Institut für Visualisierung und Interaktive Systeme

Abteilung Grafisch-Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Bachelorarbeit Nr. 122

Multi-Dokumenten-Analyse mittels erweiterter Word-Cloud-Visualisierung

Cyrill Fabian Bopp

Studiengang:	Informatik B.Sc.
Prüfer:	Prof. Dr. Thomas Ertl
Betreuer:	Steffen Lohmann, Florian Heimerl
begonnen am:	14.04.2014
beendet am:	13.10.2014
CR-Klassifikation:	H.3.3, H.5.2, I.2.7

Kurzfassung

Word-Clouds werden schon länger in der Textanalyse eingesetzt. Für die Analyse und den Vergleich mehrerer Dokumente gibt es bisher nur wenige Entwicklungen. Mit dem hier vorgestellten Hilfsmittel sollen mehrere Dokumente miteinander verglichen werden und mit Hilfe der Word-Cloud-Visualisierung, WordPie, das Ergebnis dargestellt werden. Aus benachbarten Dokumenten können die gemeinsamen Wörter extrahiert werden und in eine passenden radiale Formation angeordnet werden. Mit Hilfe der Interaktionen durch Berühren und Klicken der Wörter mit der Maus lassen sich die vollen Ausmaße der einzelnen Word-Clouds analysieren. Der WordPie wird zum Schluss mit einem Vergleichslayout durch ein Expertengespräch verglichen.

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Einführung

Eine gewichtete Wortliste, besser bekannt als Word-Cloud oder Tag-Cloud, wird schon seit Jahren erfolgreich in der Textanalyse eingesetzt. Eine der ersten Word-Clouds als gewichtete Wortliste wurde in der Novelle *Microserf* von Douglas Coupland [CS95] verwendet. Im Deutschen gab es bereits 1992 mit dem Werk *Tausend Plateaus. Kapitalismus und Schizophrenie.* von Gilles Deleuze [DG92] einen visuellen Ansatz einer Word-Cloud auf dem Cover. Word-Clouds werden meist zweidimensional angeordnet, wobei die einzelnen Wörter unterschiedlich groß dargestellt werden, um ihre Gewichtung zu repräsentieren. Durch Wörter, die häufig auftauchen und somit hoch gewichtet werden, lässt sich ein erster Eindruck über den Inhalt des Textes verschaffen. Die meisten Entwicklungen zielen dabei meist nur auf ein einzelnes Dokument ab. Um die Analyse und den Vergleich zwischen mehreren Dokumenten zu ermöglichen werden neue Hilfsmittel benötigt. In dieser Arbeit wurde ein Konzept entwickelt, die mittels maschineller Sprachanalyse, die Gemeinsamkeiten mehrerer Dokumente analysiert. Das Ergebnis der Analyse wird in eine für den Nutzer übersichtliche Form dargestellt. Dabei werden Wörter automatisch aus den Texten extrahiert und mit einer passenden Gewichtung in einer Word-Cloud-Visualisierung dargestellt. Durch zusätzliche Interaktionen wird die anschließende Analyse unterstützt.

Im folgenden Kapitel wird der aktuelle Stand der Forschung zu diesem Thema erläutert. Anschließend wird in Kapitel 3 das Konzept vorgestellt, welches in Kapitel 4 realisiert wird. In Kapitel 5 werden einige Anwendungsfälle und Sonderfälle vorgestellt gefolgt von einer Diskussion in Kapitel 6. Zum Schluss wird, in Kapitel 7, die Arbeit durch ein Fazit und einem Ausblick auf weitere Arbeiten zu diesem Thema geschlossen.

2. Forschungsstand

Durch einen stetigen Zuwachs an neuen Arten von Word-Clouds, die je nach Anwendungsgebiet Vor- und Nachteile haben, lassen sich schon viele Fragestellungen bearbeiten. Für die Verbesserung von Word-Clouds wurden unterschiedliche Aspekte betrachtet. Zum einen wurde die visuelle Darstellung von Wörtern in einer Word-Cloud erforscht, um die Suche nach interessanten Schlüsselworten zu verbessern. Zum anderen wurde die Gesamtdarstellung der Word-Clouds bearbeitet, damit sich diese den speziellen Aufgaben anpassen.

2.1. Visuelle Darstellung der Wörter

Um die Darstellung der Wörter zu verbessern, muss zuerst die Effizienz der verschiedenen Parameter getestet werden. In Benutzerstudien hatten Bateman et al. [BGN08] und Rivadeneira et al. [RGMM07] herausgefunden, dass unterschiedliche Schriftgrößen den Nutzern am meisten auffallen. Dabei wurden höher gewichtete Wörter größer dargestellt als geringer gewichtete. Dadurch wurden die wichtigen Wörter schneller erkannt, vor allem wenn die Sortierung alle wichtigen Wörter in der gesamten Visualisierung verstreut. Da oft neben den einzelnen Wörtern auch der Zusammenhang der Wörter untereinander von Interesse ist, sind diese durch interaktive Elemente auf unterschiedliche Arten verknüpft. Dörk et al. [DCCW08] arbeiteten mit farblichen Worthervorhebungen. Dabei wurden die ausgewählten Wörter mit den zugehörigen Wörtern durch andere Hintergrund- und Font-Farben verbunden. Damit können zusammengehörige Wörter, die beispielsweise häufig im gleichen Satz auftauchen, leichter erkannt und der Zusammenhang des vorkommenden Wortes besser analysiert werden.
Allerdings müssen, um die gesamte Word-Cloud einer Suchaufgabe anzupassen, grundlegende Elemente geändert werden, auf welche im nächsten Abschnitt eingegangen wird.

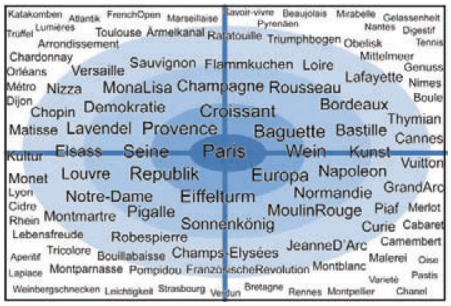
2.2. Visuelle Darstellung der Word-Cloud

Unter den klassischen Word-Clouds gibt es zwei typische Formen. Die eine Form ist rechteckig angeordnet und listet die Worte alphabetisch oder in absteigender Gewichtung auf. Dabei werden die Wörter immer auf die gleiche Grundlinie geschrieben, nur die Größe der Wörter unterscheidet sich. Die andere Form ist radial. In dieser werden meist die wichtigeren Wörter in das Zentrum geschrieben und die Wichtigkeit nimmt mit wachsendem Radius ab. Des weiteren gibt es noch geclusterte Word-Clouds, bei denen die Wörter nach speziellen Kategorien angeordnet werden. Diese können entweder durch zusammengehörige Wörter gegeben sein oder, wie in der hier vorgestellten Variante, durch verschiedene Quellen und deren Kombinationen. Ein paar dieser Visualisierungen wurden von Lohmann et al. [LZT09] in einer Benutzerstudie miteinander verglichen, wie in Abbildung 2.1 schematisch dargestellt.

Hearst und Rosner [HR08] beobachteten Schwächen der alphabetischen Anordnung bei rechteckigen Word-Clouds. Durch diese Anordnung kann es zur „falschen Gruppierung“ von Wörtern kommen. Das bedeutet, dass Wörter mit ähnlichen Bedeutungen weit auseinander liegen können und dadurch die Interpretation der Word-Cloud verfälscht wird. Außerdem muss der Leser einer solchen Word-Cloud diese erst im Gesamten erkunden, bevor er ein klares Bild über die wichtigsten Wörter bekommen kann. Allerdings lässt es sich einfacher nach bestimmten Wörtern suchen, die in dieser Word-Cloud erwartet werden, als in anderen Anordnungen. Ebenfalls kommen nach Gewichtung sortierte Word-Clouds nicht an eine ideale Sortierung heran, weil zusammengehörige Wörter nahe beieinander liegen sollten. Auch die unterschiedlichen Wortgrößen wären dadurch unnötig und damit wird die Word-Cloud als solche nicht mehr benötigt. Jedoch werden so die wichtigsten Wörter zu Beginn aufgelistet und somit kann schnell der wichtigste Zusammenhang erkannt werden. Einen Ansatz hatte Stefaner [Ste07] mit *Elastic Tag Maps* realisiert. Dabei wurden die Wörter radial mit abnehmender Wichtigkeit angeordnet. Mittels Markieren eines Wortes, durch Berühren mit der Maus, wurden die Wörter, die mit diesem Wort in Verbindung gebracht wurden, farblich hervorgehoben und mit Hilfe einer Linie verbunden.

Neben diesen allgemeinen Word-Clouds gibt es auch noch welche, die auf bestimmte Aufgaben angepasst wurden. Die von Lee et al. [LRKC10] entwickelten *SparkClouds*, siehe Abbildung 2.2, repräsentieren neben dem Wort auch noch die Verteilung des Vorkommens des Wortes im gesamten Text oder über die Zeit verteilt. Eine Studie von Lee et al. [LRKC10] unterstützte deren Hypothese, dass sich mit Hilfe einer SparkCloud der Trend eines Wortes über die Zeit effektiv vermitteln lässt. Für eine übersichtliche Darstellung von großen Word-Clouds hatten Fujimura et al. [FFM+08] *Topigraphy*, siehe Abbildung 2.3, entwickelt, die einer

(a) Eine rechteckig angeordnete Word-Cloud mit alphabetischer Reihenfolge der Wörter, die auf der gleichen Basis-Linie aufgelistet wurden.



