



Universität Hamburg  
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT  
FÜR MATHEMATIK, INFORMATIK  
UND NATURWISSENSCHAFTEN

## Bachelorarbeit

# Semi-Automatische Erweiterung eines Synonymwörterbuchs zur Query Expansion für das Hamburger Transparenzportal

**Malte Johannsen**

---

6johanns@informatik.uni-hamburg.de

Studiengang Informatik

Matr.-Nr. 6946244

Fachsemester 6

Erstgutachter: Prof. Dr. Chris Biemann

Zweitgutachter: Dr. Lothar Hotz

Abgabe: 10.2019

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Fragestellung . . . . .	4
1.3	Aufbau der Arbeit . . . . .	4
1.4	Verwandte Arbeiten . . . . .	4
<b>2</b>	<b>Grundlagen</b>	<b>7</b>
2.1	Wörterbuch für Query Expansion . . . . .	7
2.1.1	WordNet . . . . .	7
2.1.2	GermaNet . . . . .	8
2.1.3	Grenzen des Wörterbuchansatzes . . . . .	8
2.2	Word Embedding . . . . .	8
2.2.1	CBOW . . . . .	9
2.2.2	SkipGram . . . . .	10
2.2.3	Aktivierungsfunktion-Optimierung . . . . .	11
2.2.4	Doc2Vec . . . . .	13
2.3	JoBimText . . . . .	14
2.4	Usability . . . . .	16
2.5	Evaluierungs-Methoden . . . . .	18
2.5.1	System Usability Scale . . . . .	18
2.5.2	Fleiss-Kappa . . . . .	19
2.5.3	Box-Whisker-Diagramm . . . . .	20
<b>3</b>	<b>Aufbau des Wortexpansionssystems</b>	<b>23</b>
3.1	Daten und Dokumente des Hamburger Transparenzportals . . . . .	23
3.2	Solr als Erweiterungswortsammlung . . . . .	24
3.3	Generierung von Erweiterungswörtern . . . . .	27
3.3.1	Word Embeddings . . . . .	27
3.3.2	GermaNet . . . . .	29
3.3.3	JoBimText . . . . .	30
3.3.4	Kombiniertes Modell . . . . .	30
3.3.5	Filterung . . . . .	32
3.4	Benutzerschnittstelle für Erweiterungswortevaluation . . . . .	33
3.4.1	Aufbau der Benutzeroberfläche . . . . .	33

---

3.4.2	Diskussion der Benutzeroberfläche . . . . .	36
3.5	Integrieren der Applikation in ein Produktionsumfeld . . . . .	38
<b>4</b>	<b>Aufbau des Experimentes</b>	<b>41</b>
<b>5</b>	<b>Ergebnisse</b>	<b>45</b>
5.1	Oberflächenevaluation . . . . .	45
5.1.1	Pilotstudie . . . . .	45
5.1.2	Finale Evaluation . . . . .	45
5.2	Modellevaluation . . . . .	49
5.2.1	Modellevaluation durch Testpersonen . . . . .	49
5.2.2	Modellevaluation durch Doc2Vec . . . . .	53
<b>6</b>	<b>Fazit und Ausblick</b>	<b>57</b>
	<b>Literaturverzeichnis</b>	<b>59</b>
	<b>Eidesstattliche Versicherung</b>	<b>63</b>

---

Aus Gründen der leichteren Lesbarkeit wird in der vorliegenden Bachelorarbeit die männliche Sprachform bei personenbezogenen Substantiven und Pronomen verwendet. Dies impliziert jedoch keine Benachteiligung eines anderen Geschlechts, sondern soll im Sinne der sprachlichen Vereinfachung, als geschlechtsneutral zu verstehen sein.

## **Abstract**

Das Ziel dieser Bachelorarbeit war es, ein semi-automatisches Erweiterungssystem für das Hamburger Transparenzportal zu entwickeln, mit dem die lexikalische Ressource, welche zur Query Expansion verwendet wird, erweitert werden kann. Um Erweiterungswörter zu finden, welche den gefundenen Dokumenten weitere relevante Dokumente hinzufügen, wurden unterschiedliche Verfahren angewendet. Als statische Ressource kam GermaNet zum Einsatz, indem die definierten Synonyme, als Erweiterungswörter verwendet wurden. Zudem wurden verschiedene Word2Vec-Modelle auf den Dokumenten trainiert, um semantische Ähnlichkeiten zwischen Wörtern zu lernen. So ließen sich semantisch ähnliche Wörter, zu dem Suchwort ermitteln. Zuletzt wurde mit JoBimText eine distributionellen Semantik aufgebaut, aus der sich ebenfalls Erweiterungswörter ableiten ließen. Die verschiedenen Ansätze wurden mit dem Rank-Level-Fusion-Verfahren kombiniert und verwendet um Testpersonen, die besten Erweiterungswörter mit einer Benutzer-oberfläche zu präsentieren. Diese Benutzeroberfläche wurde unter Berücksichtigung von Design-Heuristiken entwickelt, um die Benutzer bei der Evaluierung zu unterstützen.

In quantitativen Tests mit 20 Testpersonen konnte gezeigt werden, dass die Benutzeroberfläche nach der System-Usability-Scale sehr gut bedienbar ist. Die Evaluation der verwendeten Modelle ergab, dass GermaNet im Verhältnis zu den generierten Erweiterungswörtern, die meisten relevanten Erweiterungswörter fand. Das kombinierte Modell übertraf zwar nicht die Modelle, aus denen es zusammengesetzt ist. Aber, da die gefundenen Erweiterungswörter von verschiedenen Modellen stammen, sind sie diverser, als die eines einzelnen Modells. Während in der lexikalischen Ressource des Hamburger Transparenzportals durchschnittlich nur ein Erweiterungswort für ein Suchwort definiert ist, wählten die Testpersonen durchschnittlich zwei Erweiterungswörter pro Suchwort aus. Folglich sind die gefundenen Erweiterungswörter, denen aus der lexikalischen Ressource qualitativ ebenbürtig und quantitativ überlegen.

Zuletzt konnte durch einen explorativen Versuch gezeigt werden, dass die Automatisierung des vorgestellten Erweiterungssystems mit Hilfe von Doc2Vec möglich sein könnte.

---

## **Danksagung**

An dieser Stelle möchte ich mich zuerst bei meinen unermüdlichen Betreuern Chris Biemann und Lothar Hotz bedanken, die mich bei der Erstellung dieser Arbeit tatkräftig unterstützten und an die ich mit jederzeit mit Fragen wenden konnte. Außerdem möchte ich meinen Dank an alle Testpersonen aussprechen, ohne die das Evaluieren des Erweiterungssystems nicht möglich gewesen wäre. Ich möchte mich bei Saba Anwar für die Unterstützung beim Trainieren des JoBimText-Modells bedanken. Zuletzt möchte ich meinen Freunden, meiner Familie und meiner Freundin für die Unterstützung beim Erstellen dieser Arbeit danken.

---

# 1 Einleitung

## 1.1 Motivation

Unsere Welt wird zunehmend komplexer, außerdem entwickelt sich unsere Sprache stetig weiter. Für viele Fachbereiche hat sich inzwischen eine Subkultur der Sprache gebildet, statt nur einige Fachbegriffe zu besitzen. Dies wird zum Problem, sobald Laien sich online selbstständig in ein solches Fachgebiet einarbeiten wollen, oder in dessen Bereich nach Informationen suchen. Daraus folgt, dass zuerst verstanden werden muss, wie sich die eigene Sprache in die Fachsprache übertragen lässt, um dann in dieser weiter suchen zu können.

Diese Problematik lässt sich am Beispiel des Hamburger Transparenzportals<sup>1</sup> zeigen. Dies ist ein Informationsregister, welches infolge des Hamburgischen Transparenzgesetzes<sup>2</sup> entstanden ist. In dem Portal (siehe Abbildung 1.1) werden Daten und Dokumente von über 50 verschiedenen Quellen veröffentlicht. Dazu gehören Hamburger Behörden, das Luftmessnetz oder der HVV. Zu diesem Zeitpunkt enthält das Transparenzportal über 100.000 Datensätze. Zu einem Datensatz können zudem noch mehrere Ressourcen gehören (siehe Abbildung 1.2). Die Datensätze und Ressourcen sind zum Teil in behördlicher Fachsprache verfasst, wodurch eine gezielte allgemeinsprachliche Suche schwerer ist.

Eine Möglichkeit, allgemeinsprachliche Suchanfragen (Querys) in einen spezifischen Kontext zu übersetzen, bietet die Suchanfragenerweiterung (Query Expansion). Hierbei wird eine Suchanfrage um Wörter erweitert, die ähnlich zu den vorkommenden Wörtern in der Suchanfrage sind. Ein Beispiel hierfür wäre, dass bei der Suchanfrage *Kita Altona* nach *Kita*, *Kindertagesstätte*, *Kindergarten* und *Altona* gesucht wird. Dies hat den Vorteil, Dokumente finden zu können, die zu der Suchanfrage passen, aber nicht die vom Benutzer eingegebenen Wörter enthalten müssen. Query Expansion führt bei vollständig ausformulierten Suchanfragen kaum zu besseren Ergebnissen. Bei kurzen Suchanfragen kann die Relevanz der zurückgegebenen Dokumente aber deutlich erhöht werden [5].

In dieser Arbeit, werden die Suchanfragen, mit einem Wörterbuchansatz erweitert. Ein Wörterbuch ist eine Datei, in der festgelegt ist, mit welchen Erweiterungswörtern ein Wort einer Suchanfrage erweitert wird. Im vorherigen Beispiel wären für das Wort *Kita* die Erweiterungswörter *Kindertagesstätte* und *Kindergarten* im Wörterbuch gespeichert gewesen. Obwohl die Erweiterungswörter durch festgelegte Beziehungen für relevante

---

<sup>1</sup><https://www.hamburg.de/transparenzportal-hamburg> zuletzt abgerufen am 29.08.2019

<sup>2</sup>HmbGVBl. 2012, S. 271

The screenshot displays the homepage of the 'Transparenzportal Hamburg' (transparenz.hamburg.de). At the top, there is a navigation bar with links for 'SUCHE', 'TRANSPARENZPORTAL', 'TRANSPARENZGESETZ', 'OPEN DATA', 'HILFE', and 'KONTAKT'. Below this is a search bar containing the query 'UHH' and a search button. A checkbox option 'Nur neueste Version eines Datensatzes anzeigen' is checked. The search results are sorted by 'RELEVANZ' and show three entries:

- Haushaltsplan 2019/2020, Nachbewilligung nach § 35 Landeshaushaltsordnung (LHO), Anmietung des ehemaligen Fernmeldeamtes an der...** (30.08.2019, PDF). Informationsgegenstand: Mitteilungen des Senats.
- Haushaltsplan 2019/2020: Nachbewilligung nach § 35 der Landeshaushaltsordnung und Mitteilung des langjährigen Trends der...** (30.08.2019, PDF). Informationsgegenstand: Mitteilungen des Senats.
- Entwurf eines Gesetzes zum Staatsvertrag über die Hochschulzulassung** (01.08.2019, PDF). Informationsgegenstand: Mitteilungen des Senats.

On the right side, there is a sidebar with 'IHRE SUCHANFRAGE' (Stichwort: UHH, Nur neueste Versionen: ja) and 'SUCHERGEBNISSE EINSCHRÄNKEN:' with filters for KATEGORIEN, INFORMATIONSgegenSTAND, DATENTYP, FORMAT (HTML (5), PDF (317), XLSX (1), ZIP (1)), and LIZENZ.

Abbildung 1.1: Beispielsuche mit Ergebnissen für den Query *UHH*

Ergebnislisten sorgen, hat dieser Ansatz auch Nachteile. Bei einer stetig wachsenden Dokumentensammlung muss beachtet werden, dass sich die Sprache der Dokumentensammlung durch neue Dokumente verändern kann. So können neue Fachbegriffe eingeführt werden oder bisher wenig verwendete Wörter in neuen Dokumenten häufiger vorkommen. Durch diese Effekte verliert die Wörterbuchdatei mit der Zeit ihren Wert. Eine Lösung ist das Aktualisieren des Wörterbuchs.

Hierfür müssen Erweiterungswörter gefunden werden, die den Ergebnissen einer Suchanfrage relevante Dokumente hinzufügen. Es gibt verschiedene Möglichkeiten, diese für ein Suchwort zu finden. Zum einen existieren Verfahren, die auf Basis der Dokumentensammlung mit Textverarbeitung Erweiterungswörter generieren können. Andererseits gibt es lexikalische Ressourcen, wie die Wörterbuchdatei, in denen über Relationen ähn-

**Transparenzportal Hamburg**  
 transparenz.hamburg.de

Hamburg | Behörde für Kultur und Medien Staatsarchiv

[SUCHE](#) | [TRANSPARENZPORTAL](#) | [TRANSPARENZGESETZ](#) | [OPEN DATA](#) | [HILFE](#) | [KONTAKT](#)

Portal durchsuchen  [ERWEITERTE SUCHE](#) | [HILFE ZUR SUCHE](#)

Nur neueste Version eines Datensatzes anzeigen

---

**V5770 Exitor DME - Aufbau, Betrieb und Support, App für Smartphones - geschützter Zugriff auf Mail-, Kalender- und Kontaktdaten**

Veröffentlichende Stelle: Finanzbehörde  
 Veröffentlichungsdatum: 24.09.2014

Vertrag, um den Zugriff von Smartphones auf dienstliche Mail-, Kalender- und Kontaktdaten zu ermöglichen. Dies geschieht mittels einer geschützten und verschlüsselten APP.

**DETAILINFORMATIONEN**

DATENTYP	Dokument
INFORMATIONEN-Verträge der Daseinsvorsorge GEGENSTAND	
KATEGORIE	Öffentliche Verwaltung, Haushalt & Steuern
E-AKTENZEICHEN	FB2a.800.01-2/3
GESAMTE METADATEN	<a href="#">Link zum Download (JSON)</a> <a href="#">Link zum Download (N3)</a> <a href="#">Link zum Download (RDF)</a> <a href="#">Link zum Download (TTL)</a> <a href="#">Link zum Download (XML)</a> <a href="#">Link zum Download (JSONLD)</a>

**Ressourcen**

Name: Akte:FB2a.800.01-2/3  
 Format: PDF (Größe: 1.9 MB)

Lizenz: Datenlizenz Deutschland Namensnennung 2.0

Anzahl der Ressourcen: 1

Namensnennung: Freie und Hansestadt Hamburg

Schlagwörter: Mobilität, Smartphone

Abbildung 1.2: Beispielsuche mit Ergebnissen für den Query *UHH*

liche Wörter definiert sind. Gemeinsam haben beide Verfahren, dass die gefundenen Erweiterungswörter nicht ohne Test in die Wörterbuchdatei aufgenommen werden können. Bei den Verfahren, die Textverarbeitung nutzen, basieren die Erweiterungswörter auf statistischen Verfahren. Deshalb kann nicht garantiert werden, dass alle gefundenen Erweiterungswörter zum Suchwort passen oder die Ergebnisse um relevante Dokumente erweitern. Bei den lexikalischen Ressourcen handelt es sich meist um allgemeingültige Wortsammlungen. Das Problem hierbei ist, dass sich diese nicht immer in spezielle Kontexte übernehmen lassen. So kann es dazu kommen, dass die definierten Erweiterungswörter zwar semantisch ähnlich zum Eingabewort sind, aber in dem betrachteten Kontext nicht zu einer Verbesserung der gefundenen Dokumente führt. Um zu verhindern, dass der Integrität des Wörterbuchs vom Hamburger Transparenzportal geschadet wird, ist es notwendig, das Verhalten des Systems beim Hinzufügen eines Erweiterungsworts durch einen Domainexperten zu prüfen.

## 1.2 Fragestellung

Aus diesen Gründen wurde im Rahmen dieser Bachelorarbeit geprüft, wie sich eine lexikalische Ressource zur Query Expansion semi-automatisch erweitern lässt. Hierfür wurden sieben Modelle zur Query Expansion erarbeitet und ausgewertet. Des Weiteren wurde geprüft, ob das Anwenden von Filterverfahren sinnvoll ist. Dies bildet die Grundlage für die Single-Page-Applikation, mit der ein Anwender beim Finden und Evaluieren von Erweiterungswörtern innerhalb eines speziellen Kontextes unterstützt werden soll. Die Bedienbarkeit (Usability) der Benutzeroberfläche wurde zudem durch Benutzertests belegt.

## 1.3 Aufbau der Arbeit

Am Anfang der Arbeit werden die verschiedenen Modelle und ihre Funktionsweise eingeleitet. Diese teilen sich in Wörterbuchansätze, Word Embeddings und dem JoBimText-Modell auf. Zudem werden Designvorlagen vorgestellt, welche helfen, die Usability einer Benutzeroberfläche zu erhöhen. Zuletzt werden verschiedene Evaluationstechniken eingeführt. Im nächsten Abschnitt werden die verschiedenen Systemkomponenten der vorgestellten Single-Page-Applikation erläutert. Dies beinhaltet neben der Benutzeroberfläche auch die Modellverarbeitung. Außerdem wird darauf eingegangen, wie Query Expansion mit der Suchmaschine Solr<sup>3</sup> funktioniert und wie dies für das vorgestellte System verwendet wird. Der nächste Abschnitt legt dar, wie das System evaluiert wurde. Darauf folgt das Auswerten der Experimente. Es werden die Ergebnisse der Usability-Tests besprochen, sowie Verbesserungsmöglichkeiten erörtert. Daneben werden die verwendeten Modelle untereinander verglichen und geprüft, ob alle dazu geeignet sind, in diesem Kontext gute Erweiterungswörter zu finden.

## 1.4 Verwandte Arbeiten

Query Expansion ist ein wichtiges Thema im Information Retrieval (IR). Es wurden in verschiedenen Veröffentlichungen unterschiedliche Methoden zur Query Expansion vorgestellt.

Wie bereits erwähnt, sind in lexikalischen Ressourcen semantisch ähnliche Wörter durch Relationen miteinander verbunden. Diese können auf verschiedene Arten generiert werden. Bei von Hand erstellten Thesauri setzt man die Wörter manuell in Relation. Bei der automatisierten Generierung von Thesauri können Relationen zwischen Wörtern anhand von Kookkurrenz gefunden werden. Kookkurrenz ist das gemeinsame Auftreten von Wörtern. Eine weitere Möglichkeit zur automatischen Generierung betrachtet die grammatikalische Funktion der Wörter in den Dokumenten. Ein Beispiel für eine sol-

---

<sup>3</sup><https://lucene.apache.org/solr/> zuletzt abgerufen am 04.09.2019

---

---

che Funktion wäre die Beziehung zwischen einem Verb und Nomen. Relationen werden hergestellt, indem Wörter gefunden werden, die ähnliche grammatikalische Funktionen erfüllen [18].

Neben den lexikalischen Ressourcen gibt es verschiedene Methoden zum automatischen Generieren von Erweiterungswörtern. Diese unterscheiden sich darin, ob sie Erweiterungswörter auf der Basis der gesamten Dokumentensammlung (global) generieren, oder ob dafür nur ein Teil der Dokumente verwendet wird (lokal). Beim lokaler Query Expansion werden Wörter zum Erweitern der Querys nur in relevanten Dokumenten ermittelt. Das Auswählen dieser kann auf verschiedene Weisen erfolgen. User-generated Relevance Feedback stellt eine Möglichkeit zum Bestimmen relevanter Dokumente dar. Hier bewerteten Nutzer die Relevanz der Dokumente, die durch eine Suchanfrage angezeigt werden. Innerhalb dieser Dokumente wird dann für die Suchanfrage nach Erweiterungswörtern gesucht. So werden die Suchergebnisse prozedural verbessert. Problematisch ist jedoch, dass Nutzer selten ihre Suchergebnisse bewerten. Deshalb wurde Pseudo-Relevance-Feedback (PRF) entwickelt [22]. Hier werden die Wörter zum Erweitern aus den ersten  $n$  Dokumenten generiert, die bei der Suche mit dem originalen Query geliefert werden [14].

Wenn Word Embeddings zum Ermitteln von Erweiterungswörtern verwendet werden, konnte gezeigt werden, dass lokale Ansätze zu besseren Ergebnissen als globale führen [6]. Dagegen wurden Ergebnisse präsentiert, die zeigen, dass PRF im Vergleich zu einem globalen Ansatz zu geringfügig schlechteren Ergebnissen führt [1]. Trotzdem sind State-of-the-Art Verfahren für Query Expansion meist PRF-Verfahren. Bei diesen ist das Auswählen und Gewichten der Dokumente zum Finden der Erweiterungswörter ein wichtiges Thema. Außerdem wird geprüft, wie die Terme innerhalb der Dokumente mit Gewichten versehen werden können, um die Relevanz der gefundenen Dokumente zu erhöhen [28]. Beispielmuster hierfür sind Bo1, Bo2 und KL [35].

Das Erweitern der Suchanfrage mit diesen automatischen Verfahren birgt jedoch ein Risiko. Es kann infolge des Hinzufügens der Erweiterungswörter zu Query-Drift kommen [17]. Bei diesem Phänomen wird das Ziel einer Suchanfrage durch die hinzugefügten Erweiterungswörter in eine andere Richtung gelenkt. In der Literatur wird dies häufig in Verbindung mit PRF diskutiert, da Erweiterungsmethoden mit einer kleineren Dokumentenbasis zum Generieren von Erweiterungswörtern anfälliger hierfür ist. Aber auch bei globalen Erweiterungsmethoden kann es zu Query-Drift kommen. Query-Drift tritt auf, wenn in den Dokumenten, die für die Erweiterung betrachtet werden, neben den Wörtern, die zum gesuchten Ziel führen, auch häufig solche enthalten sind, die einem anderem Thema angehören. Wenn ein Erweiterungsmodell eines dieser Wörter zur Erweiterung der Suchanfrage verwendet, werden Dokumente gefunden, die nicht relevant für die Suchanfrage sind. Es wurden Verfahren vorgestellt, um den Query-Drift von Erweiterungswörtern zu messen. Darüber ist es möglich, verwendete Erweiterungsmodelle miteinander zu vergleichen [32].

