erstellen Merkmals-Vektoren anhand von Attributen, die anhand von verschiedenen Korpussen wie bspw. den Wikipedia-Daten gelernt wurden. Auch wir haben diese Verfahren verwendet um Merkmale für Berufe zu generieren, allerdings verfügen diese über keine kontextuellen Informationen zu einem Beruf, d.h. die Tätigkeiten sowie verschiedene Berufsbeschreibungen bleiben unberücksichtigt. Ein anderer Nachteil dieser Verfahrer ist, dass ein Merkmals-Vektor stets nur für

[1] vgl. http://www.levenshtein.de
[2] vgl. https://code.google.com/archive/p/word2vec/
[3] vgl. https://nlp.stanford.edu/projects/glove/

ein einzelnes Wort generiert werden kann. Auch wenn die Vektoren über verschiedene Wörter gemittelt werden können, gehen sehr viele Informationen verloren.

Um zusätzliche Informationen zu generieren, haben wir deshalb die Tätigkeitsbeschreibungen analysiert. Dabei lässt sich mit einem Bag-of-Words Ansatz ein Korpus generieren, der alle relevanten Wörter durch Verwendung eines Stemmers in ihrer Rohform beinhaltet. Zu Beachten ist dabei, dass häufige Wörter rausgefiltert werden müssen, um die Relevanz der einzelnen Merkmale nicht zu gefährden. So müssen Bindewörter und Pronomen extrahiert werden, wofür sich bspw. die Python-Bibliothek NLTK[4] sehr gut eignet. Nach der Erstellung des Korpuses können nun die Merkmals-Vektoren der einzelnen Berufe berechnet werden. Dazu kann ein TfidfVectorizer bspw. von Scikit Learn[5] verwendet werden, der die relative Häufigkeit der gestemmten Wörter innerhalb einer Tätigkeitsbeschreibung betrachtet.

So können Merkmals-Vektoren für Berufts-Tätigkeiten bestimmt werden, die sehr hochdimensional sind. Da auch die meisten Tätigkeitsbeschreibungen nur einen Bruchteil des Korpuses abdecken, eignet sich die euklidische Distanz ($d_e(p,q) = \sqrt{\sum_{i=1}^{n}(p-q)^2}$) nicht, um Ähnlichkeiten (bzw. Distanzen) zwischen den Berufen zu berechnen. Stattdessen eignet sich die Kosinus-Distanz ($d_c(p,q) = 1 - cos(\theta) = \frac{p \cdot q}{||p|| ||q||}$), da dieser statt dem Pfad zwischen $p$ und $q$ den räumlichen Winkel zwischen diesen betrachtet. Um diese Merkmals-Repräsentation greifbar zu machen, eignen sich verschiedene Algorithmen, die die Dimensionalität des Merkmals-Raums reduzieren. In unserer beispielhaften Implementierung haben wir T-distributed Stochastic Neighbor Embedding[6] verwendet, da dieses Verfahren eine ansprechende Visualisierung als bspw. eine PCA oder eine LDA liefern.

Damit die Distanzen nicht zur Laufzeit bestimmt werden müssen, exportieren wir eine Distanzmatrix, die die paarweise Distanz zweier Embeddings speichert. So kann diese Matrix beim Starten der Anwendung geladen werden und so direkt auf 'dist_matrix[i][:]' zugegriffen werden, um die paarweisen Distanzen des Embeddings $i$ zu den anderen Embeddings zu bestimmen. Mit diesen Distanzen kann nun innerhalb des hochdimensionalen Raumes navigiert werden.

## 3.2 Explorative Navigation durch den Berufsraum

Der Berufsraum ist sehr hochdimensional, d.h. eine Navigation durch diesen ist sehr komplex. Beginnend mit ausgewählten Branchen können hinterlegte Berufe, die exemplarisch die gegebenen Branchen repräsentieren, geladen werden. Über diese wird dann der Durchschnitts-Vektor berechnet, um einen Start-Punkt im Raum zu generieren. Für diesen Start-Punkt können nun die Nachbarn berechnet werden, indem die Kosinus-Distanz zwischen dem gemittelten Vektor und den Embeddings der einzelnen Berufe berechnet wird. Um eine performente Anwendung zu generieren, ist die Berechnung der Distanz nicht zwischen allen Paaren zu empfehlen, weshalb durch die Auswahl jedes bspw. fünften Berufes eine zufällige Komponente in die Auswahl der Optionsvorschläge integriert wird und gleichzeitig die Ladezeit der Anwendung stark reduziert wird. Die resultierenden Berufsvorschläge innerhalb der Optionen sollten erneut stark zufällig gewählt werden, um eine flexible Exploration zu ermöglichen. Die vorgeschlagenen Top 5 Berufe dagegenen sollten als nächste Nachbarn des aktuellen Punktes im Berufsraum generiert werden, um auch wirklich die naheliegendsten Berufe abzudecken.

[4] vgl. https://www.nltk.org
[5] vgl.
[6] vgl. https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html