(b) Eine zirkuläre Anordnung eine Word-Cloud mit abnehmender Gewichtung der Wörter von Innen nach Außen.

(c) Eine Word-Cloud mit geclusterten Ansammlungen von Wörtern. Die Cluster werden in diesem Bild durch die blauen Kreise angedeutet.

Abbildung 2.1.: Drei unterschiedliche Anordnungsschemata nach Lohmann et al., wobei die blauen Kreise, Pfeile und Striche in der Word-Cloud nicht zu sehen sind. [LZT09, Figure 2]

Abbildung 2.2.: Die SparkClouds stellen neben den Wörtern auch deren Vorkommen im gesamten Text dar. Durch die Graphen unter den Wörtern lässt sich ablesen, wann das Wort im Text vorkam. [LRKC10, Figure 1]

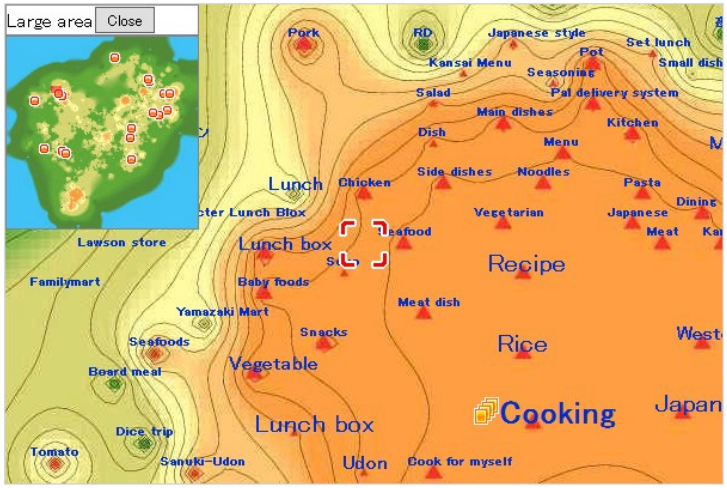


Abbildung 2.3.: Bei der Topigraphy werden die Wörter auf eine Höhenkarte gesetzt. Dabei werden wichtigere Terme höher gesetzt als unwichtigere. [FFM+08, Figure 1]

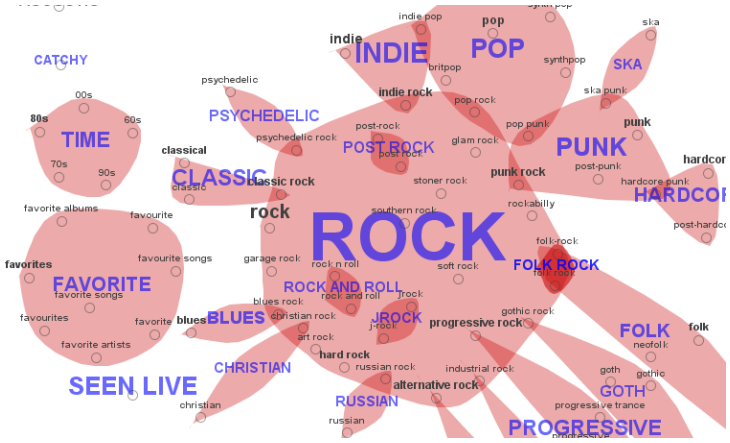


Abbildung 2.4.: Tagclusters kategorisiert vorkommende Tags und zeigt diese überlappend als Graph an. [CSBT09, Figure 2]

Höhenlandkarte ähnelt. Durch Farbverläufe können die wichtigsten Wörter leicht gefunden werden, da diese farblich hervorstechen. Ebenso entwickelten Chen et al. [CSBT09] eine Visualisierung um Word-Clouds übersichtlicher zu gestalten, indem sie die vorkommenden Begriffe in Kategorien einordnet und diese überlappend als Graph, wie in Abbildung 2.4, darstellt.

Gambette und Véronis [GV10] ordneten die Wörter in einer Baumstruktur an, bei der ebenfalls die Wortgröße die Häufigkeit widerspiegelt. Diese Visualisierung hilft die Hauptthemen eines Textes hervor zu heben. Um die Vielfalt eines vorkommenden Wortes mit dem selben Stamm zu visualisieren, haben Burch et al. [BLPW13] *Prefix Tag Clouds* entwickelt. In dieser Art von Word-Cloud werden alle vorkommenden Formen eines Wortes in einen Präfix Baum zusammengefasst und in der Word-Cloud dargestellt. In Abbildung 2.5 ist ersichtlich, dass unter anderem die Wörter „visualisation“ und „visualizing“ mit dem selben Wortstamm „visual“ in dem analysierten Text auftauchen.

Abbildung 2.5.: Eine zirkulär angeordnete Prefix Tag Cloud nach Burch et al. [BLPW13, Figure 4b]

Um eine weitere Dimension erweitert ist die *WP Cumulus* von Wordpress [Wor12]. Sie ist eine der wenigen 3D-Word-Clouds, die momentan zum Einsatz kommen. Diese haben den Nachteil, dass sie ohne interaktive Bewegung sehr unübersichtlich sind, wie in Abbildung 2.6 zu sehen ist.

Eine Erweiterung von Word-Clouds auf mehreren Dokumenten realisierten Collins et al. [CVW09] mit *Parallel Tag Clouds*. Dabei wurde für jedes Dokument eine Word-Cloud in Form einer Spalte gebildet und mittels Linien die Zusammenhänge zwischen den Wörtern und das Auftauchen derselben Wörter in unterschiedlichen Dokumenten visualisiert, wie im Beispiel in Abbildung 2.7 sichtbar. Im Gegensatz dazu wurde im hier vorgestellten Ansatz versucht Redundanzen weitestgehend, durch geschickte Anordnung der Dokumenten und Kombinieren von Wörtern, die

Abbildung 2.6.: Beispiel einer WP Cumulus von Wordpress [Wor12].

in mehreren Dokumenten vorkommen, zu vermeiden. Dadurch sollen die Gemeinsamkeiten der Dokumente hervorgehoben werden, ohne die Bedeutung dieser Wörter im einzelnen Dokument zu analysieren.

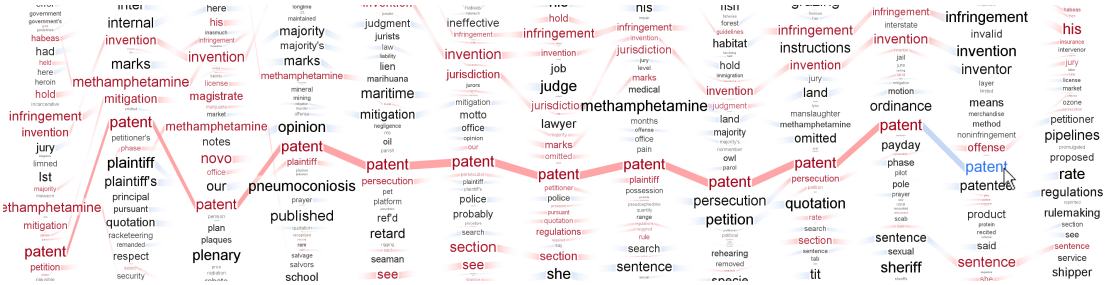


Abbildung 2.7.: Beispiel einer Parallel Tag Cloud mit Hervorhebung eines Wortes und seinen Vorkommen in den anderen Quellen. [CVW09, Figure 7]

Ein weiterer Ansatz, *RadCloud*, wurde von Burch et al. [BLB+] entwickelt. Dabei wurden ebenfalls mehrere Dokumente miteinander verglichen. Die Dokumente wurden auf einem Ring angeordnet und die wichtigsten Wörter in den Ring hineingeschrieben, wie es in Abbildung 2.8 sichtbar ist. Aus der Wichtigkeit der Wörter für die jeweiligen Dokumente wurde ein Vektor gebildet, der die Wörter in die Nähe der Dokumente verschiebt, in denen das Wort am häufigsten verwendet wird. Im Gegensatz dazu soll in dieser Arbeit der freie Raum minimiert werden. Auch soll strikt getrennt werden aus welchen und aus wie vielen Dokumenten ein Wort stammt.

Collins et al. [CCP09] hatten mit *Docuburst* eine Visualisierung entwickelt, die der hier vorgestellten sehr ähnelt, wie in Abbildung 2.9 sichtbar. Allerdings nicht

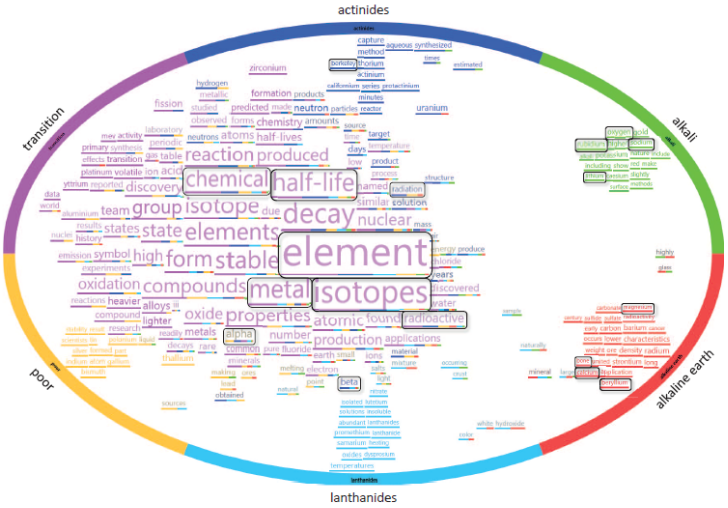


Abbildung 2.8.: Beispiel einer RadCloud, bei der 100 Wörter aus je sechs verschiedenen Arten von Metallen aus Wikipedia analysiert wurden. [BLB+, Figure 2]