---

Anders als bei automatischen Verfahren ist für den betrachteten Anwendungsfall die Qualität der gefundenen Erweiterungswörter wichtiger, als die Relevanz der gefundenen Dokumente. Die Erweiterungsmodelle werden nicht über die Relevanz der gefundenen Dokumente nach der Query Expansion bewertet, sondern es werden die gefundenen Erweiterungswörter selbst bewertet. Das liegt daran, dass eine lexikalische Ressource erweitert wird. Da diese den Anspruch hat, nur gute Erweiterungswörter zu beinhalten, müssen die Erweiterungswörter und nicht die gefundenen Dokumente bewertet werden. Sonst könnte trotz relevanten Dokumenten auch ein Erweiterungswort, dessen Verschlechterung der gefundenen Dokumente durch die restlichen Erweiterungswörter abgeschwächt wird, der lexikalischen Ressource hinzugefügt werden.

---

## 2 Grundlagen

In diesem Abschnitt werden verschiedene Modelle zum Generieren von Erweiterungswörtern im Detail erklärt. Außerdem werden Heuristiken zur Oberflächenentwicklung vorgestellt. Zuletzt werden Evaluierungsmethoden präsentiert. Diese Verfahren wurden verwendet, um das Erweiterungssystem zu entwickeln.

### 2.1 Wörterbuch für Query Expansion

Anstelle von Modellen die Erweiterungswörter generieren, werden auch lexikalische Ressourcen verwendet. Diese werden auch Wörterbuch oder Thesaurus genannt. Lexikalische Ressourcen sind nicht auf eine Art von Relation beschränkt. In ihnen werden Synonyme mit verschiedenen Relationen unterschieden [8]. Wie in Abschnitt 1.4 besprochen, können diese mit unterschiedlichen Verfahren erstellt werden. Im Folgenden werden zwei manuell erstellte lexikalische Ressourcen vorgestellt und deren Grenzen diskutiert. Für das Erweiterungssystem sind Wörterbuchansätze relevant, da in ihnen semantisch ähnliche Wörter definiert sind, auf die zugegriffen werden kann, um Erweiterungswörter zu finden.

#### 2.1.1 WordNet

WordNet wurde an der Universität Princeton entwickelt [8][18]. Denen in ihm definierten Wörtern sind Synsets zugeordnet. Ein Synset stellt eine Bedeutung des Worts dar. *Bank* hat beispielsweise Synsets für die Bedeutung als *Kreditinstitut*, sowie als *Sitzgelegenheit*. Einem Synset in WordNet sind verschiedene Synonym-Relationen zugeordnet. Zu jedem Wort gehört eine kurze Definition und Listen von Hyponymen, Hypernymen und Teilvon-Relationen<sup>1</sup>.

In Experimenten wurde gezeigt, dass bei Suchanfragen mit wenigen Wörtern die Suchergebnisse durch Query Expansion mit denen in WordNet codierten Synonymen verbessert werden können [36]. Eine andere Arbeit zeigte, dass sich semantische Ähnlichkeit zweier Wörter am besten über deren Definitionen in WordNet feststellen lässt. Sie verglich diese Relation mit anderen, die in WordNet codiert sind. Dieses Ergebnis konnte noch weiter verbessert werden, indem die Definitionsrelationen mit Ähnlichkeitsfunktionen, welche auf zusammen auftretenden Wörtern (Kookkurrenz) basieren, kombiniert werden [8].

---

<sup>1</sup><https://wordnet.princeton.edu/> zuletzt abgerufen am 17.05.2019

---

### 2.1.2 GermaNet

Die deutsche Variante von WordNet heißt GermaNet<sup>2</sup> [11]. Es ist ebenfalls eine lexikalische Ressource. Hier sind für deutsche Wörter verschiedene Arten von Synonymen gespeichert. Der Aufbau folgt dem von WordNet. Einem Wort sind verschiedene Synsets zugeordnet, welche die verschiedenen Bedeutungen repräsentieren. In diesen Synsets sind Holonyme, Meronyme, Hypernyme und Hyponyme zu Wörtern vermerkt. Da Holonyme und Meronyme Teil-von Beziehungen sind, welche nicht zu ähnlichen Wörtern führen, wurden sie nicht zur Query Expansion betrachtet. Während Hypernyme Oberbegriffe zu einem Wort darstellen, sind Hyponyme Spezifikationen, zu denen das Suchwort ein Oberbegriff bildet.

### 2.1.3 Grenzen des Wörterbuchansatzes

Wie die zuvor genannten Arbeiten gezeigt haben, ist Query Expansion mit lexikalischen Ressourcen möglich und führt besonders bei unterspezifizierten Suchanfragen zu einer Verbesserung der Ergebnisse. Jedoch hat das Verwenden einer lexikalischen Ressource auch seine Nachteile. Zum einen können Probleme durch Mehrdeutigkeit von Wörtern auftreten. Insbesondere, wenn diese in verschiedenen Kontexten unterschiedliche Bedeutungen haben. Ein klassisches Beispiel ist *Schimmel*, welches sowohl ein *Pferd* als auch einen *Pilz* bezeichnen könnte [25]. Ein weiteres Problem besteht darin, dass eine manuell erstellte lexikalische Ressource eine statische Sammlung von Relationen ist, die sich nicht automatisch mit der Sprache weiterentwickelt. Dies kann dazu führen, dass neue Begriffe nicht mit bisher verwendeten Begriffen in Relation gesetzt werden.

## 2.2 Word Embedding

Eine Methode zum Generieren von Erweiterungswörtern ist das Finden von Wortrepräsentationen (Word Embeddings). Diese wird häufig für automatische Query Expansion verwendet. Hierfür werden die Wortrepräsentationen aus einer Menge von Dokumenten generiert. Die unterschiedlichen Wörter der Dokumente bilden das Vokabular. Für jede Vokabel wird eine Wortrepräsentation erstellt. Diese können die Form eines Vektors (Vektorrepräsentation) annehmen. Die Menge aller Vektorrepräsentation eines Vokabulars wird Vektorraum genannt. Die semantische Ähnlichkeit zweier Wörter wird durch die Distanz zwischen zwei Vektorrepräsentationen bestimmt. Die Distanz zweier Vektoren kann über die Kosinus-Ähnlichkeit oder den Euklidischen Abstand berechnet werden. Um die  $n$  ähnlichsten Wörter für ein Wort zu finden, können Verfahren wie der  $k$ -Nearest-Neighbor-Algorithmus verwendet werden [1]. Das im Folgenden vorgestellte Verfahren *Word2Vec* lernt Wortrepräsentationen mit der Hilfe eines neuronalen Netzwerkes [21]. Das Verfahren ermöglicht die Verwendung von verschiedenen Modellen, um

---

<sup>2</sup><http://www.sfs.uni-tuebingen.de/GermaNet/> zuletzt abgerufen am 16.09.2019

---

Wortrepräsentationen zu lernen.

Neuronale Netzwerke (NN) bestehen aus verschiedenen Schichten (Layern). Das erste Layer nimmt die Eingabewerte des NN entgegen. Dies wird Input-Layer genannt. Mit diesem können entweder ein Hidden-Layer oder das Output-Layer verbunden sein. Ein Hidden-Layer führt arithmetische Funktionen auf den Ergebnissen des vorausgegangenen Layers aus und gibt die Ergebnisse auf das nächste Layer weiter. Das Output-Layer führt ebenfalls arithmetische Funktionen aus und stellt das Ergebnis bereit. Zuvor werden die Ergebnisse normalisiert und anschließend ausgegeben. Die Eingabewerte eines NN sind Vektoren.

Um den Input verarbeiten zu können, werden die Input-Vektoren mit Gewichtsmatrizen multipliziert. Die Werte der Matrizen werden Gewichte genannt. Die Gewichte werden während des Trainings angepasst, um den erwarteten Output zu erhalten.

### 2.2.1 CBOW

Das erste betrachtete Modell heißt Continuous Bag-of-Words (CBOW) und besteht aus einem Input-Layer, Hidden-Layer und einem Output-Layer (siehe Abbildung 2.1). Das Hidden-Layer wird im Kontext dieses Modells Projection-Layer genannt. Ziel des Modells ist es, die Gewichtsmatrix des Projection-Layers zu trainieren. Der Grund hierfür ist, dass sich in den Zeilen der Gewichtsmatrix die Vektorrepräsentationen aller Vokabeln befinden. Um das neuronale Netz trainieren zu können, benötigt es ein Ziel um die Gewichte während des Trainings sinnvoll anzupassen. CBOW soll eine Vokabel, mit  $2n$  Wörtern aus dessen Kontext in den Dokumenten finden. Der Kontext setzt sich aus den  $n$  Wörtern, welche vor der Vokabel in einem Trainingstext stehen und den  $n$  Wörtern, die der Vokabel folgen, zusammen.

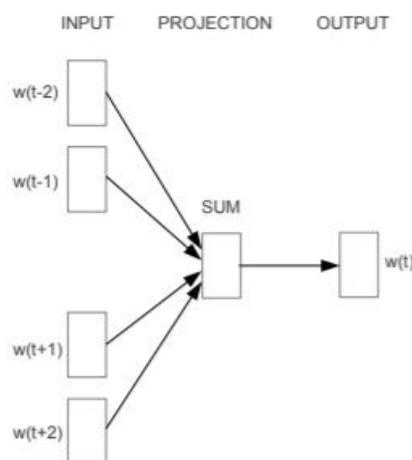


Abbildung 2.1: CBOW-Modell:  $w(t-2)$  bis  $w(t+2)$  sind die Kontextwörter,  $w(t)$  ist die Vokabel. In diesem Beispiel ist  $n=2$  [21].

Für jedes der Kontextwörter der Vokabel, erhält das Input-Layer einen One-Hot-Vektor als Eingabe. Diese Vektoren enthalten  $V$  Werte, wobei  $V$  die Anzahl der Vokabeln ist.

Jeder dieser Werte repräsentiert ein Wort aus dem Vokabular. Deshalb sind alle Werte der Vektoren Null, mit Ausnahme dessen, der das Kontextwort repräsentiert. Dieser Wert ist Eins. Der One-Hot-Vektor  $\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$  steht z.B. für das erste Wort aus einem Vokabular mit drei Wörtern. In dem folgenden Beispiel wird ein One-Hot-Vektor mit einer  $3 \times 3$  Matrix multipliziert.

$$\begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 21 & 34 & 5 \\ 11 & 26 & 40 \\ 5 & 0 & 54 \end{pmatrix} = \begin{pmatrix} 11 & 26 & 40 \end{pmatrix} \quad (2.1)$$

Die One-Hot Vektoren  $(x_1, \dots, x_m)$  werden mit einer  $V \times N$  Gewichtsmatrix  $W^T$  multipliziert.  $V$  ist die Anzahl der Vokabeln und  $N$  die Größe des Projection-Layers. Die Gewichtsmatrix  $W^T$  enthält nach dem Training die Vektorrepräsentationen für alle Vokabeln. Anschließend werden die Ergebnisse aufsummiert und der Durchschnittswert gebildet, indem durch die Anzahl der Kontextwörter  $C$  geteilt wird.

$$h = \frac{1}{C} W^T (x_1 + x_2 + \dots + x_n) \quad (2.2)$$

Da das Projection-Layer eine lineare Aktivierungsfunktion verwendet, wird  $h$  zum Output-Layer weitergegeben. Dort wird  $h$  mit einer weiteren Gewichtsmatrix  $W'$  der Form  $N \times V$  multipliziert.

$$h' = hW' \quad (2.3)$$

Dies resultiert in einem Vektor der Größe  $V$ . Um daraus eine Wahrscheinlichkeitsverteilung über alle Vokabeln zu erzeugen, wird eine Aktivierungsfunktion wie z.B. Softmax verwendet. Softmax normalisiert den Vektor auf Werte zwischen Null und Eins [21][20][30].

### 2.2.2 SkipGram

Das SkipGram-Modell ist ähnlich aufgebaut wie das CBOW-Modell. Auch hier wird das neuronale Netz trainiert, um aus der Gewichtsmatrix, zwischen Input und Projection-Layer, einen Vektorraum aufzubauen. Statt wie beim CBOW-Modell aus einer Menge von Kontextwörtern die Vokabel zu ermitteln, hat das SkipGram-Modell die Aufgabe, mit der Vokabel jene Kontextwörter zu ermitteln, die am häufigsten in Verbindung mit der Vokabel vorkommen (siehe Abbildung 2.2).

Als Input erhält das Input-Layer einen One-Hot-Vektor  $x$ , welcher die betrachtete Vokabel repräsentiert. Dieser hat erneut  $V$  Einträge, wobei  $V$  die Anzahl aller Vokabeln ist. Dieser wird mit der  $V \times N$  Gewichtsmatrix  $W^T$  zwischen Input und Projection-Layer multipliziert, aus der nach dem Training der Vektorraum erzeugt wird. Da der One-Hot-Vektor nur an der  $n$ -ten Stelle eine Eins hat und das Projection-Layer eine lineare

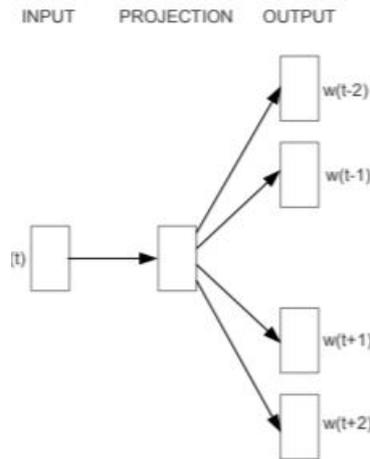


Abbildung 2.2: SkipGram-Modell:  $w(t)$  ist die Vokabel, während  $w(t-2)$  bis  $w(t+2)$  die Zielkontextwörter sind. In diesem Beispiel ist  $n=2$  [21].

Aktivierungsfunktion verwendet, ist der Output des Projection-Layers die  $n$ -te Zeile der Gewichtsmatrix.

$$h = W^T x \quad (2.4)$$

Mit dem Vektor  $h$  werden  $2n$  neue Vektoren berechnet. Aus diesen werden anschließend Wahrscheinlichkeitsverteilungen für jedes der  $n$  Wörter vor und nach der Vokabel berechnet. Die neuen Vektoren erhält man, indem das Kreuzprodukt aus dem Vektor  $h$  und der  $N \times V$  Gewichtsmatrix des Output-Layers gebildet wird.

$$h' = hW' \quad (2.5)$$

Um die Werte von  $h'$  zu einer Wahrscheinlichkeitsverteilung zu verwandeln, wird eine Aktivierungsfunktion wie Softmax eingesetzt<sup>3</sup> [21][20][16].

### 2.2.3 Aktivierungsfunktion-Optimierung

Um zu ermitteln, welche der Wörter aus dem Vokabular am wahrscheinlichsten zu einem Eingabewort passen, wird eine Wahrscheinlichkeitsverteilung über alle Vokabeln benötigt. Hierfür werden die Werte aus den Output-Layern mit Softmax auf Werte zwischen Null und Eins normalisiert. Diese stellen die Wahrscheinlichkeiten dar. Im Folgenden wird die Softmax-Funktion am Beispiel des CBOW-Modells vorgestellt.

Um die Wahrscheinlichkeit für eine Vokabel  $w_j$  für die Kontextvokabel  $w_l$  aus dem Input-Layer zu erhalten, wird die folgende Formel verwendet.

<sup>3</sup><http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/> zuletzt abgerufen am 29.04.2019

$$p(w_j|w_I) = \frac{\exp(v'_{wj} v_{wI})}{\sum_{j'=1}^V \exp(v'_{wj'} v_{wI})} \quad (2.6)$$

$v'_{wj}$  ist die  $j$ -te Spalte der Gewichtsmatrix zwischen dem Projection- und Output-Layer.  $v_{wI}$  ist die Vektorrepräsentation der Kontextvokabel aus der Gewichtsmatrix zwischen Input- und Projection-Layer. Die Summe  $\sum_{j'=1}^V$  iteriert über alle Vokabeln und multipliziert die Vektorrepräsentation der Kontextvokabel mit der  $j'$ -te Spalte der Gewichtsmatrix zwischen dem Projection- und Output-Layer.

Softmax zu verwenden ist teuer, da die Laufzeit proportional zu der Anzahl der Vokabeln  $V$  wächst. Um die Laufzeit zu verbessern, kann die Menge an Operationen pro Vokabel verringert werden. Dies wird durch hierarchischen Softmax erreicht. Hierfür wird ein binärer Baum aufgebaut (siehe Abbildung 2.3). Jedes der Blätter (weiß) von diesem stellt eine Vokabel aus dem Vokabular dar. Statt zum Berechnen der Wahrscheinlichkeit einer Vokabel über das gesamte Vokabular iterieren zu müssen, werden nur  $L(w) - 1$  Iterationen benötigt.  $L(w)$  ist die Länge des Pfades von der Wurzel zum Blattknoten der betrachteten Vokabel. Somit werden pro Vokabel nicht  $V$ , sondern nur  $\log_2(V)$  Iterationen benötigt [21][30][26].

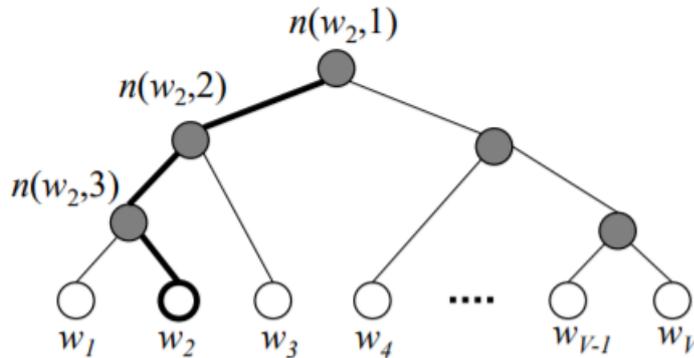


Abbildung 2.3: Binärer Baum für hierarchischen Softmax mit dem Pfad zur Vokabel  $w_2$ .  $n(w_2, j)$  bezeichnet den  $j$ -ten Knoten zum Blatt von  $w_2$  [30].

Die Wahrscheinlichkeit, dass die betrachtete Vokabel  $w$  die erwartete Vokabel  $w_O$  ist, wird wie folgt berechnet.

$$p(w|w_O) = \prod_{j=1}^{L(w)-1} \sigma([n(w, j+1) = ch(n(w, j))] * v'_{n(w, j)} v_{wI}) \quad (2.7)$$

Wie beim regulären Softmax bezeichnet  $v_{wI}$  die Vektorrepräsentation der Kontextvokabel.  $n(w, j)$  bezeichnet den  $j$ -ten Knoten zum Blatt von  $w$ . Jedem dieser Knoten ist ein Output-Vektor (Gewicht)  $v'_{n(w, j)}$  zugeordnet.  $ch(n)$  ist ein zuvor festgelegtes Kind (links oder rechts) vom Knoten  $n$ .

Hier ist  $w_I$  die Vektorrepräsentation, welche das Projection-Layer an das Output-Layer

weitergibt.  $n(w, j)$  ist der  $j$ -te Knoten auf dem Weg von der Wurzel zum Blatt von  $w$ .

$$\sigma = \frac{1}{1 + \exp(-x)} \quad (2.8)$$

Zudem wird die Sigmoid-Funktion  $\sigma$  verwendet und  $[[x]]$  wie folgt definiert.

$$[[x]] = \begin{cases} 1 & x \text{ ist wahr} \\ -1 & \text{sonst} \end{cases} \quad (2.9)$$

Eine weitere Möglichkeit zum Optimieren von Softmax ist das Negative Sampling. Statt die Berechnung an sich zu Optimieren wird bei Negative Sampling die Anzahl der zu aktualisierenden Gewichte reduziert. Anstelle von allen Gewichten wird nur eine kleine Menge  $W_{neg}$  (5-20) aktualisiert. Hierfür wird das gesamte Vokabular, bis auf die erwartete Vokabel  $w_0$  betrachtet. Abseits dessen, werden die Gewichte der erwarteten Vokabel  $w_0$  angepasst, da diese gefunden werden soll. Die Menge der zu aktualisierenden Vokabeln  $W_{neg}$  wird zufällig auf der Basis von einer Verteilung  $P_n(w)$  ausgewählt. Ein Beispiel für eine Verteilung ist die Unigram-Verteilung ( $P(w)$ ). Auf dieser Basis ist es wahrscheinlicher, dass häufigere Wörter ausgewählt werden. Zudem wurde empirisch gezeigt, dass diese Verteilung am besten funktioniert, wenn der Exponent der Terme auf  $3/4$  gesetzt wird<sup>4</sup> [21][30][23].

Die Loss-Funktion zum Aktualisieren der Gewichte der berechnet sich wie folgt.

$$E = -\log\sigma(v_{w_0}^T v_{w_I}) - \sum_{w_j \in W_{neg}} \log\sigma(-v_{w_j}^T v_{w_I}) \quad (2.10)$$

$v_{w_I}$  beschreibt weiterhin die Vektorrepräsentation der Kontextvokabel.  $v_{w_0}^T$  ist der Output-Vektor der erwarteten Vokabel.  $W_{neg}$  beschreibt die Menge der schlechten Vokabeln, die über die Verteilung  $P_n(w)$  ausgewählt wurden.  $v_{w_j}^T$  ist die Vektorrepräsentation der  $j$ -ten schlechten Vokabel.