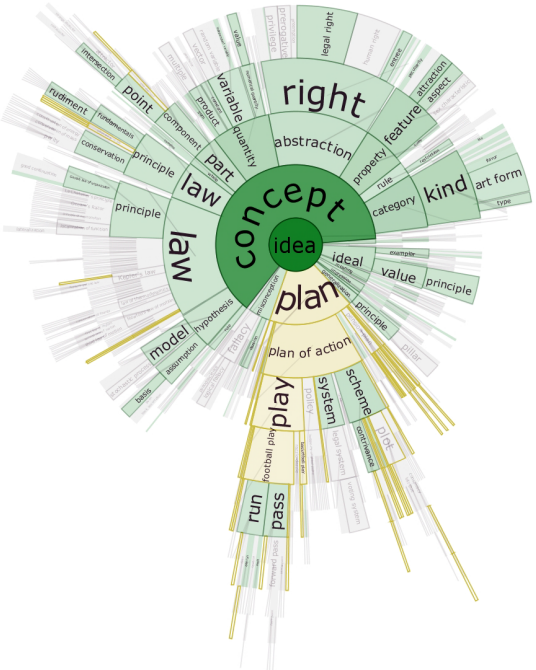


Abbildung 2.9.: Docuburst von einen naturwissenschaftlichen Buch basierend auf dem Wort „idea“. [CCP09, Figure 1]

als Word-Cloud, sondern es werden einzelne Wörter in ihren Familien hierarchisch angeordnet, wie beispielsweise die Kuh ein Rind ist, das wiederum ein Säugetier ist. Doch die Visualisierung ist recht ähnlich mit den Ringstücken, die zu einem Kreis angeordnet werden und diese mit Wörtern befüllt werden. Ebenso ist das mittlere Element ein Kreis, dass die wichtigste Rolle mit den Gemeinsamkeiten, die alle anliegenden Wörter verbindet.

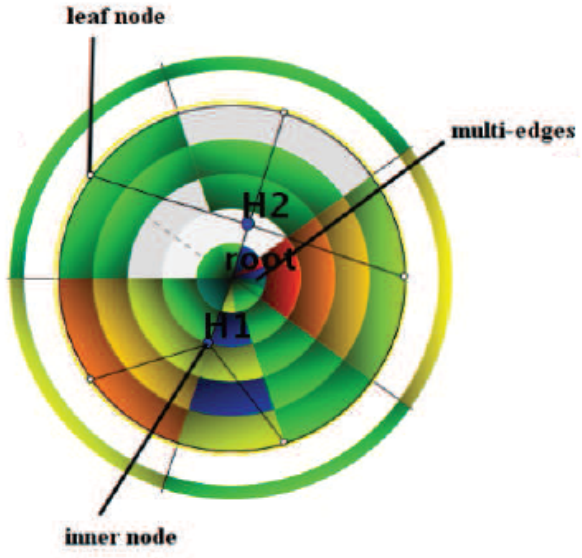


Abbildung 2.10.: Schema eines TimeRadarTrees. Dabei bedeuten unterschiedliche Einfärbungen unterschiedliche Vorgänge mit den entsprechenden Daten. [BRW13, Figure 1 abgewandelt]

Burch et. al [BRW13] verwendeten mit *TimeRadarTrees* ebenfalls eine ähnliche Visualisierung, allerdings zur Datenmodellierung und nicht als Word-Cloud. Dabei werden, wie in Abbildung 2.10 sichtbar, verschiedene Farben zur Darstellung von verschiedenen Datenvorgängen verwendet. Ebenfalls liegen die Kreisstücke parallel aufeinander und nicht versetzt, wie in diesem Ansatz.

2.3. Ästhetischer Nutzen

Neben den wissenschaftlichen Vorteilen, die sich aus den Word-Clouds ergeben, haben die Grafiker den ästhetischen Nutzen der Word-Clouds entdeckt. Eine Java-Applet mit Namen *Wordle* [Fei09] erzeugt zu jedem gegebenen Text eine Word-Cloud, die vom Nutzer visuell verschönert werden kann, indem unter anderem die

Schriftart und Farbe festgelegt werden kann. Diese Word-Clouds lassen sich als Bild exportieren und für weitere Zwecke nutzen.

2.4. Word-Cloud-Kritik

Allerdings gibt es auch Kritiker von Word-Clouds. Zeldman [Zel05] nannte Word-Clouds „the mullets of the internet“ in seinem Blog und bezeichnet sie damit als nervige Entdeckung aus einer vergangenen Epoche. Er kritisierte die zunehmende Nutzung von Word-Clouds auf diversen Webseiten. Dabei ging es vor allem um die unangemessene Nutzung von Word-Clouds, wie zum Beispiel als Browser für Kategorien auf Webseiten. Auf der Webseite Wohnungsboerse.net [woh] werden in der rechten Spalte die beliebtesten Stadtteile in einer spärlichen Word-Cloud angezeigt.

Harris [Har11] griff die Aussage von Zeldman auf und kritisierte Word-Clouds anhand eines Beispiels aus dem Irakkrieg, sichtbar in Abbildung 2.11. Er kritisierte vor allem die Nutzung von Word-Clouds außerhalb der Textanalyse. Anhand des Beispiels lasse sich gut zeigen, dass die Word-Clouds in einem solchen Zusammenhang keine nützlichen Informationen bereitstellen. Das genannte Beispiel mit den Wörtern „car“ und „blast“, die in etwa gleich groß sind, zeigt deutlich, dass sich daraus nicht erschließen lässt, ob diese in denselben Titeln vorkamen oder nicht. Ebenfalls sind viele Abkürzungen ohne Kontext nicht erschließbar. Er kritisierte, dass es Gebiete gibt, in denen eine Word-Cloud keinen informellen Gehalt hat. Denn dazu müssten noch mehr Informationen zur Word-Cloud-Generierung bekannt und angewandt worden sein, wie zum Beispiel eine nahe Anordnung von Wörtern, die oft zusammen vorkamen, was in seinem genannten Beispiel vermutlich nicht der Fall war.

Abbildung 2.11.: Word-Cloud aus allen Titeln der Irak-Kriegs-Aufzeichnungen von Fast Company

2.5. Term-Gewichtung

Die Gewichtung der Wörter, auch Term-Gewichtung genannt, sollte mit Bedacht gewählt werden. Es gibt viele Argumente, die gegen das alleinige Vorkommen der Wörter, die sogenannte Termfrequenz, als Gewichtung sprechen. Zum Beispiel tauchen Wörter in langen Texten öfter auf als in kurzen, dadurch würden manche Wörter in Texten höher gewichtet werden, nur weil der Text besonders lang ist.

Dies lässt sich verhindern, indem die Termfrequenz mit dem relativen Auftauchen des Wortes in allen Dokumenten gewichtet wird. Dadurch erhält man *tf-idf* (termfrequency × inverse documentfrequency) von Salton und McGill [SM83]. Dabei werden die Terme mit der inversen Dokumentenfrequenz gewichtet, sodass Wörter, die in vielen Dokumenten vorkommen, eine geringere Gewichtung bekommen als solche, die nur in wenigen Dokumenten vorkommen. Diese Gewichtung wird mit

$$\text{Gewicht}_w = tf_w \cdot \log \frac{N}{n_w} \tag{2.1}$$

realisiert, wobei tf_w für die Anzahl der Vorkommen des Wortes w in dem zu untersuchenden Dokument, N für die Gesamtzahl der Dokumente und n_w für die Anzahl der Dokumente, in denen das Wort w vorkommt, steht. Dieser Ansatz ist in der Art und Weise hier nicht möglich, da Wörter, die in allen Dokumenten auftauchen, mit null gewichtet werden würden.

Cressie und Read [CR84] zeigten, dass Pearson's χ^2-Test und Dunning's G^2-Test [Dun93], auch bekannt als log-likelihood, zwei gute statistische Methoden für die Textanalyse sind. Diese Methode beachtet zusätzlich zur Termfrequenz die Länge der Dokumente und berechnet aus den Erwartungswerten der einzelnen Wörter eine passende Gewichtung. G^2 wurde ebenfalls in Parallel Tag Clouds [CVW09] verwendet. Allerdings benötigt diese Methode immer die Aufteilung in einen Korpus und einen Sub-Korpus, was bei dem Vergleich aller Dokumente miteinander nicht mehr möglich ist. Auch entsteht durch das Kombinieren von G^2-Werten ein schwer interpretierbares Ergebnis, welches ebenfalls nicht erwartet wurde.

Somit bleibt zur Gewichtung bisher nur die reine Termfrequenz. Um zu verhindern, dass lange Dokumente einen Wert dominieren, wird zum Kombinieren der Werte der Mittelwert genommen. Mehr dazu in Kapitel 4.3. Aus den Erkenntnissen des bisherigen Forschungsstands hat sich ein Konzept entwickelt, dass im folgenden Kapitel vorgestellt wird.

3. Konzept

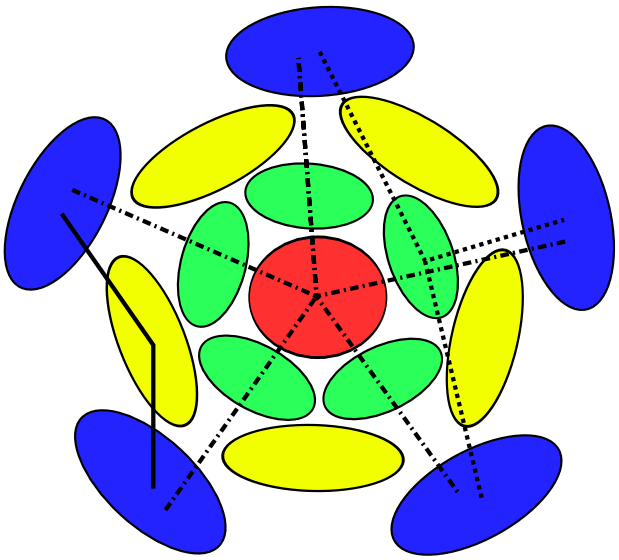


Abbildung 3.1.: Schema des Konzepts der unterschiedlichen Ebenen mit Strichen zur Visualisierung der dazugehörigen Dokumente.