#### 2.2.4 Doc2Vec

Word Embeddings lassen sich ebenfalls verwenden, um die Ähnlichkeit zwischen Dokumenten zu bestimmen. Hierfür kann Doc2Vec [15] verwendet werden. Das Bestimmen der Ähnlichkeit zwischen Dokumenten ist relevant, da es ermöglicht, die gefundenen Erweiterungswörter automatisch zu evaluieren. Je ähnlicher die Menge der hinzugefügten Dokumente, zu denen der originalen Suche ist, desto besser ist das evaluierte Erweiterungswort. Um dies zu erreichen, wird zu jedem Dokument eine Vektorrepräsentation gelernt. Wie zuvor wird die Ähnlichkeit zweier Dokumente über die Distanz zwischen den Vektorrepräsentationen berechnet. Diese werden wie bei Word2Vec ermittelt. Die Modelle von Doc2Vec stellen eine Erweiterung von Word2Vec dar. Statt das erwartete

<sup>4</sup><http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/> zuletzt abgerufen 17.09.2019

Wort nur mit dem Kontext zu finden, wird zusätzlich der Paragraph-Vektor (PV) berücksichtigt. Hierfür wird neben der Matrix, in der Wortrepräsentationen festgehalten werden, eine weitere Matrix für die Dokumente eingeführt. Jede Spalte dieser Matrix repräsentiert einen Paragraph-Vektor. Dieser stellt die Vektorrepräsentation des Dokuments dar. Wie schon bei Word2Vec gibt es zwei verschiedene Modelle für Doc2Vec.

Das Distributed Memory Model of Paragraph Vectors (PV-DM) ist ähnlich aufgebaut wie das CBOW-Modell von Word2Vec (siehe Abbildung 2.4). Der Unterschied zum CBOW-Modell besteht darin, dass jedes Dokument einer Vektorrepräsentation zugeordnet ist und diese als zusätzlicher Eingabeparameter an das Modell übergeben wird. So wird die Paragraph-Matrix analog zu der Matrix mit den Wortrepräsentationen trainiert.

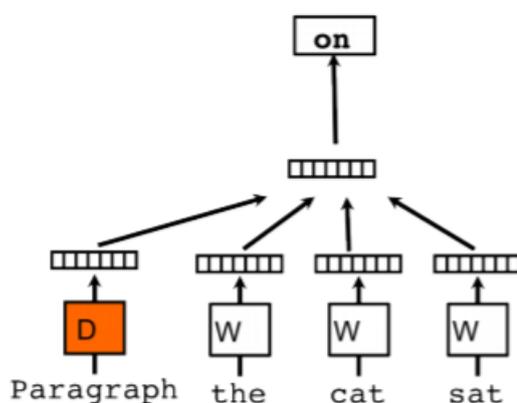


Abbildung 2.4: Aufbau des PV-DM-Modells [15]. Um das erwartete Wort *on* zu finden, werden die One-Hot-Vektoren der Kontextwörter (weiß), sowie der Paragraph-Vektor (orange) verwendet.

Das zweite Modell heißt Distributed Bag of Words version of Paragraph Vector (PV-DBOW). Der Aufbau dieses Modells ist ähnlich zu dem SkipGram-Modell von Word2Vec (siehe Abbildung 2.2.4). Statt die Kontextwörter aus einem Wort zu generieren, wird hierfür der Paragraph-Vektor verwendet. Um dies zu erreichen, wird ein zufälliges Wort aus einem Dokument ausgewählt. Für den Trainingsdurchlauf werden dessen Kontextwörter betrachtet. Die Aufgabe des Modells besteht darin zu entscheiden, ob die Kontextwörter in dem Dokument des Paragraph-Vektors vorkommen. Da das System keine Wörter als Input erhält, existiert keine Matrix mit Wortrepräsentationen. Aus diesem Grund ist dieses Modell speichereffizienter als PV-DM.

Wie auch bei den Word2Vec-Modellen, können hierarchischer Softmax und Negative Sampling zum Verringern der Trainingszeit verwendet werden.

## 2.3 JoBimText

JoBimText ist eine Open-Source-Plattform für graph-basierte distributionelle Semantik. Mit diesen können unter anderem semantisch ähnliche Wörter gefunden werden. Zuvor wird ein bipartiter Graph erstellt, indem die betrachtete Dokumentensammlung verar-

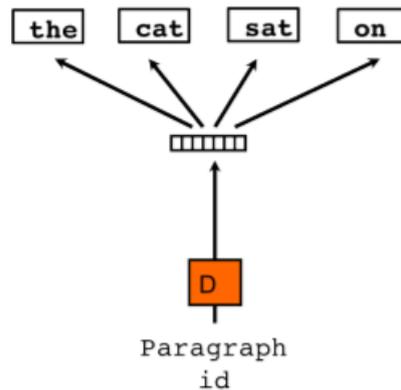


Abbildung 2.5: Aufbau des PV-DBOW-Modells [15]. Es wird der Paragraph-Vektor (orange) verwendet, um die Kontextwörter (weiß) eines zufälligen Eingabewortes zu ermitteln.

beitet wird. Hierfür werden die Dokumente in Fragmente, z.B. Sätze aufgeteilt. Innerhalb dieser werden Wortpaare gebildet, welche auf der grammatikalischen Beziehungen zwischen dem Wortpaar beruht (siehe Abbildung 2.6). Das betrachtete Wort wird *Jo* (Term) genannt und eines der Partnerwörter *Bim* (Kontext). Das Bilden der Wortpaare wird Holing-Verfahren genannt. Die zwei Mengen des bipartierten Graphen bilden jeweils die Menge aller Elemente von *Jos* und *Bims* ab. Die Kanten des Graphen besitzen Gewichte, welche die Häufigkeit der Relation zwischen *Jo* und *Bim* kennzeichnet.

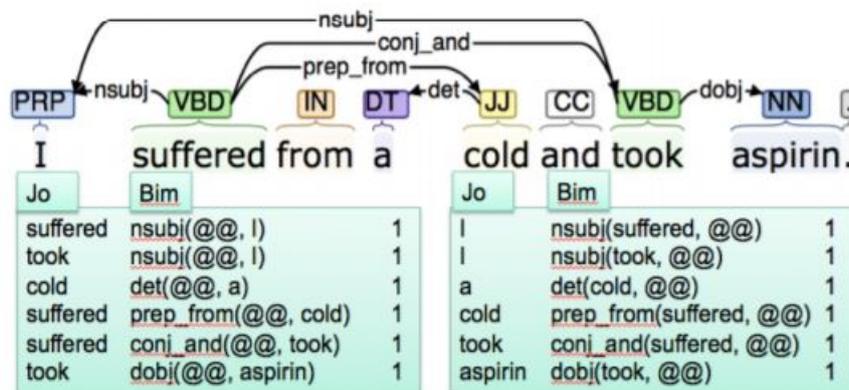


Abbildung 2.6: Ergebnis des Holing-Verfahrens mit *Jos* und *Bims* für den Satz *I suffered from a cold and took aspirin.* [3].

Um aus diesem Graphen einen Thesaurus zu generieren, werden die  $p$  häufigsten Relationen für jedes  $Jo$  gefunden. Diese Teilmenge wird zum Graphen erster Ordnung  $FO_{Jo}$ . Um nun die Ähnlichkeit zwischen *Jos* evaluieren zu können, wird ein Ähnlichkeitsgraph wie folgt erstellt. Seien  $t_1$  und  $t_2$  *Jos*, so ist das Gewicht der Kante zwischen ihnen definiert als Anzahl von *Bims*, welche sowohl eine Relation zu  $t_1$  als auch zu  $t_2$  haben. Um diese Relationen zu finden, wird  $FO_{Jo}$  verwendet. Dieser Graph  $SO_{Jo}$  stellt den Thesaurus dar.

Je mehr *Bims* sich zwei *Jos* teilen, desto ähnlicher sind sie sich [3][4].

## 2.4 Usability

Bereits zu Beginn der Entwicklung des Erweiterungssystems war bekannt, dass die Verfahren zum Finden von Erweiterungswörtern nicht nur gute Erweiterungswörter finden. Aus diesem Grund müssen diese manuell von Domainexperten evaluiert werden. Hierfür wurde eine Benutzerschnittstelle entworfen, welche diese beim Evaluieren der Erweiterungswörter unterstützt. Dafür sind die im Folgenden beschriebenen Design-Heuristiken beachtet worden. Diese beschreiben Aspekte einer guten Benutzerschnittstelle. An diesen wurde sich bei der Entwicklung der Benutzerschnittstelle orientiert, um die Usability zu erhöhen.

Die erste Heuristik sind die 8 goldenen Regeln von Ben Shneiderman [31].

1. Konsistenz: Es ist beim Verwenden von Interaktionsschnittstellen darauf zu achten, bekannte Muster oder Icons zu verwenden. So ist dem Benutzer intuitiv klar, welche Systemreaktionen bei der Interaktion zu erwarten sind.
2. Universelle Bedienbarkeit: Die Benutzeroberfläche sollte Anfängern und Experten passende Interaktionsmöglichkeiten bieten, um sie bei ihrer Arbeit zu unterstützen.
3. Rückmeldungen für Benutzeraktionen: Wenn das System Zeit zum Verarbeiten von Daten braucht, muss es dies dem Nutzer signalisieren.
4. Abgeschlossene Operationen: Sobald eine Aktion in mehreren Schritten abgeschlossen werden muss, sollten diese Schritte zusammenhängend präsentiert werden.
5. Fehler vorbeugen: Das System sollte durch das Design Fehlern vorbeugen, z.B. indem die Eingabe des Nutzers auf festgelegte Werte beschränkt ist.
6. Einfaches Rückgängigmachen: In jedem Systemzustand sollte der Nutzer in der Lage sein, seine Aktionen wieder rückgängig zu machen.
7. Nutzer steuert Programm: Die Benutzeroberfläche sollte sich nicht ohne Zutun des Benutzers verändern, um bei diesem ein Gefühl von Kontrolle und Sicherheit zu erzeugen.
8. Kurzzeitgedächtnis schonen: In der Benutzerschnittstelle sollten alle relevanten Informationen sichtbar sein, sodass der Benutzer nicht darauf angewiesen ist, sich diese zu merken.

Die nächste Design-Heuristik stellt zehn Prinzipien auf, nach denen eine Benutzerschnittstelle konzipiert werden sollte. Entwickelt wurden diese von Jakob Nielsen<sup>5</sup> [27].

---

<sup>5</sup><https://www.nngroup.com/articles/ten-usability-heuristics/> zuletzt abgerufen 27.09.2019

---

1. Sichtbarkeit des Systemstatus: Das System informiert den Benutzer in angemessener Zeit über den momentanen Systemstatus.
2. Verbindung zwischen System und Welt: Die Sprache, welche verwendet wird, um mit dem Nutzer zu kommunizieren, sollte sich an der realen Welt orientieren und nicht aus systeminternen Fachbegriffen bestehen.
3. Benutzerkontrolle und Freiheit: Wenn ein Benutzer ein Fehler macht, sollte er die Möglichkeit haben, dies rückgängig zu machen, ohne von vorne zu beginnen.
4. Konsistenz und Standards: Innerhalb des Systems sollten gleichartige Designelemente immer die gleiche Funktion auslösen, sodass der Nutzer nie rätseln muss, wie das System auf eine Aktion reagieren wird.
5. Fehlervermeidung: Statt einer guten Fehlermeldung ist es besser, mögliche Fehler vorher durch Design auszuschließen.
6. Wiedererkennen statt Erinnern: Um die Bedienung eines Systems für den Benutzer leichter zu gestalten, ist es besser, Informationen, die ein Nutzer bereits kennt, wieder aufrufbar zu machen oder weiterhin anzuzeigen, anstatt von ihm zu verlangen, sich diese zu merken.
7. Flexibilität und Effizienz: Es sollte Interaktionsmöglichkeiten geben, welche den Interaktionsprozess für erfahrende Benutzer beschleunigt.
8. Minimalistisches Design und Ästhetik: Dialogfenster sollten keine irrelevanten Informationen enthalten.
9. Benutzer bei Fehlerhandhabung unterstützen: Fehlermeldungen sollten in natürlicher Sprache geschrieben werden und Lösungsvorschläge beinhalten.
10. Hilfen und Dokumentation: Am besten wäre ein System, welches ohne Anleitung auskommt. Wenn aber eine solche vonnöten ist, dann sollte sie durchsuchbar sein, aufgabenorientiert, konkrete Arbeitsschritte beschreiben und so kurz wie möglich sein.

Die Heuristiken von Nielsen und Regeln von Shneiderman überschneiden sich häufig. Beide thematisieren jedoch Bereiche, auf welche der jeweils andere weniger eingeht. Nielsen legt Wert auf Benutzerinteraktionen, indem er anmerkt, dass alle Veränderungen der Benutzeroberflächen vom Benutzer initiiert werden sollen. Außerdem sollen aufeinanderfolgende Arbeitsschritte zusammenhängend dargestellt werden, um die Interaktionsmöglichkeiten deutlicher zu gestalten. Währenddessen ist die Benutzerunterstützung ein Schwerpunkt für Shneiderman. Er fordert, dass Fehlermeldungen in natürlicher Sprache den Fehler beschreiben werden und Lösungsvorschläge beinhalten. Zusätzlich soll der Benutzer durch eine Dokumentation oder Hilfe über Interaktionsmöglichkeiten unterstützt werden. Es wurden beide Ansätze für das Entwickeln der Benutzeroberfläche des

Erweiterungssystems herangezogen, um verschiedene Aspekte von Usability zu berücksichtigen.

## 2.5 Evaluierungs-Methoden

### 2.5.1 System Usability Scale

Die System Usability Scale (SUS) ist ein Fragebogen zur Evaluierung der Usability von Benutzeroberflächen (Siehe Abbildung 2.7) [34]. Er wird häufig verwendet und besteht aus 10 Fragen, die jeweils auf einer Skala mit 5 Abstufungen von *Trifft voll zu* bis *Trifft nicht zu* beantwortet werden. Das Ergebnis der SUS berechnet sich wie folgt.

		<i>Trifft nicht zu</i>			<i>Trifft voll zu</i>	
1.	Ich kann mir sehr gut vorstellen, das System regelmäßig zu nutzen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.	Ich empfinde das System als unnötig komplex.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	Ich empfinde das System als einfach zu nutzen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.	Ich denke, dass ich technischen Support brauchen würde, um das System zu nutzen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.	Ich finde, dass verschiedene Funktionen des Systems gut integriert sind.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.	Ich finde, dass es im System zu viele Inkonsistenzen gibt.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	Ich kann mir vorstellen, dass die meisten Leute das System schnell zu beherrschen lernen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8.	Ich empfinde die Bedienung als sehr umständlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.	Ich habe mich bei der Nutzung des Systems sehr sicher gefühlt.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.	Ich musste eine Menge Dinge lernen, bevor ich mit dem System arbeiten konnte.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Abbildung 2.7: Die zehn Fragen des SUS-Fragebogens.

$$Q_i = \begin{cases} Score_i - 1, & i \text{ ist ungerade} \\ 5 - Score_i, & i \text{ ist grade} \end{cases} \quad (2.11)$$

$Q_i$  ist das Ergebnis für die  $i$ -te Frage. Es wird zwischen Fragen mit gerader und ungerader Nummerierung bei der Score-Berechnung unterschieden. Diese Reihenfolge ist durch den Fragebogen festgelegt.

$$Score_{ges} = 2,5 * \sum_{i=1}^{10} Q_i \quad (2.12)$$

Die Ergebnisse für alle Fragen werden aufsummiert und anschließend mit 2,5 multipliziert. Dies ist nötig, um die Ergebniszahl auf einer Skala von 0 – 100 darzustellen. Je höher das Ergebnis ist, desto besser ist die Usability der Benutzeroberfläche. Jedoch sollte eine Benutzeroberfläche, dessen Usability zufriedenstellend ist, mindestens 75 Punkte erreichen [2].

### 2.5.2 Fleiss-Kappa

Um zu messen, wie sehr die Antworten von beliebig vielen Testpersonen übereinstimmen, kann das Fleiss-Kappa-Verfahren verwendet werden [9]. Der berechnete Wert zeigt die Übereinstimmung der Antworten, welche die erwartete zufällige Übereinstimmung übertrifft. Fleiss-Kappa ist eine Erweiterung von Cohens-Kappa, mit dem die Urteilsübereinstimmung für nur zwei Bewerter festgelegt wird. Im Folgenden wird erklärt, wie sich der Fleiss-Kappa-Wert berechnet.

In den folgenden Formeln betitelt  $N$  die Anzahl der bewerteten Fälle. Ein Fall ist beispielsweise ein Erweiterungswort für ein gegebenes Suchwort.  $d$  bezeichnet die Gesamtanzahl der Testpersonen/Bewerter und  $d_{ij}$  bezeichnet die Menge an Testpersonen, die für den  $i$ -ten Fall die  $j$ -te Kategorie ausgewählt haben. Kategorien kennzeichnen in diesem Anwendungsfall, ob ein Erweiterungswort ausgewählt wurde oder nicht.

$$p_i = \frac{1}{d(d-1)} \sum_{j=1}^z (d_{ij}^2 - d_{ij}) \quad (2.13)$$

$p_i$  beschreibt die Übereinstimmung der Testpersonen für das  $i$ -te Erweiterungswort.

$$p_0 = \frac{1}{N} \sum_{i=1}^N p_i \quad (2.14)$$

$p_0$  bildet den Mittelwert über alle  $p_i$ -Werte.

$$p_j = \frac{1}{N * d} \sum_{i=1}^N d_{ij} \quad (2.15)$$

$p_j$  ist das Verhältnis vom Auswählen der Kategorie  $j$  über alle Fälle und der Anzahl aller Bewertungen von Fällen.

$$p_c = \sum_{j=1}^z p_j^2 \quad (2.16)$$

$p_c$  stellt den Erwartungswert des Zufalls dar.

$$\kappa = \frac{p_0 - p_c}{1 - p_c} \quad (2.17)$$

$\kappa$  ist der Fleiss-Kappa-Wert des Verfahrens. Dieser wird für alle Eingabewörter gebildet.

Zur Interpretation der Ergebnisse gibt es verschiedene Skalen. Fleiss schlug eine Skala vor, dessen Werteintervalle zu groß sind um entscheidende Einblicke in die Übereinstimmungen von Testpersonen zu erhalten [10]. Eine Skala mit kleineren Intervallen wurde in [13] vorgeschlagen (siehe Tabelle 2.1).

Fleiss-Kappa-Wert	Interpretation
< 0	Schlechte Übereinstimmung
0.01 - 0.20	Geringe Übereinstimmung
0.21 - 0.40	Mittelmäßige Übereinstimmung
0.41 - 0.60	Angemessene Übereinstimmung
0.61 - 0.80	Beachtliche Übereinstimmung
0.81 - 1.0	Fast perfekte Übereinstimmung

Tabelle 2.1: Skala zur Bewertung von Fleiss-Kappa-Werten [13].

### 2.5.3 Box-Whisker-Diagramm

Mit dem Box-Whisker-Diagramm (siehe Abbildung 2.8) werden Verteilungen von Daten dargestellt. Darin eingeschlossen sind verschiedene statistische Auswertungen.

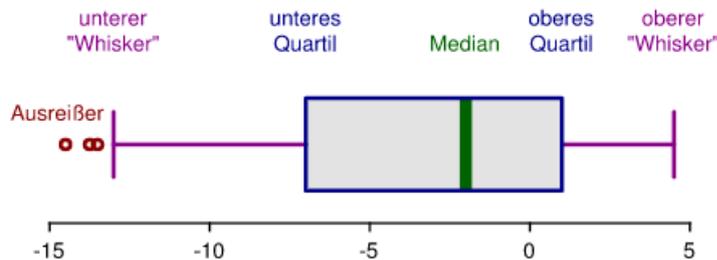


Abbildung 2.8: Beispiel für ein Box-Whisker-Diagramm <sup>6</sup>

Die Daten werden in Quartile aufgeteilt. Das heißt, sie werden in vier gleichgroße Mengen geordnet. Die Box stellt 50% der Daten dar. Das untere Quartil (Q1) kennzeichnet, dass 25% und das obere Quartil (Q3), dass 75% der Daten kleiner als der angegebene Wert sind. Über die Größe des Abstands zweier Quartile (Interquartilsabstand) lässt sich die maximale Länge der *Whisker* bestimmen. Häufig wird  $1,5 * \text{Interquartilsabstand}$  verwendet. Die oberen und unteren *Whisker* kennzeichnen jeweils die maximalen- und minimalen Werte der Daten, die innerhalb der erlaubten Whiskerlänge liegen. Datenpunkte, die außerhalb der erlaubten Whiskerlänge sind, werden als Ausreißer bezeichnet [19].

In den Grundlagen wurden verschiedene Methoden zum Finden von Erweiterungswörtern beschrieben. Außerdem wurden verschiedene Heuristiken zum Erstellen von guten Benutzeroberflächen vorgestellt. Diese wurden verwendet, um ein System zum Erweitern einer lexikalischen Ressource zu entwickeln. Das Erweiterungssystem wird

<sup>6</sup><https://de.wikipedia.org/wiki/Box-Plot> zuletzt abgerufen 18.09.2019

im nächsten Kapitel beschrieben. Die zuletzt vorgestellten Evaluierungsmethoden fanden Verwendung beim Bewerten des Erweiterungssystems.

---



### 3 Aufbau des Wortexpansionssystems

Mit dem vorgestellten Erweiterungssystem soll die lexikalische Ressource zur Query Expansion des Hamburger Transparenzportals semi-automatisch erweitert werden. Hierfür wurden verschiedene Erweiterungsmodelle entwickelt und kombiniert. Zusätzlich wurde eine Benutzeroberfläche zum Testen und Bewerten der gefundenen Erweiterungswörter entwickelt. Zu Beginn wird ein Überblick über die Dokumente und Synonymdatei des Hamburger Transparenzportals gegeben. Es wird vorgestellt, wie eine Suchanfrage erweitert wird. Außerdem wird beleuchtet, wie das Erweiterungssystem mit der Suchplattform des Hamburger Transparenzportals kommuniziert. Anschließend werden die Erweiterungsmodelle und Benutzeroberfläche vorgestellt.