Um einen Überblick über die Gemeinsamkeiten von mehreren Dokumenten zu erhalten, wird für jedes Dokument eine separate Word-Cloud erstellt. Diese werden in Kreisform verteilt angeordnet, sodass zum Zentrum hin die Kombinationen aus mehreren Dokumenten platziert werden können. Gemeinsame Wörter zwischen zwei oder mehr Dokumenten werden, für jede Kombination aus benachbarten Dokumenten, in eine zusätzliche Word-Cloud gelegt. Die im Zentrum entstehende Word-Cloud beinhaltet die Wörter, die in allen Dokumenten enthalten sind. Dadurch entsteht ein ähnliches Schema wie in Abbildung 3.1 zu sehen ist. Dabei sind im Schema die einzelnen Dokumente blau gefärbt, die Kombination aus zwei gelb, aus drei grün und aus allen Dokumenten rot. Die jeweils zugehörigen Dokumente sind schematisch durch die Striche gekennzeichnet. Die Word-Clouds, die nur Teilmengen der Dokumente zusammenfassen, werden entsprechend weiter vom Zentrum entfernt in Richtung der Dokumente platziert. Um Redundanzen zu vermeiden, werden die kombinierten Wörter aus den ursprünglichen Ebenen

gelöscht, sodass nur die Wörter in den tiefsten Ebenen bestehen bleiben. Durch diese Struktur sollen einzelne Word-Clouds mit den entsprechenden Wörtern jeder Dokumentenkombination aus benachbarten Dokumenten gefüllt werden und diese in einer passenden Visualisierung analysierbar gemacht werden. Um die abnehmende Anzahl an zusammengefassten Dokumenten zu repräsentieren, wurden die einzelnen Ebenen von Innen nach Außen heller werdend eingefärbt. An der Darstellung soll die Bedingung geknüpft sein, dass der Platz möglichst effizient ausgenutzt wird, sodass möglichst wenig freie Fläche übrig bleibt.

Um eine hohe Ähnlichkeit zwischen den benachbarten Dokumenten zu erhalten, wurden zwei Möglichkeiten verwendet. Dies kann einerseits eine natürliche Reihenfolge sein, wie die chronologische bei Romanen. Andererseits ließe sich eine gute Reihenfolge, mittels der vorkommenden Wörter, berechnen. Aus diesem Ansatz haben sich zwei Konzepte entwickelt, die im Folgenden vorgestellt werden.

3.1. WordFlower

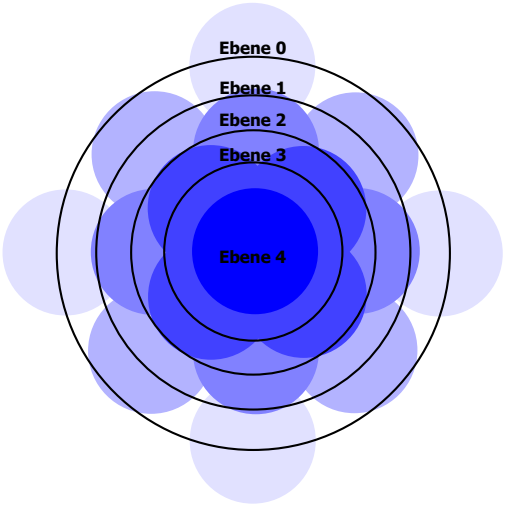
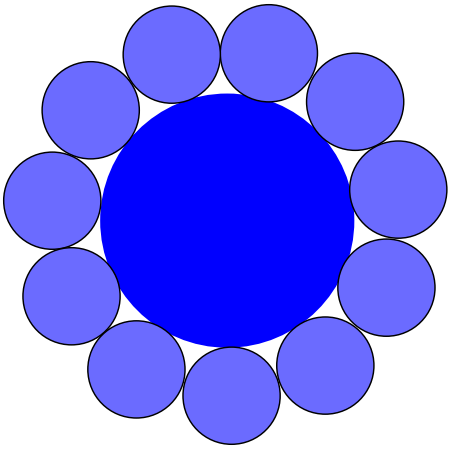


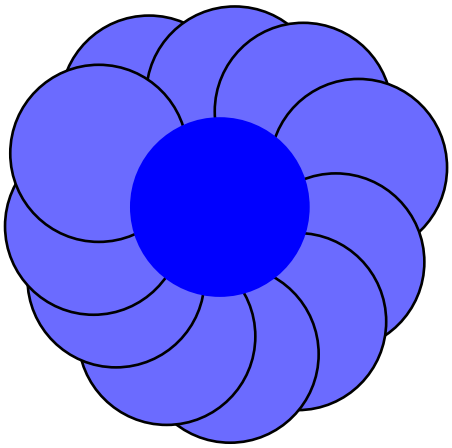
Abbildung 3.2.: Schematische Darstellung des WordFlower-Ansatzes, der Aufgrund der schlechten Platzausnutzung nicht weiter verfolgt wurde. Radien, auf denen die Word-Clouds angeordnet werden, sind mit eingezeichnet.

Das erste Konzept, WordFlower, ordnet radiale Word-Clouds in überlappenden Kreisen an, wobei sich jede Ebene versetzt zur vorherigen Ebene anordnet, sodass die Darstellung einer Blume ähnelt, sichtbar in Abbildung 3.2. Dabei werden die Word-Clouds auf einem Kreis gleichmäßig verteilt. Dies ergab sich aus der visuell

ansprechenden Form der runden Word-Cloud. Die Word-Clouds aus den kombinierten Dokumenten sollten als eigenständige Word-Cloud visualisiert werden, die allerdings durch passende Anordnung den Dokumenten zugeordnet werden können. In der Mitte befindet sich nur noch eine Word-Cloud, welche die Gemeinsamkeiten aller Dokumente beinhaltet. In der äußersten Ebene befinden sich Wörter, die nur in dem jeweiligen Dokument vorkommen und nicht in den benachbarten Dokumenten enthalten sind. In den weiter innen gelegenen Ebenen befinden sich die Wörter, die in allen zugehörigen Dokumenten auftauchen.



(a) Schema der innersten Ebenen, wenn die Überlappung durch angepasste Radien möglichst gering gehalten werden sollte. Durch die Hohe Anzahl von Dokumenten haben Word-Clouds, die nahe im Zentrum liegen wenig Platz und müssen sehr klein sein, um sich kaum zu überlagern.



(b) Schema der innersten Ebenen, wenn der Radius gleich groß gehalten werden sollte. Durch den engen Kreisumfang auf dem die Word-Clouds verteilt werden sollen, kommt es zu starker Überlappung.

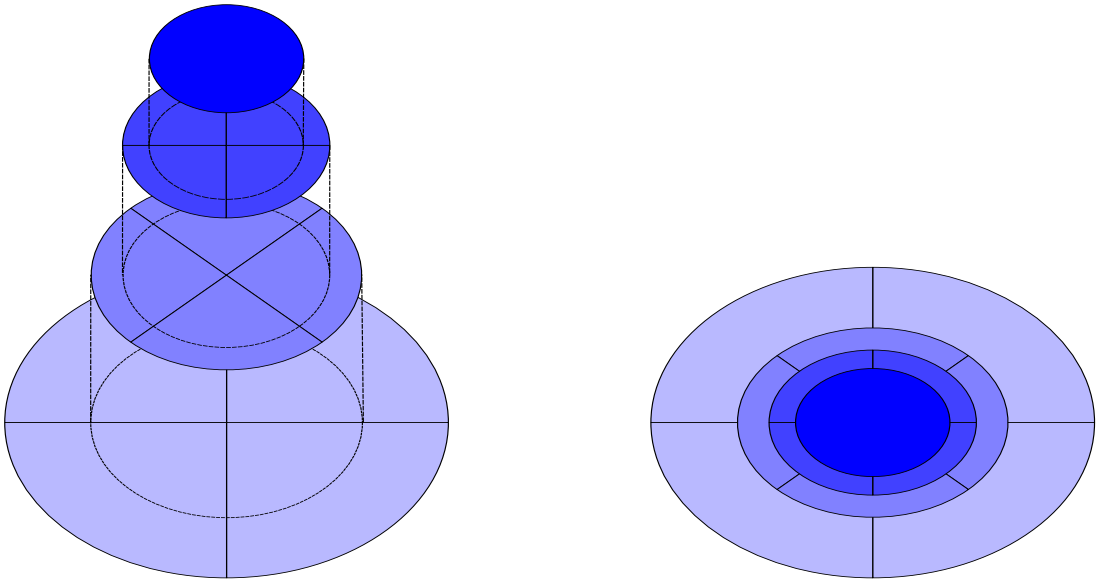
Abbildung 3.3.: Schemata der innersten beiden Ebenen bei 11 Dokumenten und die Probleme die dabei entstehen.