#### 3.1 Daten und Dokumente des Hamburger Transparenzportals

Um die Word-Embedding-Modelle und das JoBimText-Modell trainieren zu können, wurde eine Dokumentsammlung benötigt. Die Dokumente, auf denen die Modelle trainiert wurden, sind die Volltexte (siehe Abbildung 3.1) der Datensätze des Hamburger Transparenzportals. Diese Volltexte bestehen aus einer Mischung aus Fließtext und Stichworten. Insgesamt wurden zum Trainieren der Erweiterungsmodelle 130.832 Dokumente verwendet.

	Tagesordnung	Bürgerinfo	Home	Bezirksversammlung	Ausschüsse	
Fraktionen	Sitzungen	Kalender	Übersicht	Drucksachen	Übersicht	Recherche
Textrecherche	Sitzungsteilnehmende	Dokumente	Kontakt	Impressum	Hilfe	
Legende	Bezirksversammlung	Altona	Tagesordnung - Sitzung des			
Planungsausschusses	Bezeichnung		Sitzung	des Planungsausschusses	Gremium	
Planungsausschuss	Datum	Mi 19.02.2014	Status	öffentlich/nichtöffentlich	Zeit 1800 - 2045	Anlass
Sitzung	Raum	Sitzungsraum	des Technischen Rathauses	EG	Ort	Jessenstraße 1 22767 Hamburg
Nebeneingang	Jessenstraße	Ecke	Virchowstraße	TOP		Betreff
Drucksache	Ö 1	Öffentliche	Fragestunde	Ö 2	Bauvorhaben an der Hauptkirche St.	
Trinitatis	Auswertung der öffentlichen		Anhörung vom 11.12.2013	Verschoben aus der Sitzung vom		
05.02.2014	Ö 3	Bebauungsplan-Entwurf	Ottensen 60	Behringstraße	Sachstandsbericht des	
Vorhabenträgers	Ö 4	Weiterentwicklung des Schulstandorts		Königstraße /	Struenseestraße	
Vorstellung des weiteren	Vorgehens durch die beauftragten			Planungsbüros	Ö 5	Bebauungsplan-
Entwurf Sülldorf 3	Osterfeld	Vorstellung des Energiekonzeptes		durch den Vorhabenträger	Ö 6	
Inklusive Empfehlungen	zum Städtebaulichen Vertrag		Mitte Altona vom Forum	"Eine Mitte für Alle"		
Schreiben vom 10.02.2014	Neufassung	XIX-3820	Ö 7	Bebauungsplan-Entwurf	Rissen 11	Beschluss
zur Feststellung	Beschlussdrucksache des Amtes		XIX-3839	Ö 8	Stellungnahme zur Öffentlichen	
Auslegung	Bebauungsplan Nr. 88	"Businesspark	Elbufer	Wedel"	Mitteilungsdrucksache des Amtes	
XIX-3840	Ö 9	Bebauungsplan	Lurup 37	Einleitung eines		Bebauungsplanverfahrens
dem Bauausschuss vom 11.02.2014	Ö 10	Mitteilungen		Ö 11	Verschiedenes	
N 12	N 12					
Einleitung eines	Planänderungsverfahrens für das Grundstück		Kieler Straße	Eingabe vom 24.01.2014		
N 13	Senatsdrucksachen-Entwurf		"Agrarpolitisches Konzept 2020"	Stellungnahme des Amtes		
N 14	Mitteilungen	N 15	Verschiedenes			

Abbildung 3.1: Beispielvolltext eines Datensatzes des Hamburger Transparenzportals.

Die Synonymdatei des Hamburger Transparenzportals ist wie folgt aufgebaut. Pro Zei-

le sind die Synonyme durch ein Komma getrennt (siehe Abbildung 3.2). Diese erweitern sich gegenseitig. Insgesamt enthält die Synonymdatei 8.433 Synonyme. Im Durchschnitt enthält jede Zeile etwa zwei Wörter. Folglich ist für ein Wort durchschnittlich ein Synonym definiert.

```
Konjunktur, Wirtschaftstrend
Annahme des Gesetzes, Ablehnung des Gesetzes
Rat der Gemeinden und Regionen Europas, RGR
Treuhandrat UNO, UN-Treuhandrat
Sicherheitsrat UNO, UN-Sicherheitsrat
Staaten des Europarates, Länder des Europarates
Asiatisch-Pazifischer Rat, ASPAC
Konsolidierung der Schuld, Umschuldung, Schuldenkonsolidierung
Konservenfabrik, Konservenindustrie
Verbraucher, Konsument
Verbrauch der privaten Haushalte, Ausgaben der privaten Haushalte
Nordischer Rat, Skandinavischer Rat
Europäischer Rat, Konferenz der Staats- oder Regierungschefs der EG
landwirtschaftlicher Berater, Agrarberater
Staaten des Nordischen Rates, Länder des Nordischen Rates
Erhaltung der Fischbestände, Überfischung
Lebensmittelkonservierung, Haltbarmachung von Lebensmitteln
Konservatismus, Neokonservatismus
Erhaltung der Ressourcen, Schutz der Ressourcen
Wirtschaftsgebäude, Bauernhaus, Stall, Scheune
Stahlbau, Stahlkonstruktion
Schiffbau, Werftindustrie, Werft, Schiffbauindustrie
Anhörung der Arbeitnehmer, Anhörung der Gewerkschaften
Wahlanfechtungen, Wahlstreitigkeiten
Annahme des Haushaltsplans, Verabschiedung des Haushaltsplans, Feststellung des Haushaltsplans
Pro-Kopf-Verbrauch, Verbrauch je Einwohner
Verfassung, Verfassungsgesetz
Partei Gründung, Parteiverbot, Parteiprogramm, Parteiauflösung
Kontrolle der Verfassungsmäßigkeit, Verfassungsmäßigkeit der Gesetze
Verwaltungskontrolle, Kontrolle der öffentlichen Verwaltung
Steuerpflichtiger, Steuerzahler, Steuerschuldner
```

Abbildung 3.2: Ausschnitt aus der Synonymdatei des Hamburger Transparenzportals

## 3.2 Solr als Erweiterungswortsammlung

Apache Solr<sup>1</sup> ist eine Open-Source Suchplattform, die auf dem Apache Lucene Index beruht. In einem Index werden vorverarbeitete Daten gespeichert, um gespeicherte Dokumente schnell durchsuchen zu können. Für die Verwendung von Solr wurde sich an dem Hamburger Transparenzportal orientiert. Solr wird von dem Hamburger Transparenzportal für das Retrieval und die Query Expansion verwendet.

Innerhalb von Solr wird eine Suchanfrage, welche aus einem oder mehreren Wörtern bestehen kann, als eine Menge von Token betrachtet. Ein Token repräsentiert ein Wort. Auf den Token werden verschiedenen Verfahren zur Vorverarbeitung angewendet. Beispiele sind StopFilter zum Entfernen von Stoppwörtern, LowerCaseFilter zum Ersetzen von Groß- durch Kleinbuchstaben, oder der ManagedSynonymGraphFilter. Dieser Filter erweitert die Tokenmenge, um weitere Token, welche *Synonyme* für die Anfragetoken sind. Vom Hamburger Transparenzportal werden die folgenden Filter in der angegebenen Reihenfolge angewendet. Zuerst wird die Suchanfrage durch den WhitespaceTokeni-

<sup>1</sup><https://lucene.apache.org/solr/> zuletzt abgerufen am 04.09.2019

zer an Leerzeichen aufgetrennt. Anschließend werden die Token mit dem ManagedSynonymGraphFilter um Synonyme erweitert. Zum Auftrennen von Wörtern, die von nicht-alphanumerischen Zeichen getrennt werden oder in CamelCase geschrieben sind, wird der WordDelimiterGraphFilter verwendet. Darauf folgt der LowerCaseFilter. Zum Bilden des Wortstamms der Token wird der SnowballPorterFilter verwendet. Zuletzt werden mit dem ASCIIFoldingFilter alle Zeichen innerhalb der Token, die nicht unter den ersten 127 ASCII-Zeichen sind, in ihr ASCII-Äquivalent konvertiert.

Diese Synonyme werden für jedes Token mit einem Mapping (siehe Abbildung 3.3) festgelegt, welches in einer Synonymdatei gespeichert ist<sup>2</sup>.

Das Mapping innerhalb der Synonymdatei ordnet einem Token eine Liste mit weiteren Token zu. Die dort festgelegten Synonyme können auf verschiedene Arten festgelegt werden. Eine Möglichkeit zum Erstellen des Mappings ist das manuelle Definieren von Token-Listen-Paaren. Eine weitere Möglichkeit ist das Nutzen von vorgefertigten Synonymwörterbüchern. Ein Beispiel hierfür ist GermaNet. Zuletzt gibt es die Option, diese Datei mit automatisch generierten Token-Synonym-Paaren zu erstellen.

```
{
  "initArgs":{
    "ignoreCase":true,
    "format":"solr"},
  "initializedOn":"2019-08-12T16:34:15.539Z",
  "managedMap":{
    "Strom":[
      "Elektrizität",
      "Strom"],
    "Wasser":[
      "Regenwasser",
      "Wasser"]}]}
```

Abbildung 3.3: Beispiel für eine verwaltete Synonymdatei mit Erweiterungswörtern für *Strom* und *Wasser*.

In dem vorgestellten System wird die Synonym-Datei dazu verwendet Token-Listen-Paare zu speichern, welche die Ergebnismenge um weitere relevante Dokumente für das Token erweitert. Das heißt, die neuen Dokumente behandeln dasselbe Thema, wie der Großteil derer, die ohne Query Expansion gefunden werden würden. Um diese Wörter von dem Backend zu dem Solr-Server übermitteln zu können, werden Managed Resources verwendet<sup>3</sup>. Dies ermöglicht es GET/POST-Requests an den Solr-Server zu senden, mit denen sich Token-Listen-Paare abfragen, hinzufügen und löschen lassen. Das Hinzufügen beinhaltet zwei Möglichkeiten Token-Listen-Paare in die Synonymdatei einzufügen. Die erste Möglichkeit fügt nur ein einziges Paar hinzu. Dafür wird dem Post-Request ein Objekt der Form  $\{Token : [Element_1, \dots, Element_n]\}$  hinzugefügt. Die zweite Möglichkeit besteht darin, dem Post-Request eine Liste  $[Token, Element_1, \dots, Element_n]$  zu übergeben. Dies führt dazu, dass für jedes Element der Liste ein Token-Listen-Paar erstellt wird, wobei die Liste aus den restlichen Elementen besteht.

<sup>2</sup>[https://lucene.apache.org/core/6\\_4\\_1/analyzers-common/org/apache/lucene/analysis/synonym/SynonymGraphFilter.html](https://lucene.apache.org/core/6_4_1/analyzers-common/org/apache/lucene/analysis/synonym/SynonymGraphFilter.html) zuletzt abgerufen am 03.04.2019

<sup>3</sup>[https://lucene.apache.org/solr/guide/6\\_6/managed-resources.html](https://lucene.apache.org/solr/guide/6_6/managed-resources.html) zuletzt abgerufen am 12.08.2019

In dem vorgeschlagenen System wird nur die erste Möglichkeit, welche ein Token-Liste-Paar hinzufügt, verwendet. Der Grund hierfür ist, dass die Erweiterungswörter nur für ein Suchwort getestet werden. Zusätzlich sind die Erweiterungswörter oft spezifischer als das Suchwort, weshalb die Relation nicht reflexiv ist. Um die Erweiterungswörter zu testen und in der Synonymdatei zu speichern, werden von der Benutzeroberfläche verschiedenen Schnittstellen der Modellverarbeitung angesprochen. Diese stellen wiederum Anfragen an Solr oder speichern Erweiterungswörter in der Synonymdatei.

Die erste Schnittstelle, die von der Benutzeroberfläche angesprochen werden kann, findet Erweiterungswörter für ein Eingabewort. Von der Benutzeroberfläche wird das Eingabewort übergeben. Zusätzlich wird der Name des Erweiterungsmodells übergeben. Mit dem Erweiterungsmodell werden dann die Erweiterungswörter generiert. Zuletzt werden aus der Synonymdatei von Solr eventuell bestehende Erweiterungswörter ausgelesen. Falls ein generiertes Erweiterungswort in den bestehenden Erweiterungswörtern vorkommt, wird dieses Wort als bereits hinzugefügt markiert. Zuletzt werden die generierten Erweiterungswörter an die Benutzeroberfläche übergeben.

Die nächste Schnittstelle führt die Initialsuche aus. Hierfür wird von der Benutzerschnittstelle das Eingabewort übergeben. Mit diesen wird ein HTTP-Request an den Solr-Server geschickt, um die ersten 100 Dokumente zu erhalten. Neben den Dokumenten wird von Solr ebenfalls der Kontext gefordert, indem das Eingabewort vorkommt. Aus der Antwort von Solr wird anschließend die Gesamtanzahl der gefundenen Dokumente ausgelesen. Außerdem wird für jedes Dokument die ID, der Titel und der zugehörige Kontext extrahiert. Die IDs werden gespeichert, um später entscheiden zu können, ob ein gefundenes Dokument neu oder alt ist. Abschließend werden Dokumenttitel, Kontexte und Gesamtanzahl der gefundenen Dokumente an die Benutzeroberfläche zurückgegeben.

Eine der wichtigsten Schnittstellen ist verantwortlich für das Testen von Erweiterungswörtern. Beim Testen werden entweder Erweiterungswörter hinzugefügt oder entfernt. Um dies zu ermöglichen, wird das Eingabewort und die Liste mit Erweiterungswörtern übergeben. Insgesamt gibt es drei Möglichkeiten, wie sich die Schnittstelle verhält.

Die erste Möglichkeit tritt ein, wenn ein Erweiterungswort hinzugefügt wurde. Zuerst wird die Liste der Erweiterungswörter der Synonym-Datei von Solr hinzugefügt. Dann wird eine Suchanfrage mit dem Eingabewort an Solr geschickt. Die Verarbeitung der Antwort der Suche unterscheidet sich von der Initialsuche in einem Punkt. Während bei der Initialsuche alle Dokumente als bekannt markiert werden, wird nun auf die gespeicherten IDs zurückgegriffen. Jedes Dokument, dessen ID Teil der gespeicherten IDs ist, wird als neu markiert. Nun werden die IDs der gefundenen Dokumente, die alten IDs überschreiben und die zur Erweiterung verwendeten Synonyme abgespeichert. Am Ende wird der Ursprungszustand der Synonym-Datei wiederhergestellt. Dazu werden die Erweiterungswörter aus der Datei ausgelesen und entfernt. Anschließend werden wieder jene hinzugefügt, die nicht in der Liste der Erweiterungswörter vorkommen.

---

Wenn ein Erweiterungswort entfernt wurde, aber die Liste der Erweiterungswörter nicht leer ist, tritt die zweite Möglichkeit ein. Wie zuvor, werden die Erweiterungswörter der Synonym-Datei hinzugefügt und eine Suche gestartet. Anders ist, dass keine Dokumente als neu markiert werden. Die IDs der gefundenen Dokumente und die Erweiterungswörter überschreiben die alten Daten. Die hinzugefügten Erweiterungswörter werden nach den zuvor beschriebenen Verfahren entfernt.

Die letzte Möglichkeit tritt ein, wenn ein Erweiterungswort entfernt wurde und die Liste der Erweiterungswörter leer ist. Dann wird nur eine Suche ausgeführt, wie sie in der zweiten Möglichkeit beschrieben ist.

Das Speichern der Erweiterungswörter stellt die letzte Schnittstelle dar. Hierfür übergibt die Benutzeroberfläche erneut die Erweiterungswörterliste und das Eingabewort. Die Erweiterungswörter werden durch die API von Solr der Synonym-Datei hinzugefügt.

### 3.3 Generierung von Erweiterungswörtern

Für die Generierung und Verarbeitung von Erweiterungswörtern wurde ein Python-Backend entwickelt. Python als Entwicklungssprache wurde gewählt, da es eine Vielzahl nützlicher Erweiterungen bereitstellt, auf die im Folgenden weiter eingegangen wird. Das Backend unterteilt sich in zwei Bereiche. Der erste betrifft das Generieren von Erweiterungswörtern, während der zweite die Kommunikation zwischen Solr, den Erweiterungsmodellen und dem Frontend übernimmt.

#### 3.3.1 Word Embeddings

Für die Word Embedding Modelle wird das Gensim-Modul<sup>4</sup> verwendet. In diesen ist Word2Vec für SkipGram und CBOW nach Mikolov [20] implementiert. Ein Vorteil von Gensim ist, dass die Modelle in CPython geschrieben sind, sodass sich die neuronalen Netzwerke schnell trainieren lassen [29].

Es wurden insgesamt vier verschiedene Modelle trainiert. Für SkipGram und CBOW wurde jeweils ein Model mit hierarchischen Softmax als Aktivierungsfunktion und eins mit Negative Sampling und Softmax als Aktivierungsfunktion trainiert. Gensim lässt nicht zu, den hierarchischen Softmax mit Negative Sampling zu kombinieren. Bei dem Trainieren der Modelle auf den Dokumenten des Hamburger Transparenzportals, wurden nur Wörter berücksichtigt, die mindestens zehnmals in diesen vorkamen. So konnte die Trainingszeit reduziert und die Relevanz der gefundenen Erweiterungswörter erhöht werden. Wörter, die weniger als zehnmals in der Dokumentensammlung vorkommen, können als Erweiterungswort die Ergebnisse kaum erweitern und sind deshalb nicht relevant. Beim Training werden bei den beiden SkipGram-Modellen die Stoppwörter nicht

---

<sup>4</sup><https://radimrehurek.com/gensim/models/word2vec.html> zuletzt abgerufen am 05.10.2019

entfernt, um den Abstand, der zwischen verschiedenen Wörtern besteht, nicht zu verfälschen. Bei den Tests zeigte sich, dass die Stoppwörter nie als Erweiterungswörter vorgeschlagen wurden.

Beim Trainieren der Modelle fiel auf, dass beim selben Eingabewort teilweise gleiche Erweiterungswörter generiert wurden. Daraufhin wurde getestet, wie viele Überschneidungen die Erweiterungsmodelle haben. Dazu wurden für alle Wörter, aus der lexikalischen Ressource des Hamburger Transparenzportals, Erweiterungswörter durch die vier Modelle generiert. Dabei fiel auf, dass von den 8.433 Wörtern aus der lexikalischen Ressource, nur für 1.601 Wörter Erweiterungswörter gefunden werden konnten. Für die restlichen 6.832 Wörter konnten keine Erweiterungswörter gefunden werden, da sie nicht in den Dokumenten des Hamburger Transparenzportals vorkommen. Dies zeigt, dass ein datengetriebener Ansatz zum Finden von Erweiterungswörtern sinnvoller ist, als ausschließlich eine statische lexikalische Ressource zu verwenden.

Im Folgendem werden die Ergebnisse des Tests vorgestellt. Betrachtet werden die Überschneidungen zwischen den zwei CBOW und SkipGram-Modellen, sowie denen aller vier Modelle. Um die Überschneidungen der Erweiterungswörter gut darzustellen, wurden die gemeinsamen Erweiterungswörter ins Verhältnis zu den Suchwörtern gesetzt. In den Vergleichen wird die Menge der Suchwörter begrenzt, um Spezialfälle zu beleuchten. Zuletzt wird gezeigt, wie häufig sich die generierten Erweiterungswörter der betrachteten Modelle überschneiden.

Ein Suchwort liefert zehn EW <sup>5</sup>	gemeinsame Wörter pro Suchwort	gemeinsame Wörter pro Suchwort mit EW
CBOW (HS,NS) + SkipGram (HS,NS)	1,47 %	7,76 %
CBOW (HS,NS)	4,13 %	21,78 %
SkipGram (HS,NS)	6,54 %	34,46 %

Tabelle 3.1: Vergleich der gemeinsamen Wörter zwischen den Modellen. Für jeden Modell-Typ gibt es ein Modell mit Negative Sampling und eines mit hierarchischen Softmax als Aktivierungsfunktion.

Ein Suchwort liefert zehn EW	gemeinsame Wörter pro Suchwort bei der Modelle EW teilen	Häufigkeit von gemeinsamen Wörtern pro Suchwort mit EW
CBOW (HS,NS) + SkipGram (HS,NS)	18,47 %	42,04 %
CBOW (HS,NS)	28,67 %	75,95 %
SkipGram (HS,NS)	81,98 %	92,57 %

Tabelle 3.2: Vergleich der gemeinsamen Wörter, aber auch die Häufigkeit des Teilens von Wörtern zwischen Modellen.

Wie erwartet, ist die Anzahl der gemeinsamen Wörter pro Anfrage kaum aussagekräftig.

<sup>5</sup>EW steht für Erweiterungswörter

tig, da die meisten Anfragen keine Ergebnisse geliefert haben. Werden allerdings nur die Anfragen aller Modelle die ein Ergebnis haben verglichen, lässt sich erkennen, dass gemeinsame Wörter selten sind. Dagegen wurden bei CBOW etwa zwei von zehn Wörtern von beiden Modellen gefunden und bei den SkipGram-Modellen sogar drei von zehn (siehe Tabelle 3.1). Dass sich diese Modelle im Vergleich zu den CBOW-Modellen stark ähneln, wird noch deutlicher, wenn die gemeinsamen Wörter pro Anfrage mit mindestens einem gemeinsamen Wort betrachtet werden (siehe Tabelle 3.2). Es werden bei den SkipGram-Modellen im Durchschnitt etwa acht von zehn Wörtern von beiden Modellen gefunden (siehe Abbildung 3.4).



Abbildung 3.4: Beispiel für viele ähnliche Erweiterungswörter für die SkipGram-Modelle für das Suchwort *UHH*.

An der Häufigkeit von gemeinsamen Erweiterungswörtern pro Suchwort, für das Erweiterungswörter gefunden wurden, lässt sich feststellen, dass die SkipGram-Modelle sehr häufig mindestens ein gemeinsames Wort haben. Aber auch die CBOW-Modelle teilen sich häufig mindestens ein Erweiterungswort. Dagegen teilen sich alle vier Modelle in weniger als der Hälfte aller Fälle ein oder mehr Erweiterungswörter.

Aus dieser Auswertung wurde der Schluss gezogen, dass die zwei SkipGram-Modelle zu einem Modell kombiniert werden sollten. Hierbei wird vorgegangen, wie im Abschnitt 3.3.4 erklärt wird. Der Grund für diese Entscheidung ist, dass sich acht der gefundenen Erweiterungswörter der beiden SkipGram-Modelle in neun von zehn Fällen überschneiden. Würde dies nicht gemacht werden, würden die Erweiterungswörter der SkipGram-Modelle beim Kombinieren aller Modelle stärker gewichtet, da sie immer von mindestens zwei Modellen gefunden werden würden.

Die CBOW-Modelle werden nicht kombiniert, obwohl sie in drei von vier Fällen ein oder mehr gleiche Wörter finden. Grund hierfür ist, dass sie nicht mehr als drei gleiche Wörter finden. Würden sie kombiniert werden, könnten gute Erweiterungswörter verloren gehen.

### 3.3.2 GermaNet

Um Erweiterungswörter mit GermaNet zu finden, wurden nur die Hyponyme eines Wortes betrachtet. Hyperonyme würden die Suchergebnisse durch das Hinzufügen von Da-

tensätzen, die einen Oberbegriff betreffen, verallgemeinern. Da der Benutzer nach dem spezielleren Begriff gesucht hat, würde das Hinzufügen von einem allgemeineren Begriff zu Query-Drift führen.

Problematisch an den Synonymen der Synsets ist, dass diese nicht nach Relevanz sortiert sind. Um die zehn relevantesten Erweiterungswörter zu finden, wurde eine Worthäufigkeitsverteilung über alle Dokumente des Hamburger Transparenzportals erstellt. Mit dieser lassen sich die gefundenen Erweiterungswörter nach der Häufigkeit innerhalb der Dokumente sortieren. Eine weitere Bedingung ist, dass die Erweiterungswörter mindestens einmal in den Dokumenten vorkommen müssen. Somit sind die zehn generierten Erweiterungswörter nach Relevanz sortiert und in den Dokumenten vertreten.

### 3.3.3 JoBimText

Bei dem JoBimText-Modell wurde nur der erstellte Distributional Thesaurus (DT) verwendet, da die in ihm codierten Informationen ausreichen, um Erweiterungswörter zu generieren. Hierfür mussten die Dokumente des Hamburger Transparenzportals in Sätze aufgeteilt werden, auf denen das Modell trainiert wurde. Es wurden nur *Jos* (Terme) betrachtet, die sich mehr als zwei *Bims* (Kontexte) teilen. Anschließend ließen sich die Erweiterungswörter geordnet nach Relevanz aus dem DT auslesen. Es fiel auf, dass die gefundenen Erweiterungswörter meist semantisch nicht ähnlich zu dem Suchwort sind (siehe Abbildung 3.5). Einer Erklärung hierfür könnte sein, dass die Dokumentensammlung des Hamburger Transparenzportals zu klein ist, um das JoBimText-Modell ausreichend zu trainieren.

Webseite	Webseite	6.0
Webseite	Bürgerschaft	5.0
Webseite	FHH	3.0
Webseite	Internetseite	3.0
Webseite	BSB	3.0
Webseite	Maßnahme	3.0
Webseite	Planung	3.0
Webseite	Bezirksämter	3.0
Webseite	BSU	3.0
Webseite	Verwaltung	3.0

Abbildung 3.5: Beispiel aus dem Distributional Thesaurus für *Webseite*. Zahl zeigt die Anzahl der geteilten *Bims* an.

### 3.3.4 Kombiniertes Modell

Das letzte Modell kombiniert die sechs vorgestellten Modelle zu einem. Aufgrund der Ergebnisse der Evaluation des Word Embeddings werden die Modelle in zwei Schritten kombiniert (siehe Abbildung 3.6). Da sich die SkipGram-Modelle stark ähneln, werden sie zuerst zu einem Modell kombiniert und im Anschluss mit den verbleibenden vier kombiniert.

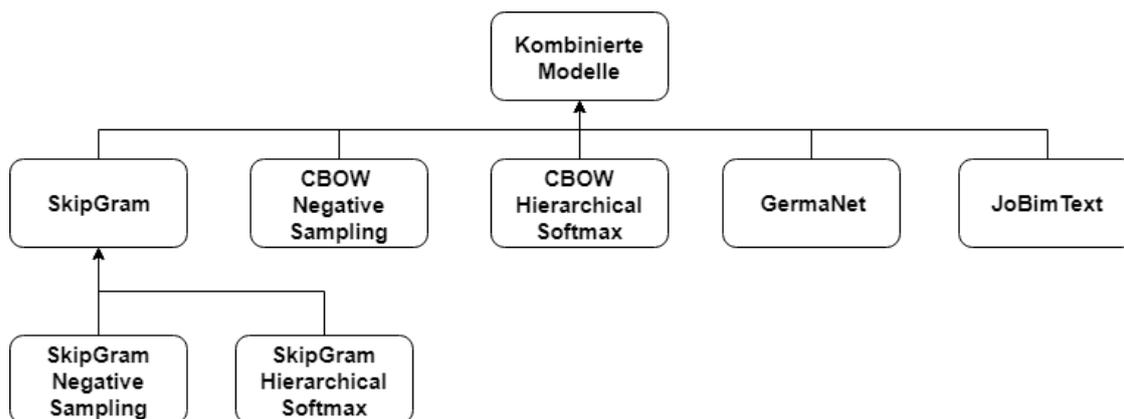


Abbildung 3.6: Aufbau des kombinierten Modells

Um Modelle zu kombinieren, wird von jedem dieser Modelle eine sortierte Liste mit Erweiterungswörtern generiert. Bei einer Liste mit zehn Wörtern wird jedem Wort absteigend eine Zahl von zehn bis eins zugeordnet (siehe Abbildung 3.7). Dies wird für alle Erweiterungslisten durchgeführt. Anschließend werden die Listen zusammengeführt, indem bei gleichen Wörtern die zugeordneten Zahlen addiert werden. Kommt ein Wort nur einmal über alle Ergebnislisten vor, behält es die ursprünglich zugeordnete Zahl. Die so entstandene Liste wird mithilfe der zugeordneten Zahlen sortiert. Dies führt dazu, dass mehrfach vorkommende Wörter besser bewertet werden als solche, die nur in einer Erweiterungsliste vorkommen. Der Stellenwert der Wörter wird ebenfalls berücksichtigt. So ist ein Wort, welches zweimal generiert wurde, aber jeweils an der letzten Stelle der Ergebnisliste platziert wurde, weniger relevant als ein Wort, welches nur in einer Liste vorkommt, aber dafür an der ersten Stelle platziert wurde.

Rang	Modell 1	Modell 2	Modell 3
5	Grundschule	KITA	KiTa
4	Kindertagesstätte	Turnhalle	Kindertagesstätte
3	Schule	Schule	Tagesstätte
2	KiTa	Kindertagesstätte	Sporthalle
1	Turnhalle	Grundschule	Schule

Die besten fünf Ergebnisse sind:

Kindertagesstätte, KiTa, Schule, KITA, Grundschule

Abbildung 3.7: Beispiel Rank-Level-Fusion mit drei Modellen und fünf Erweiterungswörtern.

Dieses Verfahren ist ein Voting-Verfahren aus dem Bereich des Ensemble Learning und wird Ranked-Level-Fusion genannt [24].

### 3.3.5 Filterung

Wie die bisherigen Beispiele vermuten lassen, finden die Modelle nicht nur Erweiterungswörter, welche die Suchergebnisse verbessern. Aus diesem Grund wurde geprüft, ob sich die Menge der Erweiterungswörter, die die Suchergebnisse verschlechtern, verringern lässt. Wichtig ist hier, dass die Generierung von guten Erweiterungswörtern nicht darunter leidet. Hierfür wurden zwei Filteransätze diskutiert und getestet. Es wurde sich gegen die Verwendung von beiden Filteransätzen entschieden. Im Folgenden werden die Filtermethoden beschrieben und diskutiert, warum die Verwendung von beiden verworfen wurde.

Der erste Filter ist ein Part-Of-Speech-Filter. Für diesem Filter wird zunächst für jede Vokabel durch Part-Of-Speech-Tagging ermittelt, welche grammatikalische Funktion es einnimmt. Dadurch wird für jede Vokabel eine Wahrscheinlichkeitsverteilung über alle grammatikalischen Funktionen erstellt. Um das Labeln der Vokabeln zu erleichtern, wurden die Label auf Nomen, Verben, Adjektive und Sonstige verringert. Anschließend wird zu jeder Vokabel die wahrscheinlichste Funktion gespeichert. Ziel des Filters ist es, alle Erweiterungswörter zu entfernen, bei denen die grammatikalische Funktion von der des Suchwortes abweicht. Dies hat den Vorteil, dass zum Erweitern eines Nomens nur weitere Nomen generiert werden.

Der Filter wurde nicht verwendet, da das POS-Tagging meist nur 85% der Label korrekt zuordnet [12].

Das Trainieren des POS-Taggers wurde zudem dadurch erschwert, dass die Dokumente des Hamburger Transparenzportals nicht nur aus Fließtext bestehen. Deshalb wurden häufig Wörter mit dem falschen Label versehen. Durch die falschen Label hat der Filter mehr gute als schlechte Wörter entfernt.

Der zweite Filter ist der Over-Expand-Filter. Mit ihm sollen Erweiterungswörter herausgefiltert werden, welche die Ergebnismenge übermäßig erweitern. Solche Wörter sind fast immer unbrauchbar, da die neu gefundenen Dokumente, die alten in ihrer Masse verschwinden lassen. Zusätzlich wird der Kern der Ursprungssuche verändert (Query Drift) [17]. Um dies zu erreichen, gibt es zwei Ansätze.

Der erste Ansatz testet für alle gefundenen Erweiterungswörter, ob die Anzahl der gefundenen Ergebnisse einen Schwellenwert überschreitet. Der Schwellenwert beträgt das 2,5-fache der Anzahl der Ergebnisse ohne Erweiterungswörter. Problematisch an diesem Ansatz ist, dass die Antwortzeit des Backend-Servers deutlich erhöht und dass nicht alle Erweiterungswörter, die überproportional viele Dokumente finden, schlecht sind. Beispielsweise fügt das Erweiterungswort *Kita*, den 421 gefundenen Dokumenten des Suchworts *Kindergarten*, 1.802 neue Dokumente hinzu. Diese Menge überschreitet den Schwellenwert und würde herausgefiltert werden, obwohl die gefundenen Dokumente die Ergebnisse sinnvoll erweitern und nicht zu Query Drift führen.

Der zweite Ansatz approximiert die Menge an neuen Dokumenten. Hierfür wird die Häufigkeit der Wörter über die gesamte Dokumentmenge betrachtet. In diesem An-

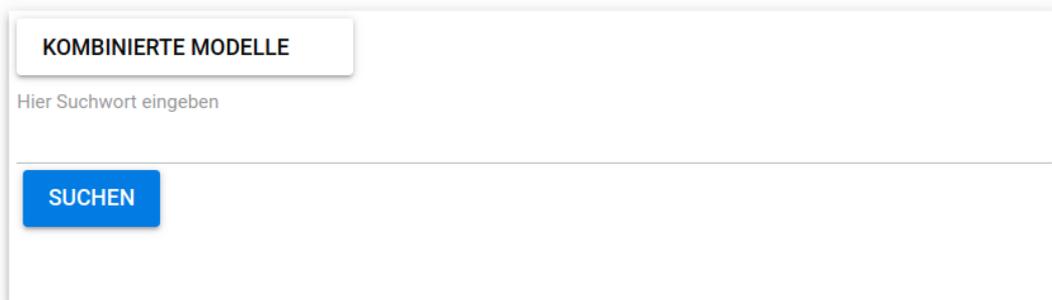
---

satz wird die gleiche Schwellenwertdefinition wie im ersten Ansatz verwendet. Erweiterungswörter werden nur beachtet, wenn sie den Schwellenwert nicht überschreiten. Vorteilhaft an dem Ansatz ist, dass es die Antwortzeit des Backend-Servers schont, während das Filtern von *schlechten* Erweiterungswörtern bestehen bleibt.

Der Over-Expand-Filter wird nicht für das Erweiterungssystem verwendet, da es sinnvoller ist, die Anzahl der hinzugefügten Dokumente durch die Domainexperten evaluieren zu lassen. Dadurch werden Erweiterungswörter wie *Kita* nicht ohne intelligentes Testen ausgeschlossen.

### 3.4 Benutzerschnittstelle für Erweiterungswortevaluation

Die letzte Komponente des Erweiterungssystems ist die Benutzeroberfläche (siehe Abbildung 3.8 und 3.9). Hier wählt der Benutzer Erweiterungswörter aus und vergleicht Ergebnislisten. So werden nur Erweiterungswörter ausgewählt, die neue relevante Dokumente finden. Der Benutzer übernimmt folglich die Rolle eines intelligenten datengetriebenen Filters.



The image shows a search interface with a white background. At the top, there is a button labeled "KOMBINIERT E MODELLE". Below it, there is a text input field with the placeholder text "Hier Suchwort eingeben". At the bottom of the input field, there is a blue button labeled "SUCHEN".

Abbildung 3.8: Startseite

#### 3.4.1 Aufbau der Benutzeroberfläche

Auf der Benutzeroberfläche gibt es verschiedene Bedienelemente, um die Suche zu starten, zum Testen von Erweiterungswörtern, zum Sortieren von der Ergebnisliste und zum Speichern der ausgewählten Erweiterungswörter. Zusätzlich gibt es ein Bedienelement zum Auswählen des verwendeten Modells.

Ein Benutzer kann nach Erweiterungswörtern für ein eingegebenes Wort suchen, indem er entweder auf den Suchen-Button klickt, oder auf Enter drückt. Der Benutzer erhält als Antwort darauf zehn Erweiterungswörter (siehe Abbildung 3.10) und zwei Ergebnislisten (siehe Abbildung 3.11). Die erste Ergebnisliste beinhaltet die ursprünglich gefundenen Dokumente. Die andere zeigt die gefundenen Dokumente an, nachdem Er-

The screenshot shows a search interface with the following elements:

- KOMBINIERTE MODELLE** (Combined Models)
- Suchwort eingeben: **Smartphone**
- Suchergebnisse: **original Suchergebnisse: 160** and **neue Suchergebnisse: 160** (with a 'NEUE DATENSÄTZE HOCH' button).
- Suchergebnisse sind in zwei Spalten dargestellt. Jede Spalte enthält eine Liste von Dokumenten mit Titel und Kontext. Die Suchergebnisse sind mit 'Smartphone' und anderen Mobilgeräten 1.1.2. verknüpft.
- Buttons: **SUCHEN**, **Beitrag**, **Bezirksam**, **Endgerät**, **Gutachten**, **Handy**, **Mobiltelefon**, **SMS**, **Schiff**, **Tablet**, **Ticket**.
- Buttons: **ZEIGE MEHR DATENSÄTZE** (Show more data sets).

Abbildung 3.9: Benutzeroberfläche, nachdem ein Suchwort eingegeben wurde (hier *Smartphone*).

weiterungswörter hinzugefügt wurden. Die nächsten Interaktionsmöglichkeiten stellen die generierten Erweiterungswörter dar.

The screenshot shows a search interface with the following elements:

- KOMBINIERTE MODELLE** (Combined Models)
- Suchwort eingeben: **Wasser**
- Suchergebnisse: **original Suchergebnisse: 2143** and **neue Suchergebnisse: 2143** (with a 'NEUE DATENSÄTZE HOCH' button).
- Suchergebnisse sind in zwei Spalten dargestellt. Jede Spalte enthält eine Liste von Dokumenten mit Titel und Kontext. Die Suchergebnisse sind mit 'Wasser' und anderen Wasserarten verknüpft.
- Buttons: **SUCHEN**, **Abwasser**, **Fernwärme**, **Fließgewässern**, **Gebäude**, **Oberflächenwasser**, **Regenwasser**, **Schiff**, **See**, **Strom**, **Studierendenwerk**.
- Buttons: **ZEIGE MEHR DATENSÄTZE** (Show more data sets).

Abbildung 3.10: Erweiterungswörter für Eingebewort *Wasser*

The screenshot shows a search interface with the following elements:

- KOMBINIERTE MODELLE** (Combined Models)
- Suchwort eingeben: **Wasser**
- Suchergebnisse: **original Suchergebnisse: 2143** and **neue Suchergebnisse: 2143** (with a 'NEUE DATENSÄTZE HOCH' button).
- Suchergebnisse sind in zwei Spalten dargestellt. Jede Spalte enthält eine Liste von Dokumenten mit Titel und Kontext. Die Suchergebnisse sind mit 'Wasser' und anderen Wasserarten verknüpft.
- Buttons: **SUCHEN**, **Abwasser**, **Fernwärme**, **Fließgewässern**, **Gebäude**, **Oberflächenwasser**, **Regenwasser**, **Schiff**, **See**, **Strom**, **Studierendenwerk**.
- Buttons: **ZEIGE MEHR DATENSÄTZE** (Show more data sets).

Abbildung 3.11: Gefundene Dokumente des Hamburger Transparenzportals für Eingabewort *Wasser* (ohne Erweiterungswörter).

Wenn der Benutzer auf eines der Erweiterungswörter klickt, dann passt sich die Ergebnisliste der *neuen Suchergebnisse* an, indem die Ergebnisse einer Suche angezeigt werden, die mit dem ausgewählten Erweiterungswort erweitert wurde (siehe Abbildung 3.13). Die Anzahl der gefundenen Dokumente wird in der Kopfzeile der Ergebnisliste aktualisiert. Pro Dokument werden drei Informationen angezeigt. Ob ein Dokument in den *originalen Suchergebnissen* vorkommt, wird durch die Zahl vor dem Titel und Kontext indiziert. Nicht vorkommende (neue) Dokumente sind mit einer Eins versehen. Im Kontext der Dokumente werden das Eingabewort und die ausgewählten Erweiterungswörter farblich hervorgehoben.

Für den Fall, dass sich die Ergebnismenge verringert, indem der Benutzer ein gespeichertes Erweiterungswort entfernt, oder ein unerwarteter Fehler dafür sorgt, wird eine Warnung angezeigt (siehe Abbildung 3.16). Eine Warnung wird ebenfalls verwendet, um die Problematik mit einer überproportional gewachsenen Dokumentmenge zu lösen. Statt Erweiterungswörter automatisch zu entfernen, wird dem Domainexperten über eine Warnung mitgeteilt, dass die Dokumentmenge überproportional gewachsen ist. So kann individuell entschieden werden, ob das hinzugefügte Erweiterungswort durch die Menge an Dokumenten zu Query-Drift führt, oder relevante Dokumente hinzufügt (siehe Abbildung 3.17).



Abbildung 3.12: Ausgewählte Erweiterungswörter *Endgerät*, *Handy* und *Mobiltelefon* für Eingabewort *Smartphone*.

Sobald ein Benutzer ein Erweiterungswort hinzugefügt hat, lassen sich neue Dokumente sortieren, indem der Benutzer auf den entsprechenden Button klickt (siehe Abbildung 3.13). Die Dokumente werden nach Neuheit sortiert, wobei neue über alten Dokumenten angezeigt werden.

Die letzte wichtige Benutzerinteraktion ist das Speichern von den ausgewählten Erweiterungswörtern. Dies geschieht durch das Klicken auf Speichern. Daraufhin wird ein Dialogfenster geöffnet, in dem der Benutzer das Speichern bestätigen oder abbrechen kann (siehe Abbildung 3.14). Am unteren Bildschirmrand erscheint nach der Entscheidung des Nutzers eine Benachrichtigung über die jeweilige Systemreaktion (siehe Abbildung 3.15).

neue Suchergebnisse: 302

NEUE DATENSÄTZE HOCH

1	<p><b>Titel:</b> Stellungnahme des Senats zum 24. Tätigkeitsbericht Datenschutz des Hamburgischen Beauftragten für Datenschutz und Informationsfreiheit (Drucksache 20/10974)</p> <p><b>Kontext:</b> Lösung wurde eine breitere Nutzung der am Markt verfügbaren <b>Endgeräte</b> insbesondere <b>Smartphones</b> für den;</p>
1	<p><b>Titel:</b> PC-Richtlinie</p> <p><b>Kontext:</b> <b>Endgeräten</b>; Arbeitsplatzrechnern und sonstigen <b>Endgeräten</b> PC-RL vom 01.06.2005 MittVw Seite 7 4 1. Allgemeine Vorschriften;</p>
1	<p><b>Titel:</b> Bezirk Eimsbüttel, Drucksache 20-2029</p> <p><b>Kontext:</b> mit unterschiedlichen <b>Endgeräten</b> nicht nachvollziehen. Die WLAN-Technik funktionierte jeweils;</p>
1	<p><b>Titel:</b> stationäre Endgeräte sowie Zubehör für die Schulen Hamburg</p> <p><b>Kontext:</b> stationäre <b>Endgeräte</b> sowie Zubehör für die Schulen Hamburg durch die Fa. on line GmbH , Berlin; stationäre <b>Endgeraete</b> sowie Zubehoer fuer die Schulen Hamburg;</p>
1	<p><b>Titel:</b> Umsatzsteuer; Entgelt von dritter Seite bei Zahlung eines Gerätebonus durch ein Mobilfunkunternehmen für die Abgabe eines Endgeräts durch den Vermittler eines Mobilfunkvertrags - BFH-Urteil XI R 39/12 vom 16. Oktober 2013</p> <p><b>Kontext:</b> Mobilfunkunternehmen für die Abgabe eines <b>Endgeräts</b> durch den Vermittler eines Mobilfunkvertrags.; Mobilfunkunternehmen für die Abgabe eines <b>Endgeräts</b> durch den Vermittler eines Mobilfunkvertrags - BFH-Urteil; Mobilfunkunternehmen fuer die Abgabe eines <b>Endgeraets</b> durch den Vermittler eines Mobilfunkvertrags - BFH-Urteil XI;</p>
1	<p><b>Titel:</b> V5553 Rahmenvertrag über Betriebs- und Supportleistungen für Endgeräte, lokale Netze und Clientkomponenten, 2. Änderungsvertrag</p> <p><b>Kontext:</b> Rahmenvertrag über Betriebs- und Supportleistungen für <b>Endgeräte</b>, lokale Netze und; V5553 Rahmenvertrag ueber Betriebs- und Supportleistungen fuer <b>Endgeraete</b>, lokale Netze und;</p>
1	<p><b>Titel:</b> Bezirk Bergedorf, Drucksache 20-0832.1</p> <p><b>Kontext:</b> sich die Wartezeiten bei neueren Aufrufanlagen auch auf mobilen <b>Endgeräten</b> darstellen lassen allerdings;</p>

ZEIGE MEHR DATENSÄTZE

Abbildung 3.13: Ergebnisliste mit sortierten Daten, nachdem für das Suchwort *Smartphone* das Erweiterungswort *Endgerät* ausgewählt wurde.