Diese Darstellung hatte allerdings den Nachteil, dass aufgrund der Überlappungen kaum Platz für die Wörter vorhanden war oder es zu viel freien Raum aufgrund des zunehmenden Radius kam. Der zunehmende freie Raum ist gut in der schematischen Abbildung 3.2 sichtbar, da sich mit jeder weiteren Ebene der Radius und somit der Kreisumfang, auf denen die einzelnen Word-Clouds einer Ebene angeordnet werden, vergrößert und entweder größere Word-Clouds benötigt werden oder zunehmender leerer Raum entsteht. Auch durch das Anpassen der Word-Clouds zu deren Inhalt, sodass leere Word-Clouds gar nicht gezeichnet wurden, führte zu leerem Raum oder einer unübersichtlichen Struktur.

Vor allem bei vielen Dokumenten kam es schnell zum Platzmangel, da entweder die inneren Ebenen, außer die innerste, extrem klein werden müssen, wie in Abbildung 3.3(a) sichtbar, oder sich zu über 50 % überlagerten, in Abbildung 3.3(b) gezeigt, und somit nur wenig Platz für die Wörter übrig blieb. Da das Kriterium der effizienten Platzausnutzung nicht gewährleistet werden konnte, wurde dieses Konzept nicht weiter verfolgt.

3.2. WordPie

Beim zweiten Konzept, WordPie, werden die Word-Clouds in Form von Kuchenstücken übereinander gelegt und somit entsteht für jede Ebene ein Ring, schematisch dargestellt in Abbildung 3.4. Um die Maße eines eher breiten Fensters auszunutzen, sind die Ringe nicht kreisrund, sondern in passender Form einer Ellipse angeordnet. Dies bietet zusätzlichen Platz für Wörter, da diese oft breiter als höher sind. Die mittlere Ebene besteht nur aus einer einzigen Ellipse, die im Zentrum liegt.



(a) Ein in z-Richtung verschobenes Schema des WordPies.

(b) Finale Form des WordPies.

Abbildung 3.4.: Schematische Darstellung des WordPies von vier Dokumenten. Dabei werden die einzelnen Wolken in Kuchenstücken dargestellt, sodass ein runde Form entsteht.

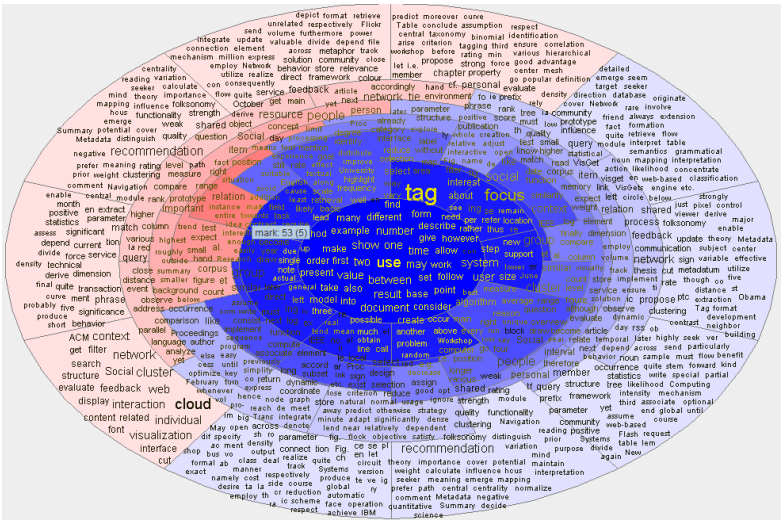


Abbildung 3.5.: WordPie mit gelöschten leeren Wolken. Dadurch entsteht ein starkes Durcheinander durch die überlappenden Wörter und den stark variierenden Radien, wodurch der WordPie unübersichtlich wird.

Um nun einen zu hohen Anteil an freien Raum zu vermeiden, werden die leeren Word-Clouds entfernt, sodass der Ebene darunter der freien Platz zur Verfügung gestellt wird. Dadurch kam es zu starken Verformungen, wie in Abbildung 3.5 sichtbar, die zu einem größeren Durcheinander geführt haben. Durch dieses Gewirr wurde von den vorkommenden Wörtern zu stark abgelenkt und dadurch die Benutzung schwerer gemacht.

Darum wurden alle bis auf die innerste und äußerste Ebene, zu einer Ebene zusammengefügt. Die Grundstruktur blieb erhalten nur wurde der Hintergrund der mittleren Ebenen einheitlich eingefärbt, sodass die Unterscheidung der einzelnen Ebenen nicht mehr möglich ist, dafür jedoch der Hintergrund ruhiger wurde und der Text in den Vordergrund rückte. So konnte der Platz effizient ausgenutzt werden und gleichzeitig die Anordnung der Wörter im Verhältnis zu ihre zugehörigen Dokumenten erhalten bleiben. Die strikte Trennung der einzelnen Word-Clouds wird dadurch nicht mehr benötigt. Durch einen radialen Farbverlauf soll der Eindruck der abnehmenden Anzahl an zusammengesetzten Dokumenten erhalten bleiben. Die innerste und die äußerste Ebene bleiben einfarbig, da diese besonders wichtig sind und nicht mit den anderen Ebenen vermischt werden sollen, wie in Abbildung 3.6 erkennbar. Dadurch entsteht auch die Bedingung, dass nur Wörter, die zur innersten Word-Cloud gehören, in dieser auch angezeigt werden dürfen. Ebenso sollen nur Wörter in den äußersten Word-Clouds angezeigt werden, die nicht auch in benachbarten Dokumenten vorkommen.

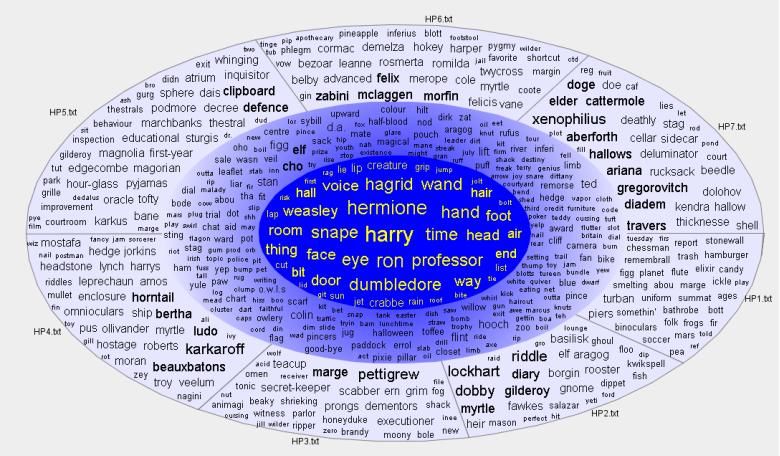


Abbildung 3.6.: WordPie bestehend aus den drei Ebenen und dem Farbverlauf von Innen nach Außen.

Da die äußerste und die innerste Ebene am interessantesten sind, bekommen diese mehr Platz als die restlichen Ebenen. Ein gutes Maß ergab die Aufteilung, dass die gesamte radiale Höhe in drei Teile aufgeteilt wird. Die innerste und die äußerste Word-Cloud bekommen jeweils ein Drittel und die restlichen mittleren Ebenen das restliche Drittel. Zusätzlich werden die Wörter in der innersten Word-Cloud gelb geschrieben, damit diese besonders hervorgehoben und durch den Kontrast mit Blau leichter zu lesen sind.

3.3. Vergleichslayout

Neben den oben vorgestellten Darstellungen wurde noch ein Vergleichslayout in Annäherung eines icicle plots entwickelt. Dabei werden die Word-Clouds nicht radial in Form von Tortenstücken, sondern senkrecht in Form von Rechtecken angeordnet. Dadurch entsteht eine rechteckige Form des WordPies wie in Abbildung 3.7 zu sehen.

Genau wie beim WordPie werden auch hier alle bis auf die innerste und äußerste Ebene zu einer gemeinsamen Ebene verschmolzen, sodass die Wörter den Platz besser ausnutzen können. Dadurch entsteht ein mittleres Band mit graduellem Farbverlauf, dass die Abnahme der zusammengefügten Dokumente visualisiert. Mit dem Anwendungsfall „Harry Potter“ sieht das Vergleichslayout wie in Abbildung 3.8 aus. Ebenso wurde die Bedingung an das Vergleichslayout gestellt, dass in der innersten Word-Cloud nur die Wörter vorkommen dürfen, die in allen Dokumenten vertreten sind. Ebenfalls sollen in den äußersten Word-Clouds keine

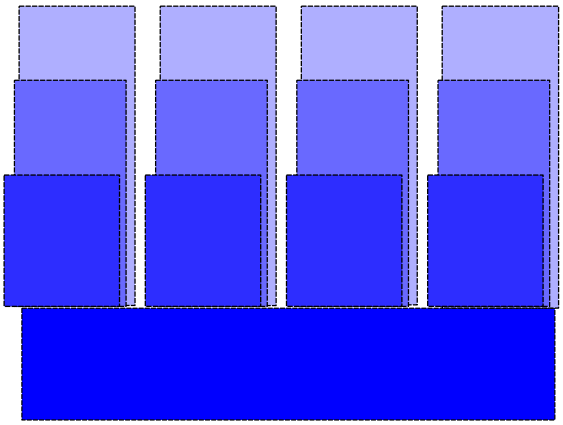


Abbildung 3.7.: Schematischer Aufbau des Vergleichslayouts.

Abbildung 3.8.: Vergleichslayouts mit dem Anwendungsfall „Harry Potter“.

Wörter gezeichnet werden, die in mehreren benachbarten Dokumenten auftauchen. Dadurch lassen sich die gemeinsamen Wörter aller Dokumente und die, die nur in den einzelnen Dokumenten auftauchen, leicht analysieren.