### 3.4.2 Diskussion der Benutzeroberfläche

Für die Benutzeroberfläche wurden viele der vorgestellten Design-Heuristiken befolgt.

Nach der dritten Regel von Ben Shneiderman und der ersten Heuristik von Nielsen wurde dem Design der Benutzeroberfläche ein Ladekreislauf für das Suchen hinzugefügt. Dies ist nötig, da das Generieren von Erweiterungswörtern und die damit verbundene Initialsuche merkbar Zeit beansprucht. Des Weiteren wird die Ergebnisliste ausgegraut, wenn ein Erweiterungswort hinzugefügt oder entfernt wird. Weiterhin wurden Dialogfenster verwendet, um mit dem Nutzer zu kommunizieren (siehe Abbildung 3.14).

Um das Eingeben und Suchen von Erweiterungswörtern intuitiver zu gestalten, wurde

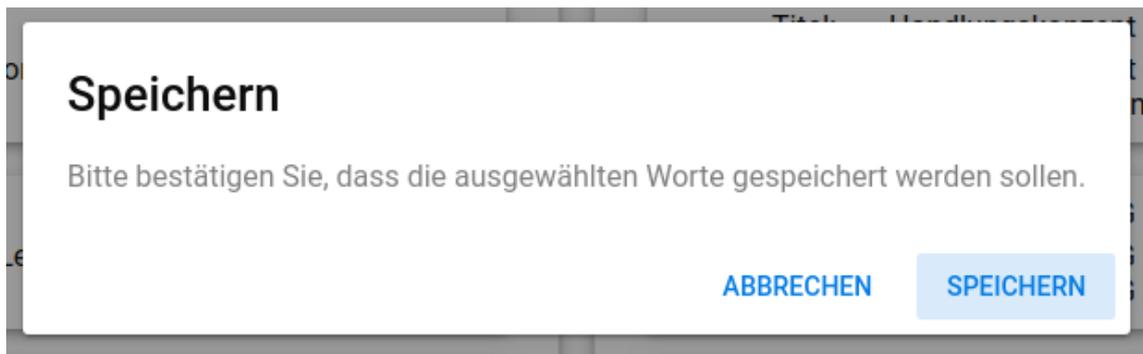


Abbildung 3.14: Dialog-Box zum Speichern



Abbildung 3.15: Benachrichtigung, nachdem ein Benutzer Erweiterungswörter speichert.

das Starten der Suche durch das Bestätigen mit Enter möglich gemacht. Diese Designentscheidung deckt sich mit der zweiten Regel von Shneiderman und der siebten Heuristik von Nielsen.

Die sechste Regel von Shneiderman und dritte Heuristik von Nielsen betrifft das Rückgängigmachen von Benutzeraktionen. Dies wurde durch das Entfernen von Erweiterungswörtern umgesetzt. Außerdem können einmal gespeicherte Erweiterungswörter bei einer erneuten Suche wieder entfernt werden.

Der ersten Regel und vierten Heuristik folgend, wurden die Farben der Benutzeroberfläche gewählt. Buttons für Systeminteraktionen sind blau, ausgewählte Erweiterungswörter grün und nicht ausgewählte rot. Diese Farben wurden aufgrund ihrer unterbewussten Assoziationen gewählt. Blau vermittelt Sachlichkeit und bezeichnet die technische Interaktion. Auch wenn Rot als Warnfarbe gilt und Benutzer im ersten Moment davon abhält auf Erweiterungswörter zu klicken, ist es das beste Gegenstück zu Grün. Grün und Rot wurden gewählt, da sie am verständlichsten ausgewählte und nicht ausgewählte Erweiterungswörter kennzeichnen. Dies ist wichtig, da Benutzer möglicherweise ein Eingabewort mehr als einmal testen. Dann soll auf einen Blick erkennbar sein, welche ausgewählt sind und welche nicht. Dies wäre bei anderen Farbkombinationen nicht gegeben.

Die achte Heuristik von Nielsen besagt, dass eine Benutzeroberfläche nur die relevanten Informationen beinhalten sollte. Dies wurde ebenfalls umgesetzt. Hierfür wurden auf begleitende Informationen, wie die IDs der Dokumente, verzichtet. Diese würden für den Benutzer keine Hilfe für seine Aufgabe darstellen.

Zuletzt wurde die neunte Heuristik für das Oberflächendesign herangezogen. Sie mahnt,

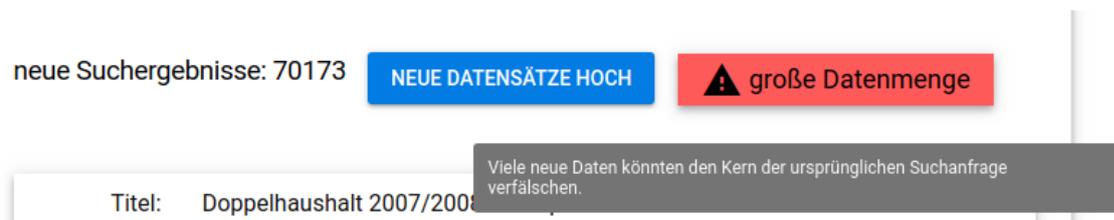
den Benutzer mit natürlich-sprachlichen Fehlermeldungen oder Warnungen zu versorgen. Diese wurden als Tooltip für die Warnungen implementiert (siehe Abbildung 3.17). Sie erscheinen, wenn der Benutzer den Cursor auf die Warnung bewegt.

Während der Entwicklung der Benutzeroberfläche wurden Heuristiken verletzt. So wurde keine Eingabeevaluierung implementiert, welche die Benutzereingaben auf Korrektheit überprüft und so Fehler vermeidet. Dies wurde von Nielsen (fünfte Heuristik), sowie Shneiderman (vierte Regel) gefordert. Da die Benutzeroberfläche im Rahmen der Bachelorarbeit nur in Verbindung mit Benutzertests verwendet wird, in denen die Suchwörter vorgegeben sind, schadet es der Usability nicht. Außerdem regt Shneiderman in seiner zehnten Regel den Benutzern eine Dokumentation zur Verfügung zu stellen. Auch dies wurde zurückgestellt, da die Benutzertests einzeln stattfinden und die Testpersonen dem Versuchsleiter Fragen zur Bedienung stellen können.



neue Suchergebnisse: 739 NEUE DATENSÄTZE HOCH ⚠ verringerte Datenmenge

Abbildung 3.16: Warnung für verringerte Daten



neue Suchergebnisse: 70173 NEUE DATENSÄTZE HOCH ⚠ große Datenmenge

Titel: Doppelhaushalt 2007/2008

Viele neue Daten könnten den Kern der ursprünglichen Suchanfrage verfälschen.

Abbildung 3.17: Beispiel für den Tooltip einer Warnung

### 3.5 Integrieren der Applikation in ein Produktionsumfeld

Um das System zum Erweitern einer Wörterbuchdatei sinnvoll einsetzen zu können, wird im folgenden Abschnitt ein Abriss über die Systemvoraussetzungen, sowie Vorschläge zur Verwendung in der Praxis gegeben.

Um das System so betreiben zu können, wie es für das Erstellen der Bachelorarbeit getan wurde, muss die Benutzeroberfläche durch Vue/Quasar verwaltet, das Backend in Python geschrieben und Solr als Suchmaschine verwendet werden. Jedoch sind alle drei Systembereiche Server (siehe Abbildung 3.18). Folglich können Systembereiche gegen andere ausgetauscht werden, sofern dies das Integrieren in ein bestehendes System erleichtert. So könnte beispielsweise die Benutzeroberfläche statt mit Vue auch mit jQuery implementiert werden, solange die gleichen Schnittstellen im Python-Backend mit gleichen Datei-Paketen angesprochen werden. Zu beachten ist noch, dass in Solr das Verwenden von Managed Synonyms aktiviert werden muss.

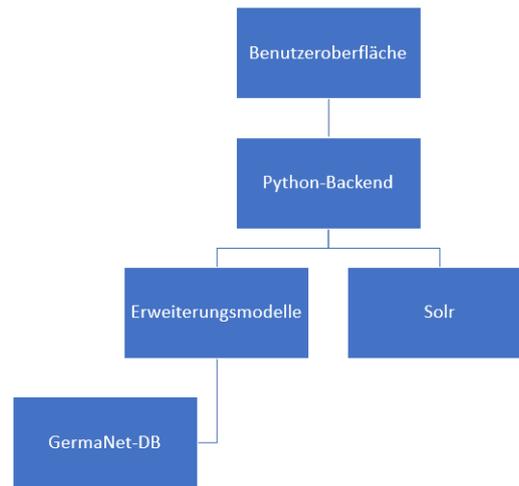


Abbildung 3.18: Server-Struktur

Bei der Verwendung von GermaNet muss beachtet werden, dass das Wörterbuch erworben werden muss und entweder in eine Datenbank importiert werden, oder über die erworbenen XML-Dateien von GermaNet die Daten ausgelesen werden.

Da mit dem System die Erweiterungswörter in der Synonym-Datei von Solr auf einem aktuellen Stand gehalten werden sollen, sollte die Überprüfung regelmäßig stattfinden. Der Abstand zwischen den Überprüfungen muss individuell eingeschätzt werden und hängt mit der Häufigkeit der Änderung des Korpus zusammen. Als Startwert wäre halbjährlich empfehlenswert. Bevor die Überprüfung durch Testpersonen beginnen kann, müssen verschiedene Vorbereitungen getroffen werden. Zuerst müssen die Word Embedding Modelle und JoBimText auf den aktuellen Dokumenten trainiert werden. Danach muss die Menge der Suchwörter bestimmt werden. Hierfür eignet sich die Schnittmenge aus den Wörtern, nach denen Benutzer häufig suchen und solchen, die oft in den Dokumenten des Suchsystems vorkommen. Auch hier sind die Schwellenwerte individuell zu bestimmen.

Die Oberflächenevaluation zeigte, dass es keine Anforderungen an die Domainexperten gibt, da die Lernphase sehr kurz ist. In den meisten Fällen wurden nicht mehr als vier Suchanfragen benötigt, um sich an die Aufgabe zu gewöhnen.

Um Bewerten zu können, ob das kombinierte Modell relevante Erweiterungswörter für Suchanfragen findet und ob die Benutzeroberfläche Domainexperten beim Auswählen dieser angemessen unterstützt, wurde das System von Testpersonen ausprobiert. Im nächsten Kapitel wird der Ablauf des Experimentes beschrieben.



## 4 Aufbau des Experimentes

Im folgenden Abschnitt wird beschrieben, wie das Experiment mit den Testpersonen durchgeführt wurde. Es wird gezeigt, wie die Usability der Benutzeroberfläche bewertet wurde. Zudem wird erklärt, wie die Daten zum Evaluieren der Modelle des kombinierten Modells gesammelt werden. Die Experimente mit Testpersonen wurden durchgeführt, um das Erweiterungssystem auf der Basis von quantitativen Daten bewerten zu können.

Die Aufgabe der Testpersonen war für 16 Suchwörter mit Hilfe der Benutzeroberfläche gute Erweiterungswörter auszuwählen. Diese zehn besten Erweiterungswörter des kombinierten Modells wurden den Testpersonen angezeigt. Die 16 Suchwörter bestehen aus acht festgelegten und acht zufällig ausgewählten Wörtern. Die zufälligen Wörter stammen aus der Schnittmenge aus Querys, die von mindestens zehn Benutzern des Hamburger Transparenzportals gesucht wurden und den 4000 häufigsten Wörtern<sup>1</sup> in den Dokumenten. Um zu entscheiden, ob ein vorgeschlagenes Wort eine sinnvolle Erweiterung darstellt, sollten die Testpersonen bewerten, ob die neu gefundenen Datensätze zu denen der originalen Suchergebnisse passen. Anschließend füllten die Testpersonen den System-Usability-Scale-Fragebogen (SUS-Fragebogen) aus und sprachen mit dem Versuchsleiter über die Funktionalität und das Design des Erweiterungssystems. Die Bemerkungen aus dem Einzelinterview wurden auf dem Fragebogen (siehe Abbildung 4.1) festgehalten.

Der Vorteil von dem Einzelinterviewverfahren ist, dass man sich im Gespräch mit den Testpersonen befindet. So hat man die Möglichkeit, bei Unklarheiten nachzufragen und so ein tieferes Verständnis für das Empfinden der Testperson beim Verwenden der Benutzeroberfläche zu erhalten. Des Weiteren gibt es die Möglichkeit, dass beim Beobachten der Interaktion zwischen Benutzer und Oberfläche weitere Verbesserungsmöglichkeiten auffallen. Der Nachteil eines Interviews ist, dass Gespräche schwer quantifizierbar sind. Deshalb wurden von den Testpersonen nach dem Durchführen der Testaufgaben ein SUS-Fragebogen ausgefüllt. Damit lassen sich quantifizierbare Aussagen über alle Testpersonen mitteln [33].

Während die Testperson verschiedene Erweiterungswörter evaluiert, werden die Benutzerinteraktionen im Hintergrund in einer Log-Datei (siehe Abbildung 4.2) aufgezeichnet. Dies ermöglicht das Evaluieren der Modelle zum Finden von Erweiterungswörtern. Neben diesen und den Modellen, von denen sie gefunden wurden, wird ebenfalls aufgezeichnet, welche Erweiterungswörter vom Benutzer gespeichert werden. Es wird eben-

---

<sup>1</sup>Beim Finden der 4000 Wörter wurden Stoppwörter nicht berücksichtigt.

TestpersonenNr: \_\_\_\_\_

Datum: \_\_\_\_\_

## System Usability Scale

### Aufgabe:

Du bist ein Mitarbeiter, der für sein Unternehmen das halbjährliche Wort-Erweiterungs-Update durchführt. Hierfür testest du, ob das Erweiterungssystem, für eine Liste von Wörtern, gute Erweiterungen findet. Gute Erweiterungen zeichnen sich dadurch aus, dass sie die originale Ergebnismenge um Datensätze erweitern, die ähnlich zu den ursprünglich gefundenen sind.

### Testwörter:

Smartphone, Kindergarten, UHH, StVO, bremsen, Verletzung, verschmutzt, schwierig, Baukosten, Behinderung, Kirchwerder, Bescheid, Umbau, Heimfeld, Personal, Grundwasser

**Anleitung:** Für jede der folgenden Fragen die Antwortmöglichkeit markieren, welche am besten zu der Benutzeroberfläche passt.

		<i>Trifft nicht zu</i>			<i>Trifft voll zu</i>	
1.	Ich kann mir sehr gut vorstellen, das System regelmäßig zu nutzen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.	Ich empfinde das System als unnötig komplex.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	Ich empfinde das System als einfach zu nutzen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.	Ich denke, dass ich technischen Support brauchen würde, um das System zu nutzen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.	Ich finde, dass verschiedene Funktionen des Systems gut integriert sind.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.	Ich finde, dass es im System zu viele Inkonsistenzen gibt.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	Ich kann mir vorstellen, dass die meisten Leute das System schnell zu beherrschen lernen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8.	Ich empfinde die Bedienung als sehr umständlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.	Ich habe mich bei der Nutzung des Systems sehr sicher gefühlt.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.	Ich musste eine Menge Dinge lernen, bevor ich mit dem System arbeiten konnte.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Bemerkungen:

Abbildung 4.1: Ein leerer Fragebogen

falls überprüft, wie viele Erweiterungswörter ausgewählt wurden und ob Erweiterungswörter vom kombinierten Modell gefunden worden sind. Mit den aufgezeichneten Daten kann in der Evaluation nachvollzogen werden, wie viele Erweiterungswörter von den Testpersonen ausgewählt wurden und welche der verwendeten Erweiterungsmo-

delle die besten Erweiterungswörter generiert haben.

```

DEBUG:root:FOUND {'Handy': ['CBOW_HS', 'CBOW_NS', 'SkipGram'], 'Mobiltelefon': ['CBOW_HS', 'CBOW_NS',
'SkipGram'], 'Ticket': ['CBOW_HS', 'CBOW_NS'], 'SMS': ['CBOW_HS', 'CBOW_NS'], 'Tablet': ['CBOW_HS',
'SkipGram'], 'Bezirksamt': ['Jobim'], 'Gutachten': ['Jobim'], 'Schiff': ['CBOW_HS'], 'Endgerät':
['SkipGram'], 'Bett': ['CBOW_HS']} for Smartphone
INFO:werkzeug:127.0.0.1 - - [09/Aug/2019 11:04:44] "GET /getSynonyms?word=Smartphone&model=Combined
HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [09/Aug/2019 11:04:44] "GET /solrSearch?word=Smartphone HTTP/1.1" 200 -
DEBUG:root:TESTED ['Handy'] on Smartphone
INFO:werkzeug:127.0.0.1 - - [09/Aug/2019 11:05:30] "GET /testSynonym?
query=Smartphone&synList=Handy&model=Combined HTTP/1.1" 200 -
DEBUG:root:TESTED ['Handy', 'Mobiltelefon'] on Smartphone
INFO:werkzeug:127.0.0.1 - - [09/Aug/2019 11:06:16] "GET /testSynonym?
query=Smartphone&synList=Handy,Mobiltelefon&model=Combined HTTP/1.1" 200 -
DEBUG:root:TESTED ['Handy'] on Smartphone
INFO:werkzeug:127.0.0.1 - - [09/Aug/2019 11:07:10] "GET /testSynonym?
query=Smartphone&synList=Handy&model=Combined HTTP/1.1" 200 -
DEBUG:root:SAVING {'Handy': ['CBOW_HS', 'CBOW_NS', 'SkipGram']} for Smartphone found by Combined

```

Abbildung 4.2: Ein Beispiel eines Log-Eintrages für das Eingabewort *Smartphone*. Getestet und gespeichert wurde das Erweiterungswort *Handy*

Die Ergebnisse der Analyse der Log-Dateien, sowie der SUS-Fragenbögen und der qualitativen Einzelgespräche werden im nächsten Kapitel vorgestellt und diskutiert.



## 5 Ergebnisse

In diesem Abschnitt werden die erhobenen Daten durch die Testpersonen vorgestellt und diskutiert. Dies umfasst sowohl die Ergebnisse der Oberflächenevaluation, als auch die Log-Dateien. Mit Hilfe der Log-Dateien können die verwendeten Modelle miteinander verglichen werden. Betrachtet wird die Menge und Wahrscheinlichkeit mit der Erweiterungswörter ausgewählt werden, sowie die Übereinstimmung zwischen den Testpersonen.

### 5.1 Oberflächenevaluation

#### 5.1.1 Pilotstudie

Im Vorfeld zu der eigentlichen Oberflächenevaluierung wurde eine Pilotstudie durchgeführt. Ziel dieser war, das vorzeitige Aufdecken von wichtigen Designverbesserungen und eventuellen Programmfehlern. Die hierfür ausgewählte Testperson verfügt über Kenntnisse im UI-Design.

Es wurden eine Reihe von Verbesserungsmöglichkeiten gefunden. Die wichtigsten betreffen die Responsiveness der Benutzeroberfläche. Es wurde eine Sicherheitsabfrage für den Speicherbutton hinzugefügt. So sollen Fehler durch den Nutzer vorgebeugt werden (siehe Abbildung 3.14). Außerdem wird die rechte Ergebnisliste ausgegraut, wenn ein Wort zur Erweiterung getestet wird. Dies hat das Ziel zu zeigen, wann die Daten vom Backend-Server fertig verarbeitet wurden, wenn das Ergebnis sich nicht von dem Original unterscheidet. Um die Evaluation der Erweiterungswörter zu unterstützen, werden nicht die neuen Dokumente aller hinzugefügten Erweiterungswörter angezeigt, sondern nur die Dokumente, welche durch das letzte Erweiterungswort hinzugefügt wurden.

Zusätzlich wurden noch verschiedene kleine Änderungen vorgenommen, um die Konsistenz der Benutzeroberfläche zu verbessern.

Im Rahmen der Pilotstudie wurde auch der SUS-Fragebogen ausgefüllt. Die Benutzeroberfläche erreichte zu diesem Zeitpunkt 72,5 Punkte und lag damit unter den zu erreichenden 75 Punkten.

#### 5.1.2 Finale Evaluation

Die Ergebnisse der Oberflächenevaluation, umfassen die Bemerkungen der Testpersonen aus den Einzelinterviews, sowie die Auswertung der SUS-Testbögen. Zur Evaluation

---

wurden 20 Testpersonen befragt. Für die gestellte Aufgabe benötigten sie durchschnittlich eine Stunde. Von diesen waren zehn Testpersonen weiblich und zehn männlich. Die meisten der Testpersonen waren jünger als 30 (siehe Diagramm 5.1.2). Zudem erreichten mindestens die Hälfte aller Testpersonen die allgemeine Hochschulreife (siehe Diagramm 5.1.2).