4. Implementierung

Um die vorgestellte Visualisierung zu realisieren, werden einige Algorithmen und externe Implementierungen benötigt, die im folgenden Abschnitt vorgestellt werden.

4.1. Textverarbeitung

Um Texte zu analysieren müssen diese erst verarbeitet werden. Dazu wird hier das Stanford CoreNLP der Stanford Natural Language Processing Group verwendet. Trotz hoher Auslastung des Arbeitsspeichers bietet es alle Funktionen, die in diesem Kontext benötigt werden. Zum einen wird die Lemmatisierung genutzt, um die vorkommenden Wörter auf ihre Grundform zu bringen und miteinander vergleichbar zu machen. Dadurch lassen sich keine Rückschlüsse auf die einzelnen, vorkommenden Wörter mehr schließen, die zur weiteren Analyse der Texte hilfreich sein können, dennoch lassen sich die Gemeinsamkeiten der Dokumente dadurch gut analysieren. Zum anderen werden die für die Lemmatisierung benötigten Part-Of-Speech-Tags [TKMS03, TM00] dazu verwendet um nur Nomen aus dem Text herauszufiltern. Meist sind diese von besonderem Interesse beim Vergleich mehrerer Dokumente, daher wird im Normalfall nach Nomen gefiltert. Durch eine Einstellung lässt sich festlegen, ob nur nach Nomen, nach Nomen und Adjektive oder nach allen Wörtern gesucht werden soll.

Die Anzahl der gezählten Wörter werden in einer Integer-Liste gespeichert, in der jeder Index für ein vorkommendes Wort reserviert wird. Für jede Word-Cloud wird eine solche Liste verwaltet. Parallel dazu wird eine Liste aus Strings mitgeführt, indem die vorkommenden Wörter in der selben Reihenfolge gespeichert werden, wie die dazugehörigen Integer-Werte in den Listen der Word-Clouds. Dadurch lassen sich leicht neue Wörter hinzufügen und zu entsprechenden Gewichtungen den zugehörigen String finden. Zusätzlich zu den oben genannten Restriktionen, die an die zu zählenden Begriffe gestellt werden, existieren noch drei weitere Filter, die auf die zu analysierenden Wörter angewandt werden.

4.1.1. Stopwörter

Ein Filter, der eingebaut ist, ist der Stopwort-Filter. Dabei handelt es sich um eine Liste von Wörtern und Zeichen, die nicht gezählt werden sollen und daher entfernt werden. Neben allgemeinen Satzzeichen werden auch Terme wie „mr.“ herausgefiltert, da ohne Kontext das Wort keinen informativen Gehalt hat. Zusätzlich kann zu jeder Ansammlung von Dokumenten, mit einer Datei `stopwords.txt`, eine spezifische Stopwortliste hinzugefügt werden. Dadurch lassen sich Wörter, die im Allgemeinen keine Stopwörter sind, im entsprechenden Kontext als solche definieren und werden somit in der Textanalyse ignoriert.

4.1.2. Mindestlänge

Ebenso ist ein Faktor einstellbar, der festlegt wie groß ein Wort mindestens sein muss, sodass es gezählt wird. Dieser ist standardmäßig auf drei gesetzt, damit alle Wörter der Länge zwei oder kleiner nicht als Wort gezählt werden. Es lässt sich davon ausgehen, dass die geläufigen englischen Wörter dieser Länge, wie „I“ und „am“, ohne den dazugehörigen Kontext keinen informativen Gehalt haben und daher nicht beachtet werden müssen.

4.1.3. Zahlen-Filter

Als letzter Filter werden noch alle Wörter, die Zahlen enthalten entfernt. Die dadurch gefilterten Zahlen ließen sich nicht mehr in Zusammenhang mit einem Kontext bringen. Außer Jahreszahlen dürfte man keine Zahl miteinander kombinieren, da es sich meist um unterschiedliche Kontexte handelt.

4.2. Bestimmung der Dokumentenreihenfolge

Neben den zu analysierenden Wörtern ist auch die Reihenfolge der Dokumente wichtig. Diese kann auf zwei Arten bestimmt werden. Die Erste sortiert die Dokumente in der Reihenfolge, in der diese im Verzeichnis sortiert sind. Dadurch lässt sich eine Sortierung nach Namen oder auch nach Erscheinungsdatum erzeugen.

Die Zweite bestimmt aus den vorkommenden Wörtern die Reihenfolge. Um möglichst volle Wolken im Inneren zu erhalten, und somit die Redundanz gering zu halten, sollten Dokumente, die viel gemeinsam haben, nahe beieinander stehen, damit deren Inhalt zum Zentrum hin zusammengefasst wird. Dabei wird eine symmetrischen Matrix mit der Ähnlichkeit zwischen Quelle i und Quelle j als Einträge

q_{ij} und q_{ji} erstellt. Ein Beispiel für fünf Dokumente ist in (4.1) gegeben. Auf der Hauptdiagonalen stehen die Einträge −1, damit Dokumente, die mit keinem weiteren Dokument Ähnlichkeit haben, am Ende auch gepaart werden können.

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & -1 & q_{AB} & q_{AC} & q_{AD} & q_{AE} \\ B & q_{AB} & -1 & q_{BC} & q_{BD} & q_{BE} \\ C & q_{AC} & q_{BC} & -1 & q_{CD} & q_{CE} \\ D & q_{AD} & q_{BD} & q_{CD} & -1 & q_{DE} \\ E & q_{AE} & q_{BE} & q_{CE} & q_{DE} & -1 \end{array} \tag{4.1}$$

Die Ähnlichkeit q_{AB} wird mit Hilfe von

$$q_{AB} = \frac{\langle A, B \rangle}{|A|\,|B|}, \tag{4.2}$$

des Kosinus des Winkels zwischen den Vektoren A und B bestimmt.

Der Algorithmus wählt immer den größten Eintrag in der Matrix aus, dadurch werden die beiden dazugehörigen Quellen als Nachbarn definiert. Diese Auswahl wird in einer separaten Matrix gespeichert, mit der am Enden die Dokumente sortiert werden. Wenn in einer Zeile oder Spalte zwei Einträge markiert wurden, werden alle anderen Einträge in der Zeile oder Spalte auf −1 gesetzt, weil dadurch ein Dokument zwei Nachbarn bekommen hat und keine weiteren mehr haben kann.

Der Pseudocode der Implementierung ist in Listing A.1 im Anhang zu finden.

4.3. Kombination der Wort-Werte

Die Wörter, die danach noch übrig bleiben, werden zu den inneren Ebenen kombiniert. Dabei wird eine Menge von nebeneinander liegenden Dokumenten genommen und alle Wörter, die in jeder dieser Dokumente vorkommen zu einem neuen Wert kombiniert. Damit große Dokumente, in denen das Wort durchschnittlich häufiger vorkommt als in Anderen, den neuen Wert nicht dominieren und die Interpretation des Ergebnisses verfälschen, wird der Durchschnitt der Wortvorkommen berechnet und als neuer Wert in die entsprechenden Wolke gespeichert. Dadurch erhält man das durchschnittliche Vorkommen des Wortes in den zugehörigen Dokumenten.

Damit häufig vorkommende Wörter die Größe des Fonts nicht dominieren und alle weiteren Wörter klein und nahezu gleichgroß werden, wird, nachdem die Werte für jedes Wort berechnet wurden, auf alle Wörter der Logarithmus angewandt.

Wenn dadurch mindestens eine vorgegebene Anzahl, standardmäßig vier, an Abstufungen nicht entsteht, wird der Logarithmus nicht angewandt, weil dadurch alle Wörter etwa die gleiche Größe hätten. Zusätzlich werden die genauen Wortvorkommen parallel gespeichert und addiert für die jeweilige Wolke, sodass diese ebenfalls angezeigt werden können.

4.4. Winkelweite der Wolken

Um die Größe eines Dokuments in der Visualisierung anschaulich zu machen werden die Winkelweiten der einzelnen Wolken an die gesamte Größe des Dokuments angepasst. Dabei wird der Umkreis von 2π in zwei gleichgroße Teile aufgeteilt. Die eine Hälfte wird gleichmäßig auf alle Dokumente verteilt, sodass kein Dokument von einem anderen fast vollständig verdrängt wird. Die zweite Hälfte wird Abhängig von der Größe der einzelnen Dokumente zwischen allen Dokumenten verteilt, dadurch bekommen große mehr Platz als kleinere Dokumente. Somit erhält im Worst Case ein Dokument einen Winkelweite von $\pi + \frac{\pi}{n}$ bei n Dokumenten. Der Fall von 6 leeren und einem vollen Dokument ist in Abbildung 4.1 gezeigt. Die innere Ebenen sind ebenfalls leer, da es keine gemeinsamen Wörter gibt und brauchen daher nicht gezeichnet werden.

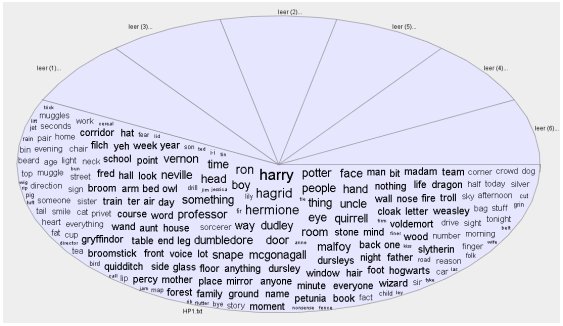


Abbildung 4.1.: Worst-Case-Szenario bei sieben Dokumenten. Sechs Dokumente sind leer und nur eines befüllt. Das eine Dokument bekommt einen Winkel von $\pi + \frac{\pi}{6}$.