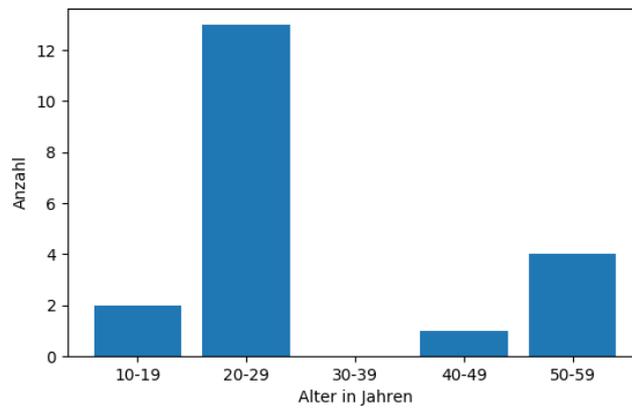


Abbildung 5.1: Altersverteilung der Testpersonen.

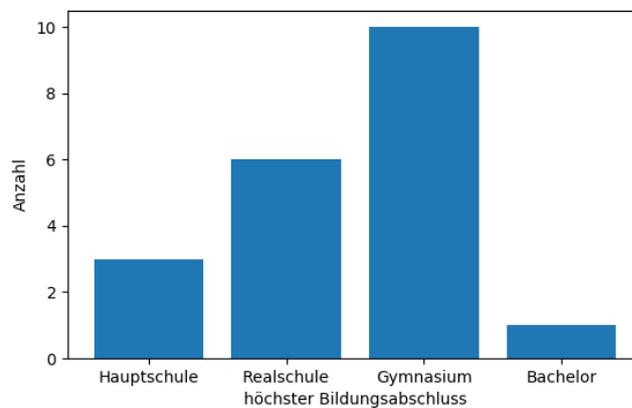


Abbildung 5.2: Höchster Abschluss der Testpersonen.

### Ergebnisse der Einzelinterviews

In den Einzelinterviews wurden verschiedene Bemerkungen an der Interaktion, sowie dem Design der Benutzeroberfläche angesprochen. Die erste betrifft die Kontexte der

Dokumente. Diese sind für einige Dokumente zu klein, um erkennen zu können, in welchem Kontext ein markiertes Wort steht. Die nächste Bemerkung betrifft Hotkeys, welche die Bediengeschwindigkeit erhöhen sollen. Geeignete Funktionen für Hotkeys wären das Sortieren von neuen Dokumenten und das Speichern. Die Usability kann noch erhöht werden, indem das Suchwort, die Erweiterungswörter und die Ergebnislisten zurückgesetzt werden. Dies vermeidet auch Bedienfehler.

Eine weitere Bemerkung betrifft die Tooltips, welche Hinweise für die Warnungen liefern. Diese wurden von den Testpersonen häufig nur durch Zufall gefunden. Deshalb muss deutlich gemacht werden, dass weitere Informationen durch das Bewegen der Maus auf die Warnung zu erhalten sind. Eine häufige Bemerkung war, dass die Null und Eins schlechte Indikatoren für neue und alte Datensätze sind. Besser wäre eine Kennzeichnung durch ein Icon oder eine farbliche Hervorhebung. Die letzte Bemerkung regte an, den Button für das Sortieren der neuen Datensätze um ein Pfeil-Icon zu erweitern. Dies könnte die Intuitivität des Buttons erweitern.

Neben Bemerkungen, die Verbesserungen anstoßen, gab es auch solche, die das Design oder Funktionalität lobten. In mehreren Fällen wurde die einfache Verwendung des Systems hervorgehoben, nachdem man in das System eingeführt wurde. Außerdem wurde das minimalistische Design und die daraus resultierende intuitive Anwendung von den Testpersonen gelobt.

### Ergebnisse der SUS-Fragebögen

Im Folgenden werden die Bewertungen der einzelnen Fragen in der absteigenden Reihenfolge ihrer Abweichung diskutiert.

Fragennummer	Durchschnittswert	Abweichung
1	4,11	0,89
2	1,37	0,37
3	4,42	0,58
4	1,95	0,95
5	4,47	0,53
6	1,47	0,47
7	4,68	0,32
8	1,32	0,32
9	4,32	0,68
10	2,26	1,26
Gesamtbewertung	84,21	

Tabelle 5.1: Die Tabelle zeigt die Durchschnittswerte der einzelnen Fragen des SUS-Fragebogens, sowie die Abweichung vom Optimalwert und den Durchschnitt der Gesamtbewertung. Der Optimalwert für gerade Fragen ist 1 und für ungerade 5.

Die Testperson, welche die Pilotstudie durchführte, vergab bei der zweiten Testdurchführung 84 Punkte, was für eine deutliche Verbesserung der Usability der Benutzerober-

fläche spricht. Die 72,5 Punkte der Pilotstudie wurde auch von dem Durchschnittswert aller Testpersonen (84,21) deutlich übertroffen (siehe Tabelle 5.1). Auch der Schwellenwert von 75 Punkten wurde überschritten und folglich erscheint die Benutzeroberfläche als eine sinnvolle Unterstützung für das manuelle Auswählen von Erweiterungswörtern. Von den 20 Testpersonen wurden dreimal weniger als 75 Punkte vergeben.

In der zehnten Frage wurde gefragt, wie viel die Testperson lernen musste, um die Benutzeroberfläche zum Lösen der Aufgabe zu verwenden. Bei dieser Frage wurde die größte Abweichung vom Optimalwert gemessen. Dies ist aber bei der Komplexität der Aufgabe wenig überraschend, da die Testperson mit Dokumenten außerhalb des Alltäglichen konfrontiert wird, und dann bewerten soll, ob neu gefundene Dokumente gute Erweiterungen für die bereits vorhandenen sind. Die meisten Testpersonen hatten sich nach den ersten vier Suchwörtern an die Aufgabe gewöhnt und haben so eine Lernphase abgeschlossen. Anschließend stellte die Verwendung des Systems für die Testpersonen keine Herausforderung mehr dar. Die vierte Frage betrifft das Benötigen von technischem Support. Die erhöhte Abweichung liegt, wie sich in den Gesprächen herausstellte, an der behördlichen Sprache, in der die Dokumente des Hamburger Transparenzportals verfasst sind. Dazu kommt, dass die Tooltips der Warnungen häufig nicht gefunden wurden, was zu technischen Fragen führte. Der letzte zu berücksichtigende Faktor ist das Sortieren der neuen Dokumente. Hier kam es zu Verwirrungen, da neue Dokumente gefunden wurden, sie aber nicht unter den 100 relevantesten Ergebnissen waren und so nicht bei der Sortierung auftauchten. Die erste Frage betrifft die Vorstellung, das System regelmäßig zu nutzen. Eine mögliche Erklärung für die Abweichung vom Optimum ist monotone Arbeit, welche die Art der Aufgabe mit sich bringt. Dies lässt sich schwer durch die Benutzeroberfläche verbessern. Die neunte Frage zielt auf das Sicherheitsgefühl der Testpersonen beim Verwenden der Benutzeroberfläche ab. Der Grund für das Gefühl von Unsicherheit lag bei manchen Testpersonen an der mangelnden Vertrautheit mit Softwaresystemen im Allgemeinen. Dazu kommen die Rückfragen, welche bereits bei der vorherigen Frage besprochen wurden.

Bei der Frage, ob das System einfach zu verwenden sei (Frage 3), gab es gemischte Antworten. Dies liegt zum einen an der unterschiedlichen Vertrautheit mit Softwaresystemen, aber auch an den Faktoren, die in Verbindung mit der Frage nach technischen Support beleuchtet wurde. Dem entgegen wirkt das minimalistische Design, aber auch das Anbieten von unterstützenden Funktionen, wie ein Shortcut zum Suchen oder dem Sortieren von Dokumenten. Bei den Bewertungen zur Integration von Funktionen (Frage 5) kritisierten manche Testpersonen, dass weitere Shortcuts fehlen. Aber auch die Sortierfunktion, die nicht alle neuen Dokumente, sondern nur die ersten 100 berücksichtigt, wurde als Kritik genannt. Bei Inkonsistenzen innerhalb des Systems (Frage 6), war die Festlegung auf Kritikpunkte schwer. Mit dem Design waren die meisten Testpersonen zufrieden. Der einzig negative angemerkt Punkt war die Farbcodierung der Buttons für die Erweiterungswörter.

---

Besonders gut bewerteten die Testpersonen, die Komplexität der Benutzeroberfläche (Frage 2), sowie die Bedienfreundlichkeit (Frage 8). Die Testpersonen begründeten dies mit dem minimalistischen Design und der daraus entstehenden Klarheit über die Bedienoptionen. Wie bereits im Abschnitt 3.4 diskutiert wurde, war dies unter anderem ein Fokus bei dem Design der Benutzeroberfläche. Auch wenn die Menge der Dinge, welche gelernt werden musste, von vielen Testpersonen als größer angesehen wurde, gaben die meisten ebenfalls an, dass die Benutzeroberfläche schnell zu beherrschen war (Frage 7). Dies wird durch das minimalistische Design, sowie der impliziten Kommunikation mit dem Benutzer durch Ladekreisel und dem Ausgrauen gefördert.

### **Mögliche Erweiterungen der Benutzeroberfläche**

Wie in Abschnitt 3.4.2 besprochen, wurden einige Funktionen nicht umgesetzt. Für den Einsatz in einer Produktivitätsumgebung wären einige Funktionen sinnvoll. Dazu gehört das Erstellen einer Ansicht mit einer Dokumentation. Wichtig ist auch das Evaluieren der Benutzereingaben um Fehlern vorzubeugen. Da eine bestehende lexikalische Ressource erweitert wird, sind in dieser Erweiterungswörter enthalten, die nicht unter den gefundenen Erweiterungswörtern sind. Deshalb sollten diese auf der Benutzeroberfläche angezeigt werden, um über alle bereits hinzugefügten Erweiterungswörter für ein Suchwort zu informieren. Da Erweiterungswörter von Menschen evaluiert werden, wäre es sinnvoll diesen zu ermöglichen Erweiterungswörter einzugeben und zu testen. So können mögliche Synonyme, die der Domainexperte kennt, ausprobiert werden.

## **5.2 Modellevaluation**

In diesem Abschnitt werden die Ergebnisse der Modellevaluation vorgestellt. Es wird betrachtet, wie viele Erweiterungswörter von den Testpersonen gefunden wurden und von welchen Modellen diese generiert wurden. Dann wird die Qualität der Erweiterungswörter zwischen den Modellen verglichen. Zuletzt wird betrachtet, wie sehr sich die ausgewählten Erweiterungswörter zwischen Testpersonen unterscheiden.

### **5.2.1 Modellevaluation durch Testpersonen**

Zum Erstellen der folgenden Statistiken wurden 168 Suchwörter getestet. Insgesamt wurden 1680 Erweiterungswörter evaluiert. Diese sind das Ergebnis der Rank-Level-Fusion der verwendeten Erweiterungsmodelle. Für die Erweiterungswörter, welche für die acht Suchwörter, die für alle Testpersonen gleich waren, generiert wurden, wurde jeweils die häufigste Kombination aus Erweiterungswörtern berücksichtigt. So wurde für jedes Suchwort die jeweils wahrscheinlichste Auswahl an Erweiterungswörtern ausgewählt und die folgenden Statistiken würden nicht unnötig verfälscht. Anders wäre es, wenn alle ausgewählten Erweiterungswörter unabhängig von der Häufigkeit ihrer Auswahl berücksichtigt werden würden.

Suchwort mit EW pro Suchwörter	100,00 %	ausgewählte EW pro Suchwort	1,559
Suchwort mit ausgewählten EW pro Suchwörter	70,24 %	ausgewählte EW pro Suchwort mit ausgewählten EW	2,220

Tabelle 5.2: Links; Häufigkeit der gefundenen und ausgewählten Erweiterungswörtern. Rechts; Durchschnittlich ausgewählte EW.

Das kombinierte Modell konnte für alle Suchwörter zehn Erweiterungswörter finden. Von diesen wählten die Testpersonen, in sieben von zehn Fällen, mindestens ein Erweiterungswort aus (siehe Tabelle 5.2). Diese Zahl könnte größer sein, wenn man beim Erstellen der Suchwörter alle Ortsnamen herausfiltert. Die Testpersonen haben in nur einem Fall<sup>1</sup> Ortsnamen erweitert. Das Testen von Ortsnamen ist bei dem kombinierten Modell zu vernachlässigen, da das Modell nur weitere Ortsnamen zu einem Ort findet (siehe Abbildung 5.3). Grund hierfür ist, dass GermaNet keine Synsets für die Ortsnamen hat und CBOW, SkipGram und JoBim nur andere Ortsnamen generieren. Diese sind nicht hilfreich. Wenn ein Benutzer nach einem bestimmten Ort sucht, sollten keine Dokumente gefunden werden, die einen anderen Ort betreffen. Von den 168 Suchwörtern waren 28 Ortsnamen. Würden diese mit Hilfe einer Liste herausgefiltert, dann wären für 84.29% aller Suchwörter, mindestens ein Erweiterungswort ausgewählt worden.

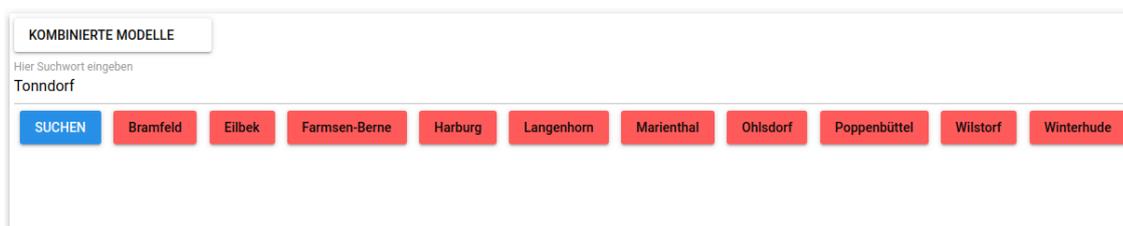


Abbildung 5.3: Ergebnisliste für das Suchwort *Tonndorf*.

Im Durchschnitt haben die Testpersonen 1,5 Erweiterungswörter ausgewählt. Für den Fall, dass eine Testperson mindestens ein Erweiterungswort ausgewählt hat, beträgt der Durchschnittswert der ausgewählten Erweiterungswörter mindestens zwei. Da die zu erweiternde Synonymdatei pro Suchwort durchschnittlich nur ein Erweiterungswort speichert, ist das Hinzufügen von zwei Erweiterungswörtern durch das Erweiterungssystem zufriedenstellend.

Bei der Tabelle ist zu beachten, dass ein Erweiterungswort meist von mehr als einem Modell gefunden wird. Deshalb addieren sich die Spalten nicht zu 100% auf.

Die CBOW-Modelle finden beide etwa 50% aller Erweiterungswörter. Interessant ist, dass beim Modell mit Negative Sampling (CBOW NS) die Menge der ausgewählten Erweiterungswörter mit 20% höher ist. Folglich liefert dieses Modell häufiger gute Erweiterungswörter als das Modell, welches hierarchischen Softmax (CBOW HS) verwendet.

<sup>1</sup>Farmsen wurde um Berne erweitert

	EW vom Modell pro EW	ausgewählte EW vom Modell pro ausgewählte EW	ausgewählte EW vom Modell pro EW vom Modell
CBOW HS	52,26%	53,44%	15,95%
CBOW NS	55,06%	73,28%	20,76%
SkipGram	44,76%	61,45%	21,41%
GermaNet	6,31%	16,79%	41,51%
JoBim	30,71%	9,92%	5,04%
Kombiniert	100,00%	100,00%	15,59%

Tabelle 5.3: Links zeigt den Anteil eines Modells an allen Erweiterungswörtern. In der Mitte sieht man den Anteils eines Modells an ausgewählten Erweiterungswörtern. Recht stellt das Verhältnis zwischen ausgewählten und gefundenen Erweiterungswörtern dar.

Dies zeigt auch das Verhältnis zwischen ausgewählten und allen Erweiterungswörter des Modells (siehe rechts in Tabelle 5.3).

Wie bereits beim Trainieren des Modells vermutet, sind die Ergebnisse des JoBim-Modells im Vergleich zu den anderen wenig beeindruckend. Es wurden nur neun Prozent aller gefundenen Erweiterungswörter ausgewählt, obwohl 30% aller Erweiterungswörter vom JoBim-Modell gefunden wurde. Dies liegt vermutlich daran, dass die Menge der Trainingsdokumente nicht ausreichte, um das Modell ausreichend zu trainieren. Dies zeigt die Willkür der gefundene Erweiterungswörter (siehe Abbildung 5.4).



Abbildung 5.4: Ergebnisliste für das Suchwort *StVO* mit Erweiterungswörtern, welche durch das JoBim-Modell gefunden wurden.

Das SkipGram-Modell, welches zwei kombinierten SkipGram-Modellen (SkipGram NS, SkipGram HS) besteht, erreichte das zweitbeste Verhältnis zwischen ausgewählten und gefundenen Erweiterungswörtern des Modells. Dies ist auch der Fall für das Verhältnis zu allen ausgewählten Erweiterungswörtern. Etwa 60% aller ausgewählten Erweiterungswörter wurden durch das SkipGram-Modell gefunden.

Obwohl GermaNet den geringsten Anteil an den gefundenen Erweiterungswörtern hat, übertrifft es das JoBim-Modell mit der Menge der ausgewählten Erweiterungswörter. Zudem werden von GermaNet die qualitativ hochwertigsten Erweiterungswörter geliefert. Dies belegt das Verhältnis zwischen ausgewählten und generierten Erweiterungswörtern des Modells, welches fast doppelt so hoch ist wie das Verhältnis vom zweitplatzierten Modell SkipGram. Dieses Ergebnis ist zu erwarten, da in GermaNet manuell erstellte Synonymlisten eingepflegt sind. Problematisch ist nur, dass für mache, Suchwör-

ter wenige Synonyme definiert sind oder keine Synsets existieren, sodass nicht für alle Suchworte mit GermaNet Erweiterungswörter gefunden werden können. Beispielsweise wird für *Kindergarten* nur das Erweiterungswort *Schulkindergarten* gefunden.

Durch das Sortieren der Synonyme eines Synsets, nach der Häufigkeit innerhalb der Dokumente des Hamburger Transparenzportals, wurden möglicherweise Erweiterungswörter begünstigt, die zu überproportionalen Erweiterungen eines Suchworts führen. Diese wären folglich nicht von den Testpersonen ausgewählt worden. Es bleibt zu prüfen, ob eine andere Methode zum Erstellen eines Rankings ein besseres Ergebnis liefert.

Jedoch zeigt das Ergebnis auch, dass selbst die Synonyme aus GermaNet nicht mal in der Hälfte der Fälle geeignete Erweiterungswörter sind. Dies liegt zum einen an unterschiedlichen Wortbedeutungen, aber auch an dem spezifischen Kontext, für den Erweiterungswörter gefunden werden sollen. Dies zeigt, dass fertige Wörterbuchansätze allein nicht ausreichen, um in spezifischen Kontexten als Erweiterungswörterbuch zu dienen.

Für das kombinierte Modell werden nur etwa 15% aller gefundener Erweiterungswörter ausgewählt. Somit unterliegt es allen Erweiterungsmodellen, außer JoBimText, im Finden von guten Erweiterungswörtern. Mögliche Erklärungen sind zum einen, dass das kombinierte Modell neben den wenigen meist guten Erweiterungswörtern von GermaNet auch einige schlechte Erweiterungswörter von JoBimText präsentiert hat. Dies führte dazu, dass das Verhältnis von ausgewählten zu gefundenen Erweiterungswörtern abnahm. Ein Vorzug des kombinierten Modells gegenüber allen anderen Erweiterungsmodellen ist die Vielfalt der gefundenen Erweiterungswörter. Sie stammen zum einen aus den betrachteten Dokumenten, aber auch aus einer lexikalischen Ressource. Dadurch werden die wenigen Synonyme aus GermaNet um weitere Erweiterungswörter aus den anderen Modellen erweitert. So können zu allen Suchwörtern zehn Erweiterungswörter gefunden werden.

Nachdem die Ergebnisse der Modelle betrachtet wurden, wird im Folgenden darauf eingegangen, welche Unterschiede es zwischen den Erweiterungswörtern gibt, welche die Testpersonen auswählten.

Um die ausgewählten Erweiterungswörter der Testpersonen vergleichen zu können, waren die ersten acht Suchwörter für alle Testpersonen gleich. Im Folgenden wird die Anzahl der ausgewählten Erweiterungswörter dieser acht Suchwörter in einem Box-Whisker-Diagramm dargestellt (siehe Abbildung 5.5). Zuerst sieht man, dass die Anzahl der ausgewählten Erweiterungswörter in fast allen Fällen zwischen mindestens drei Wörtern schwankt. Die einzige Ausnahme bildet *bremsen*.

Bei diesem Wort haben alle, bis auf eine Testperson, nur ein Erweiterungswort ausgewählt. Dies wird durch den Ausreißer dargestellt. Der Median der Boxen spiegelt den ermittelten Durchschnittswert von etwa zwei gut wider. In den meisten Fällen liegt zwischen Q1 und Q3 ein zusätzliches Erweiterungswort, was heißt, dass sich mindestens die Hälfte der Testpersonen in der Anzahl um nur ein Wort unterscheidet. Deutliche Abweichungen hiervon sind *StVO* und *verschmutzt*. Bei diesen Wörtern schwankt die Anzahl

---

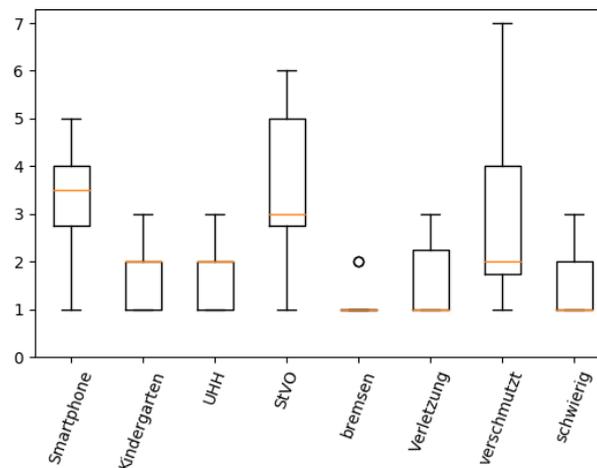


Abbildung 5.5: Verteilung der Anzahl der ausgewählten Erweiterungswörter für die acht konstanten Suchwörter.

der ausgewählten Erweiterungswörter von einem bis zu sechs, bzw. sieben. Die Abweichungen der Anzahl der Erweiterungswörter zeigt, dass sich die Testpersonen nicht immer gleich entscheiden.