4.5. Befüllen einer Word-Cloud

Nachdem die Größe der einzelnen Word-Clouds und deren Inhalt bestimmt wurde, werden die Wörter in ihre Word-Clouds gezeichnet. Hierbei werden vom Zentrum

ausgehend konzentrische Kreise mit zunehmenden Radius nach einer passenden Stelle abgesucht. Durch eine Matrix aus bool'schen Werten wird gewährleistet, dass sich keine Wörter überlappen. Diese Matrix beinhaltet die Information, ob ein Pixel bereits von einem Wort besetzt ist oder nicht. Zusätzlich wird durch das Polygon, das als Hintergrund für die Word-Clouds verwendet wird, überprüft, ob das Wort auch vollständig im Polygon liegt.

Zur Überprüfung, ob ein Wort passt wird dessen Größe benötigt, die mittels `FontMetrics` und dem übergebenen Font berechnet wird. Dabei wird die Größe des Fonts durch die Formel

$$s_{\text{wort}} = \left\lfloor s_{\max} - \frac{p \cdot s_{\max} \cdot (f_{\max} - f_{\text{wort}})}{f_{\max} - f_{\min}} \right\rfloor, \tag{4.3}$$

wobei die Schriftgröße s des Wortes mit Hilfe der Wertung f und dem Verhältnis p zwischen der minimalen und der maximalen Schriftgröße, welches standardmäßig $0,7$ beträgt, bestimmt. Hierbei stehen $s_{\max}$ für die maximale Größe, die Normalfall den Wert 30 beträgt, und $f_{\min}$ und $f_{\max}$ jeweils für die höchste, bzw. kleinste vorkommende Gewichtung, die nicht null ist. Beim Sonderfall, dass jede Gewichtung gleich oft vorkommt, wird die Größe des Fonts auf $\frac{s_{\max}}{4}$ gesetzt, da diese Größe gut zu lesen ist, dennoch die Wörter nicht zu groß werden, sodass nur wenige gezeichnet werden können.

4.6. Awareness

Trotz allen Anpassungen können in den seltensten Fällen alle Wörter gleichzeitig angezeigt werden. Daher sollte dem Benutzer noch die Möglichkeit gegeben werden, sowohl die Anzahl der angezeigten und nicht angezeigten Wörter als auch deren genauen Wortlaute, Gewichtung und Vorkommen zu erhalten. Dazu wird an der rechten Seite eine Liste angezeigt, wie in Abbildung 4.2 gezeigt, die sobald ein Wort in der Word-Cloud angeklickt wird mit allen Wörtern, die in dieser Word-Cloud vorkommen befüllt wird. Diese werden nach Termfrequenz sortiert und die angezeigten Wörter werden blau hervorgehoben. Über der Liste ist die genaue Anzahl der gezeichneten Wörter und der Anzahl an Wörtern insgesamt dargestellt. Darunter wird die Anzahl der übersprungenen Wörter sowohl diskret als auch prozentual angezeigt. Dieser Faktor wird im Failure-Mode visuell als Hintergrundfarbe für die einzelnen Word-Clouds visualisiert, siehe Abbildung 4.3.

Neben der normalen Darstellung gibt es noch einen Failure-Mode, der grafisch die Anzahl der Wörter anzeigt, die übersprungen wurden. Dabei werden die Hin-

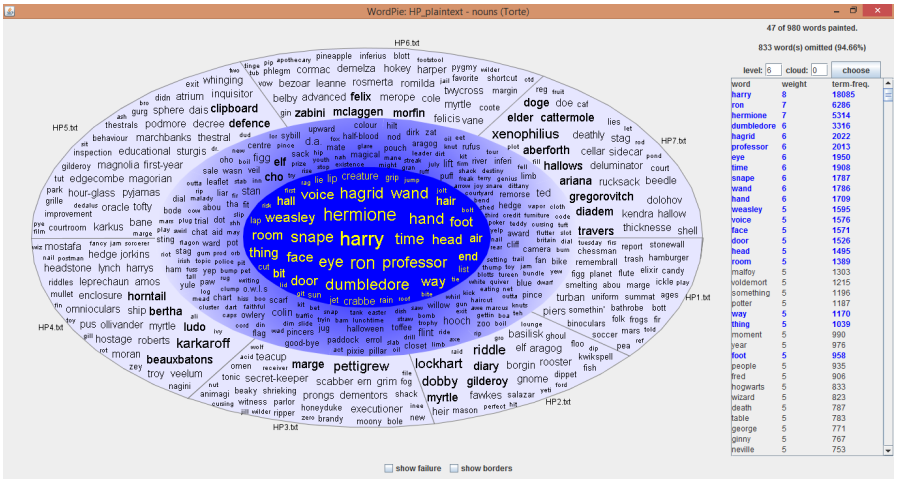


Abbildung 4.2.: Gesamte grafische Oberfläche. Links ist der WordPie zu sehen. In der Liste rechts werden alle Wörter, die in einer Word-Cloud vorkommen aufgelistet. Darüber werden die Anzahl der gezeichneten Wörter, sowie die übersprungenen gezeigt und es lassen sich einzelne Word-Clouds manuell auswählen. Im unteren Bereich lassen sich der Failure-Mode und die Ränder in der mittleren Ebene einstellen.

tergrundfarben der Word-Clouds durch den Faktor

$$f_w = \frac{\#\text{Übersprungene}_w}{\#\text{bisGemalt}_w}, \tag{4.4}$$

wobei #Übersprungene die Anzahl der Wörter ist, die in Word-Cloud w übersprungen wurden, und $\#\text{bisGemalt}_w$ die Anzahl der Wörter bis zum zuletzt gemalten Wort ist. Anhand dieses Faktors wird die Word-Cloud rot, wenn viele Wörter ausgelassen wurden und dadurch der Faktor niedrig ist, oder weiß, wenn fast keine Wörter ausgelassen wurden und der Faktor nahe bei eins liegt, eingefärbt. Dadurch wird ein Überblick über die Anzahl der Wörter verschafft, die gezeigt werden sollten, allerdings aufgrund von Platzmangel nicht gezeichnet werden konnten. In Abbildung 4.3 ist gut zu sehen, dass in dem rechten Teil der Word-Cloud eher weniger große Wörter enthalten sind, als im linken Teil und somit weniger Wörter übersprungen werden mussten.

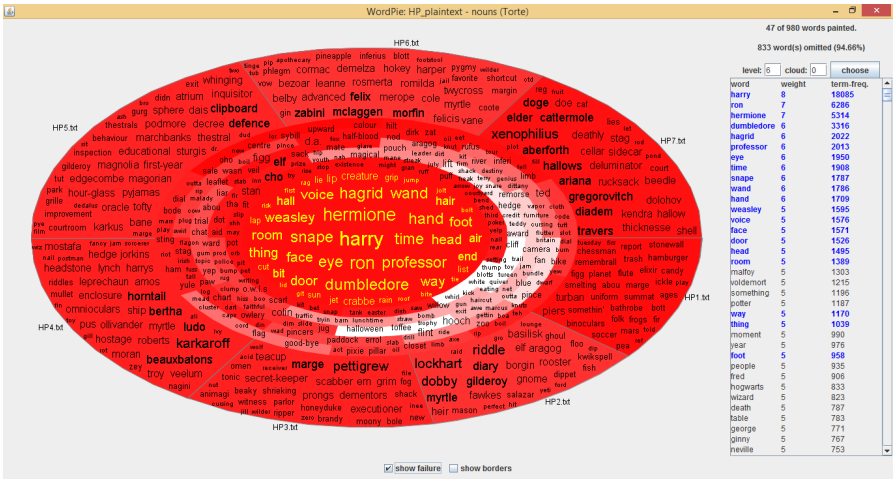


Abbildung 4.3.: Grafische Darstellung der übersprungenen Wörter der jeweiligen Word-Clouds beim Zeichnen, dabei bedeutet eine weiße Word-Cloud, dass keine Wörter übersprungen wurden und eine rote, dass viele Wörter ausgelassen wurden.

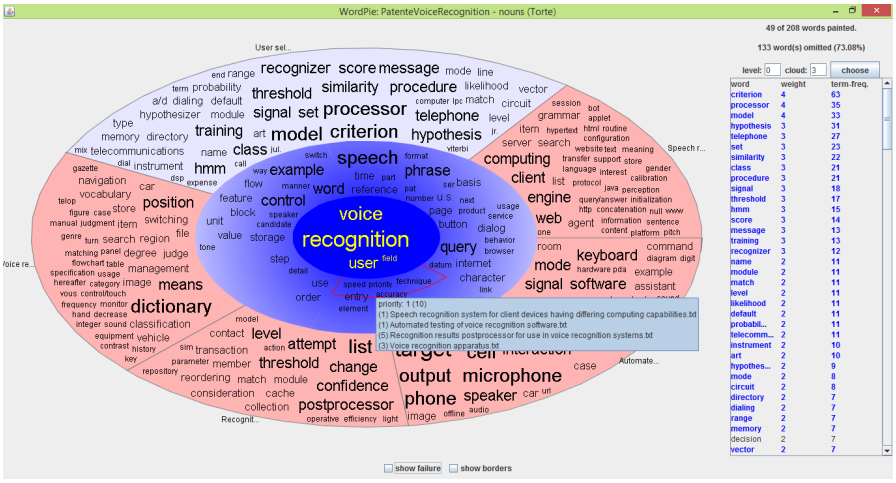


Abbildung 4.4.: Durch die Berührung mit der Maus werden die dazugehörigen Dokumente sichtbar. Ebenfalls wird im Tooltip die Gewichtung, die Wortfrequenz und die dazugehörigen Quellen aufgelistet. Zu den Quellen wird auch die Aufteilung der Termfrequenz auf die einzelnen Dokumente aufgelistet.