Um genauer zeigen zu können, wie einig sich die Testpersonen beim Auswählen der Erweiterungswörter sind, wurde der Fleiss-Kappa-Wert berechnet. Dieser beträgt 54.43, womit die Testpersonen, nach der vorgeschlagenen Skala, eine angemessene Übereinstimmung für die ausgewählten Erweiterungswörter haben. Erst ab 75 spricht man von einer wirklich guten Übereinstimmung [13][7].

Wie schon beim Box-Plot zu sehen war, gibt es zwischen den Testpersonen Unterschiede bei der Auswahl der Erweiterungswörter. Aus diesem Grund ist es ratsam, ein Suchwort von mehr als einer Person testen zu lassen. Anschließend werden nur die Überschneidungen der ausgewählten Erweiterungswörter der lexikalischen Ressource hinzugefügt. So wird die realistischste Kombination von Erweiterungswörtern berücksichtigt.

### 5.2.2 Modellevaluation durch Doc2Vec

Im Folgenden wird ein explorativer Versuch zum Implementieren einer automatischen Evaluierung von Erweiterungswörtern vorgestellt. Dies ist interessant, da bei jedem Aktualisieren der Synonymdatei alle Suchwörter von Hand evaluiert werden müssten. Da dies zeitintensiv ist, ist dies nur bei kleinen Dokumentensammlungen möglich. Das Ziel ist zu prüfen, ob es möglich ist Erweiterungswörter automatisch zu evaluieren, indem neue Dokumente aus den erweiterten Ergebnisdokumenten, mit denen aus den unerweiterten Ergebnisdokumenten verglichen werden. Dies wurde durchgeführt, um einzuschätzen, wie wahrscheinlich es ist automatische Query Expansion mit menschlicher Genauigkeit zu erreichen und ob Forschung in dieser Richtung vielversprechend ist.

Die Implementierung stützt sich auf den Doc2Vec-Modellen von Gensim<sup>2</sup>. Für jedes Suchwort wird das Doc2Vec-Modell auf den unerweiterten Ergebnisdokumenten trainiert. Um ein Erweiterungswort zu evaluieren, wird bestimmt wie ähnlich die neu gefundenen Dokumente zu den unerweiterten Ergebnisdokumenten sind.

Sei  $K$  die Menge der unerweiterten Ergebnisdokumente und  $N$  die Menge der neuen Dokumente. So wird für jedes Dokument  $n \in N$  die Ähnlichkeit  $Sim(n, K)$  zu den unerweiterten Ergebnisdokumenten bestimmt. Diese Ähnlichkeit ergibt sich aus dem Durchschnitt der Distanzen zwischen dem neuen Dokument und den unerweiterten Ergebnisdokumenten.

$$Sim(n, K) = \frac{\sum_{i=1}^K d(n, k_i)}{|K|} \quad (5.1)$$

Die euklidische Distanz  $d$  zwischen den Dokumentvektoren  $p$  und  $q$  ist wie folgt definiert:

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (5.2)$$

Die Ähnlichkeit  $Sim$  zu den Suchdokumenten wird für jedes neue Dokument bestimmt und bildet die Menge  $S$ . Der Durchschnitt der Werte aus der Menge  $S$  ist der Ähnlichkeitswert, welcher dem betrachteten Erweiterungswort zugewiesen wird. Wie bei der Evaluierung durch die Testpersonen, wurden für jedes Suchwort zehn Erweiterungswörter evaluiert. Dieser werden in absteigender Reihenfolge nach Ähnlichkeit sortiert.

Um zu testen, wie verlässlich die automatische Evaluierung von Erweiterungswörtern ist, werden sie mit den ausgewählten Erweiterungswörtern der Testpersonen verglichen. Hierfür wird die Übereinstimmung der ausgewählten/akzeptierten Erweiterungswörter bewertet. D.h. wenn eine Testperson vier Erweiterungswörter ausgewählt hat, werden die vier ähnlichsten Erweiterungswörter der automatischen Evaluation mit diesen verglichen. Für den Vergleich entspricht die Anzahl der akzeptierten Erweiterungswörtern immer der Anzahl von Erweiterungswörtern, die von der Testperson ausgewählt wurden.

In der folgenden Tabelle ist die Übereinstimmung von fünf Ansätzen zum automatischen Auswählen von Erweiterungswörtern mit den Testpersonen beschrieben. Die ersten vier sind die Kombinationen aus Doc2Vec mit Distributed Memory (PV-DM), Distributed Bag-of-Words (PV-DBOW), sowie Negative Sampling (NS) und hierarchischen Softmax (HS). Der fünfte Ansatz kombiniert die ersten vier mit dem Rank-Level-Fusion-Verfahren (siehe Abschnitt 3.4.4).

Für den Vergleich mit den ausgewählten Erweiterungswörtern der Testpersonen wurden ersten acht Suchwörter der Testbögen verwendet. Dies hat den Vorteil, dass diese für alle Testpersonen gleich waren und sich so verschiedene Kombinationen von ausge-

<sup>2</sup><https://radimrehurek.com/gensim/models/doc2vec.html> zuletzt abgerufen 05.10.2019

wählten Erweiterungswörtern betrachten lassen. Für den Vergleich wurde die häufigste und die größte Kombination von Erweiterungswörtern betrachtet. Während die häufigste (siehe Tabelle 5.4) Kombination von den meisten Testpersonen ausgewählt wurde, sind bei der größten Kombination die meisten Erweiterungswörter von einer Testperson ausgewählt worden. Für *Smartphone* ist die häufigste Kombination *Endgerät, Handy, Mobiltelefon und SMS*, während die größte Kombination *Endgerät, Handy, Mobiltelefon, SMS und Tablet* ist.

Suchwort	EW 1	EW 2	EW 3	EW 4
Smartphone	Endgerät	Handy	Mobiltelefon	SMS
Kindergarten	Kita	Schulkindergarten	-	-
UHH	Universität	-	-	-
StVO	STVO	StVO.	Straßenverkehrsordnung	-
bremsen	abbremsen	-	-	-
Verletzung	Formvorschriften	-	-	-
verschmutzt	verdreht	-	-	-
schwierig	problematisch	-	-	-

Tabelle 5.4: Häufigste Kombination von Erweiterungswörtern für die acht Suchbegriffe.

Während die häufigste Kombination die realistische Auswahl von Erweiterungswörtern darstellt, birgt sie auch ein Problem. In fünf Fällen besteht die häufigste Kombination aus nur einem Erweiterungswort. Dies führt dazu, dass z.B. für das Suchwort *verschmutzt* alle Ansätze zu 0% mit den Testpersonen übereinstimmen, obwohl das häufigste Erweiterungswort *verdreht* nur um 0,87% schlechter als *zugewachsen* ist. Um die Anzahl der betrachteten Erweiterungswörter pro Suchwort zu erhöhen, wird deshalb neben der häufigsten Kombination auch die größte Kombination verwendet.

Modelle	häufigste Kombination von ausgewählten EW der Testpersonen	größte Kombination von ausgewählten EW der Testpersonen
PV-DBOW NS	38,54%	54,76%
PV-DBOW HS	12,50%	63,10%
PV-DM NS	30,21%	48,81%
PV-DM HS	36,46%	52,98%
Kombiniert	38,54%	58,93%

Tabelle 5.5: Vergleich von Erweiterungswörtern, die jeweils von den Testpersonen und den automatischen Ansätzen ausgewählt wurden.

Wie erwartet, ist die Übereinstimmung der Ansätze bei den häufigsten Kombinationen geringer als bei den größten Kombinationen. Wie bei den Erweiterungsmodellen führt auch hier das Kombinieren der Ansätze durch Rank-Level-Fusion zu keiner Verbesserung der Ergebnisse. Die Evaluation der Modelle zeigte, dass das beste Modell eine 63% Übereinstimmung mit den Testpersonen erreichte. Daraus lässt sich schließen, dass über die Hälfte aller ausgewählten Erweiterungswörter ebenfalls bei der automatischen Eva-

luierung akzeptiert wurde. Da die für den explorativen Versuch verwendeten Modelle einfach gehalten sind, gibt es noch viel Verbesserungspotential. Die Übereinstimmung zur händischen Evaluierung könnte durch ein verfeinertes Auswahlverfahren der Erweiterungswörter verbessert werden. Hierfür könnte festgelegte oder dynamische Ähnlichkeitsschwelle verwendet werden. Eine mögliche dynamische Ähnlichkeitsschwelle könnte sein, dass nur Erweiterungswörter akzeptiert werden, die nicht mehr als ein festgelegter Prozentsatz von dem besten Ergebnis abweichen.

Ein so verbessertes Verfahren könnte eingesetzt werden um die generierten Erweiterungswörter zu filtern, um dem Domainexperten nur die relevanten Erweiterungswörter anzuzeigen. Abhängig von der Erfolgsrate des automatischen Evaluierens könnte es den Domainexperten ersetzen und so das System zu einem voll-automatischen Erweiterungssystem machen. Weitere Versuche in dieser Richtung wirken vielversprechend.

Die Ergebnisse der Modelle bei der semi-automatischen Erweiterung sowie die der Oberflächenevaluation werden im letzten Kapitel zusammengefasst. Zudem wird die Leitfrage beantwortet und ein Ausblick auf weiterführenden Forschungsmöglichkeiten geboten.

---

## 6 Fazit und Ausblick

Ziel der Bachelorarbeit war es zu prüfen, ob sich eine lexikalische Ressource zur Query Expansion mit einem semi-automatischen Verfahren erweitern lässt. Getestet wurde das Verfahren an den Dokumenten des Hamburger Transparenzportals. Es lässt sich jedoch auch auf andere Dokumentensammlungen anwenden. Besonderen Fokus erhielt das Finden von guten Erweiterungswörtern und das Unterstützen von Domainexperten, welche diese mit Hilfe einer Benutzerschnittstelle testen. Als Testmodell wurden Word Embeddings, ein Wörterbuch mit Synonymen, sowie ein POS-Modell miteinander kombiniert, um ein kombiniertes Worterweiterungsmodell zu erhalten. Statt die verwendeten Modelle über die Relevanz der gefundenen Dokumente zu evaluieren, wurden die jeweils gefundenen Erweiterungswörter bewertet. Die Benutzerschnittstelle zum Testen der Erweiterungswörter wurde nach verschiedenen Design-Heuristiken entwickelt, um den Testprozess zu unterstützen. In Personentests konnte gezeigt werden, dass so eine Benutzeroberfläche mit hoher Usability entwickelt werden konnte, welche den Domainexperten die Evaluation der Erweiterungswörter ermöglicht. Zusammenfassend lässt sich urteilen, dass das hier vorgestellte Erweiterungssystem gut geeignet ist, um die lexikalische Ressource des Hamburger Transparenzportals zu erweitern.

Das kombinierte Modell hatte zwar ein schlechteres Verhältnis zwischen ausgewählten und generierten Erweiterungswörtern als die meisten der zum Kombinieren verwendeten Modelle. Jedoch sind die gefundenen Erweiterungswörter diverser. Dieser Vorteil beruht darauf, dass verschiedenen Modelle verwendet wurden, um sie zu finden. Die Anzahl der im Durchschnitt ausgewählten Erweiterungswörter, wenn mindestens eins ausgewählt wurde, war zwei. Somit werden beim Erweitern der Synonymdatei mindestens genau so viele Erweiterungswörter pro Suchwort hinzugefügt, wie durchschnittlich für die restlichen Synonyme in der Synonymdatei definiert sind. Außerdem zeigte sich, dass das Verwenden von Negative Sampling beim CBOW-Modell zu besseren Erweiterungswörtern als Hierarchischer SoftMax führt. Im Gegensatz dazu unterscheiden sich die Ergebnisse der Modelle von SkipGram beim Testen der Erweiterungswörter kaum.

Außerdem wurde mit quantitativen Tests gezeigt, dass die Benutzeroberfläche zum Auswählen von Erweiterungswörtern den Ansprüchen einer guten Bedienbarkeit (Usability) gerecht wird. Die wichtigen Verbesserungen für die Benutzeroberfläche sind das Hinzufügen von weiteren Shortcuts und das Verwenden von einem intuitiven Indikator für neue Datensätze. Abseits von dem Erweiterungssystem zeigten die Ergebnisse, dass die ausgewählten Erweiterungswörter zwischen den Testpersonen variieren. Deshalb wird bei der Anwendung empfohlen, mehr als eine Testperson pro Suchwort testen

---

zu lassen und deren Überschneidungen hinzuzufügen.

Zuletzt wurde mit einem explorativen Versuch getestet, ob automatische Query Expansion mit menschlicher Genauigkeit realistisch ist. Die Ergebnisse des Versuchs zeigen, dass Forschung in dieser Richtung sinnvoll wäre. Zudem wurde festgestellt, dass die Doc2Vec-Modelle nicht nur zum automatischen Evaluieren verwendet werden können. Ein weiteres Einsatzgebiet könnte das Filtern von Erweiterungswörtern sein, bevor die Verbleibenden von einem Domainexperten evaluiert werden.

Im Folgenden werden verschiedene Möglichkeiten zur Weiterentwicklung des kombinierten Modells und des Erweiterungssystems vorgeschlagen. Es gibt verschiedene Ansätze, welche verfolgt werden können, um das kombinierte Modell zu optimieren. Eine Möglichkeit ist das Optimieren der verwendeten Modelle. Hier könnte beispielsweise ein anderes Verfahren zum Ranking der Erweiterungswörter bei Wörterbuchansätzen getestet werden. Ein weiterer vielversprechender Ansatz ist das Gewichten der Modelle innerhalb des kombinierten Modells. Hierfür müsste zuerst ein Test ohne Gewichtung durchgeführt werden. Anschließend können die Modelle auf Basis der Ergebnisse unterschiedlich gewichtet werden.

Eine mögliche Verbesserung der Automatisierung könnte das Verwenden von Filtern sein. Es würden allgemein schlechte Erweiterungswörter auf Kosten von Sonderfällen entfernt werden. Dies ist sinnvoll, falls die Sonderfälle nicht von Doc2Vec erkannt werden. Eine weitere Möglichkeit zum Verbessern des automatischen Evaluierens von Erweiterungswörtern ist das Testen von verschiedenen Ähnlichkeitsschwellen. Zuletzt führt die Möglichkeit vom automatisierten Finden von Erweiterungswörtern dazu, dass verschiedene Konfigurationen von kombinierten Modellen getestet werden könnten. Diese können entweder manuell oder durch automatische Verfahren erstellt werden. Eine Konfiguration könnte die verwendeten Modelle und deren Gewichtung, aber auch die Schwellenwerte der Filter und Ähnlichkeitsschwellen für Doc2Vec enthalten.

---

---

## Literaturverzeichnis

- [1] ALMASRI, M., BERRUT, C., AND CHEVALLET, J.-P. A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information. In *European conference on information retrieval* (Berlin, Germany, 2016), pp. 709–715.
  - [2] BANGOR, A., KORTUM, P. T., AND MILLER, J. T. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.
  - [3] BIEMANN, C., COPPOLA, B., GLASS, M. R., GLIOZZO, A., HATEM, M., AND RIEDL, M. JoBimText visualizer: a graph-based approach to contextualizing distributional similarity. In *Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing* (Seattle, Washington, USA, 2013), pp. 6–10.
  - [4] BIEMANN, C., AND RIEDL, M. Text: Now in 2D! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling* 1, 1 (2013), 55–95.
  - [5] CARPINETO, C., AND ROMANO, G. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys* 44 (2012), 1–50.
  - [6] DIAZ, F., MITRA, B., AND CRASWELL, N. Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Berlin, Germany, 2016), pp. 367–377.
  - [7] DOWN, M., CZUBAK, F., GRUSKA, G., STAHLEY, S., AND BENHAM, D. Measurement system analysis—reference manual, 2010, [http://www.rubymetrology.com/add\\_help\\_doc/MSA\\_Reference\\_Manual\\_4th\\_Edition.pdf](http://www.rubymetrology.com/add_help_doc/MSA_Reference_Manual_4th_Edition.pdf).
  - [8] FANG, H. A re-examination of query expansion using lexical resources. In *Proceedings of Association for Computational Linguistics-08: Human Language Technology* (Columbus, Ohio, USA, 2008), pp. 139–147.
  - [9] FLEISS, J. L. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76, 5 (1971), 378–382.
  - [10] GWET, K. L. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC, 2014.
-

- [11] HAMP, B., AND FELDWEG, H. GermaNet - a lexical-semantic net for German. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications* (Madrid, Spain, 1997), pp. 9–15.
  - [12] KHAN, W., DAUD, A., KHAN, K., NASIR, J. A., BASHERI, M., ALJOHANI, N., AND ALOTAIBI, F. S. Part of speech tagging in Urdu: Comparison of machine and deep learning approaches. pp. 38918–38936.
  - [13] LANDIS, J. R., AND KOCH, G. G. The measurement of observer agreement for categorical data. *Biometrics* (1977), 159–174.
  - [14] LAVRENKO, V., AND CROFT, W. B. Relevance based language models. In *Proceedings of the 24th Annual International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval* (New Orleans, Louisiana, USA, 2001), pp. 120–127.
  - [15] LE, Q., AND MIKOLOV, T. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32* (Beijing, China, 2014), pp. II–1188–II–1196.
  - [16] LEVY, O., AND GOLDBERG, Y. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Baltimore, Maryland, USA, 2014), vol. 2, pp. 302–308.
  - [17] MACDONALD, C., AND OUNIS, I. Expertise drift and query expansion in expert search. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management* (New York, New York, USA, 2007), pp. 341–350.
  - [18] MANDALA, R., TOKUNAGA, T., AND TANAKA, H. Combining multiple evidence from different types of thesaurus for query expansion. In *Special Interest Group on Information Retrieval* (Berkeley, California, USA, 1999), vol. 99, pp. 15–19.
  - [19] MCGILL, R., TUKEY, J. W., AND LARSEN, W. A. Variations of box plots. *The American Statistician* 32, 1 (1978), 12–16.
  - [20] MIKOLOV, T., CORRADO, G., CHEN, K., AND DEAN, J. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations* (Scottsdale, Arizona, USA, 2013).
  - [21] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. S., AND DEAN, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (Nevada, California, USA, 2013), pp. 3111–3119.
  - [22] MITRA, M., SINGHAL, A., AND BUCKLEY, C. Improving automatic query expansion. In *ACM Special Interest Group on Information Retrieval* (Melbourne, Australia, 1998), pp. 206–214.
-

- 
- [23] MNIH, A., AND WHY TEH, Y. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning* (Edinburgh, Scotland, 2012), pp. 419–426.
- [24] MONWAR, M. M., AND GAVRILOVA, M. L. Multimodal biometric system using rank-level fusion approach. *Institute of Electrical and Electronics Engineers, Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 4 (2009), 867–878.
- [25] MORATO, J., MARZAL, M. A., LLORÉNS, J., AND MOREIRO, J. WordNet applications. In *Proceedings of 2nd Global WordNet Conference*. (Brno, Czech Republic, 2004), pp. 270–279.
- [26] MORIN, F., AND BENGIO, Y. Hierarchical probabilistic neural network language model. In *Aistats* (Barbados, 2005), vol. 5, pp. 246–252.
- [27] NIELSEN, J. Enhancing the explanatory power of usability heuristics. In *Proceedings of the Special Interest Group on Computer-Human Interaction Conference on Human Factors in Computing Systems* (Boston, Massachusetts, USA, 1994), pp. 152–158.
- [28] OUNIS, I., AMATI, G., PLACHOURAS, V., HE, B., MACDONALD, C., AND LIOMA, C. Terrier: A high performance and scalable information retrieval platform. In *Proceedings of the Open Source Information Retrieval Workshop* (Seattle, Washington, USA, 2006), pp. 18–25.
- [29] ŘEHŮŘEK, R., AND SOJKA, P. Software framework for topic modelling with large corpora. In *Proceedings of the Language Resources and Evaluation Conference 2010 Workshop on New Challenges for NLP Frameworks* (Valletta, Malta, 2010), pp. 45–50.
- [30] RONG, X. Word2vec parameter learning explained. *arXiv:1411.2738v4*, <https://arxiv.org/abs/1411.2738> (2014).
- [31] SHNEIDERMAN, B., AND PLAISANT, C. *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India, 2010.
- [32] SHTOK, A., KURLAND, O., AND CARMEL, D. Predicting query performance by query-drift estimation. In *Advances in Information Retrieval Theory* (Berlin, Heidelberg, 2009), pp. 305–312.
- [33] THIELSCH, M., AND HIRSCHFELD, G. *Expertise Website - Evaluation: Übersicht über bestehende Evaluationsmethoden und Entscheidungshilfe für die Evaluation bestehender sowie neu geschaffener Websites*. Technical Report, Bundeszentrale für gesundheitliche Aufklärung, 2019.
- [34] TULLIS, T. S., AND STETSON, J. N. A comparison of questionnaires for assessing website usability. In *Usability professional association conference* (Minneapolis, Minnesota, USA, 2004), vol. 1, pp. 1–12.
-

- [35] VAIDYANATHAN, R. A study on retrieval models and query expansion using PRF. *International Journal of Scientific and Engineering Research* 6 (2015), 13–18.
- [36] VOORHEES, E. M. Query expansion using lexical-semantic relations. In *Special Interest Group on Information Retrieval* (Dublin, Ireland, 1994), pp. 61–69.
-

# Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Ich bin mit einer Einstellung in den Bestand der Bibliothek des Fachbereiches einverstanden.

Hamburg, den \_\_\_\_\_ Unterschrift: \_\_\_\_\_