4.7. Interaktion

Da in der mittleren Ebene keine eindeutige Unterscheidung zwischen den Word-Clouds mehr möglich ist, wird durch ein interaktives Element die Möglichkeit gegeben die zu einem Wort gehörenden Dokumente herauszufinden. Beim Berühren der Wörter mit der Maus werden, auf der äußersten Ebene, die Word-Clouds der zugehörigen Dokumente mit einer roten Hintergrundfarbe hervorgehoben, wie in Abbildung 4.4 zu sehen. So werden bei Wörtern aus der innersten Ebene alle Word-Clouds markiert und bei Wörter in der äußersten Ebene nur die Word-Cloud selbst.

Im Tooltip wird nochmals das Wort, die Gewichtung und die Termfrequenz dargestellt. Darunter werden die vollständigen Namen der Dokumente aufgelistet. Neben den Namen der Dokumente ist in Klammern die Termfrequenz des Wortes in diesem Dokument vorkommt, dadurch kann leicht unterschieden werden wie wichtig das Wort für die einzelnen Dokumente ist.

5. Anwendungsfälle

Um dieses Konzept mit seiner Implementierung zu evaluieren wurden Anwendungsfälle gesucht und getestet.

5.1. Patente

Patente sind durch Google leicht zu finden und können danach miteinander verglichen werden. Hierbei wurden die Texte direkt aus dem Browser kopiert und die geläufigsten Wörter, die immer enthalten sind, als Stopwörter markiert und dadurch entfernt.

5.1.1. Squirrel

Werden die Patente nach dem Stichwort „Squirrel“ durchsucht, so lassen sich die Treffer in zwei Kategorien einteilen. Einerseits finden sich viele Patente, bei denen es sich um die „Squirrel-Cage-Engine“ handelt. Andererseits ergeben sich einige Patente, bei denen das Objekt in Zusammenhang mit Eichhörnchen eingesetzt wird, wie zum Beispiel ein Vogelhäuschen, dass gegen Eichhörnchen geschützt ist.

In Abbildung 5.1 werden jeweils drei Dokumente der beiden Kategorien miteinander verglichen. Da von zwei grundsätzlich unterschiedlichen Objekten die Rede ist, kommen in der innersten Word-Cloud kaum Wörter vor. Allerdings lassen sich in der mittleren Ebene gut erkennen, dass die drei Dokumente auf der rechten Seite und die drei auf der linken Seite viel miteinander gemeinsam haben. Ebenfalls lassen sich durch die Schlüsselwörter „animal“ und „feeder“ darauf schließen, dass bei dem linken Dokument um das Tier „Squirrel“ und durch die Schlüsselwörter „reactance“ und „ring“ bei dem rechten um eine Maschine handelt.

In der Failure-Mode-Ansicht in Abbildung 5.2 ist gut zu sehen, dass nur wenige Wörter ausgelassen wurden, da die meisten Word-Clouds weiß sind. Nur in der innersten Ebene und den rechten beiden Dokumenten und ihren Kombinationen kommt es vor, dass einige Wörter nicht gezeichnet werden konnten und somit nicht nur die wichtigsten Wörter in der Word-Cloud stehen, sondern auch ein paar ausgelassen wurden. Dies rührt daher, dass nicht ausreichend Platz gefunden wurde, da die Wörter nicht in die anderen Ebenen hineinragen dürfen.

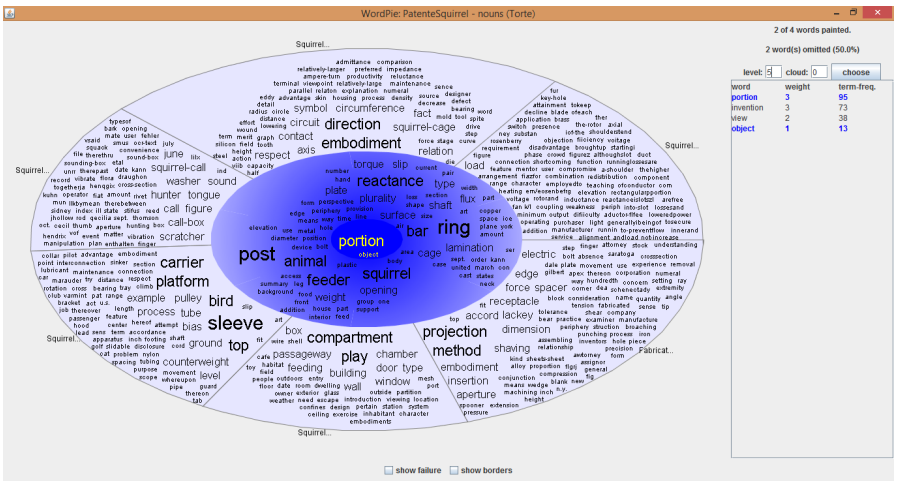


Abbildung 5.1.: Vergleich von sechs Treffern bei Google Patents mit dem Stichwort „Squirrel“.

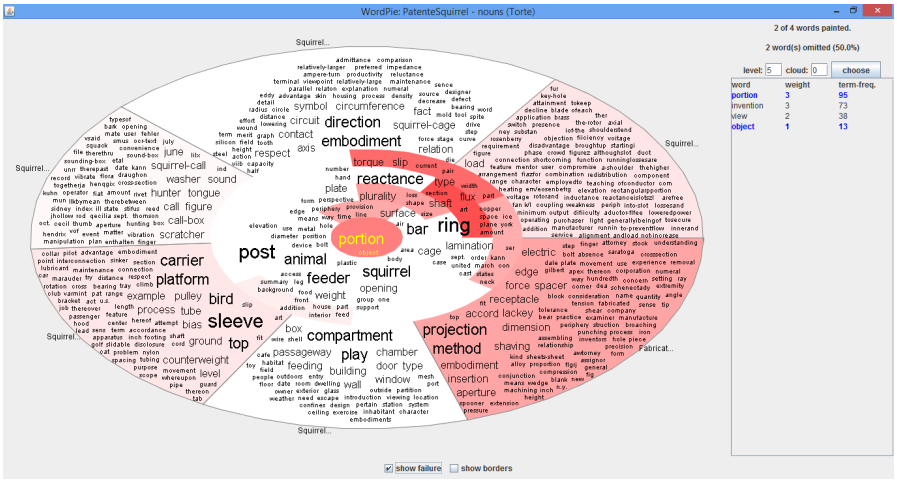


Abbildung 5.2.: Vergleich dreier Treffer bei Google Patents mit dem Stichwort „Squirrel“ in der Failure-Mode-Ansicht, um die Anzahl der übersprungenen Wörter zu analysieren.

5.1.2. Voice Recognition

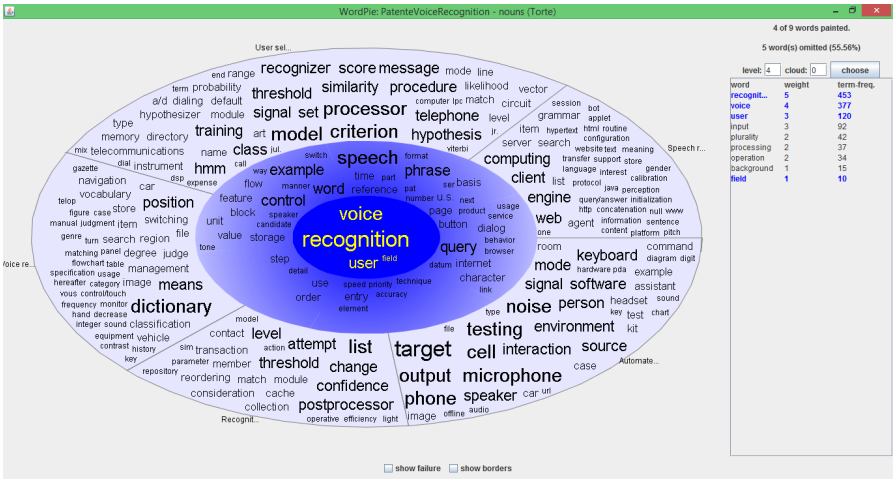


Abbildung 5.3.: Vergleich von sechs Patenten mit dem Stichwort „voice recognition“.

Ein weiterer Anwendungsfall lässt sich bei Patenten mit dem Stichwort „voice recognition“ erkennen. In Abbildung 5.3 lassen die Stichwörter „dictionary“ und „command“ die Interpretation zu, dass sich diese Patente von der Analyse des Gesprochenem handeln. Auf der rechten Seite deuten die Wörter „button“, „target“ und „dialog“ eher auf eine Entwicklung zum Testen oder Anwendenden von Spracherkennung hin.

In der Failure-Mode-Ansicht, siehe Abbildung 5.4, zeigt sich, dass in fast allen Word-Clouds kaum Wörter ausgelassen werden mussten. Allerdings gibt es zwei größere Dokumente, das ganz oben und das unten rechts, bei denen viele höher gewichtete Wörter ausgelassen wurden und dadurch einen roten Hintergrund haben. Um diese explorieren zu können müssten die Liste mit den Wörtern durchsucht werden oder das Dokument alleine als Word-Cloud betrachtet werden. Oberhalb der Wortliste, die zum oberen Dokument gehört, lässt sich ablesen, dass 133 Wörter ausgelassen wurden und das 73.08 % der Wörter entspricht, die insgesamt gezeigt werden sollten.

5.2. Harry Potter

Ein Anwendungsfall für Romane sind die sieben Harry Potter Romane. Durch das Analysieren aller vorkommenden Nomen in allen Bänden kommt ein gefüllter

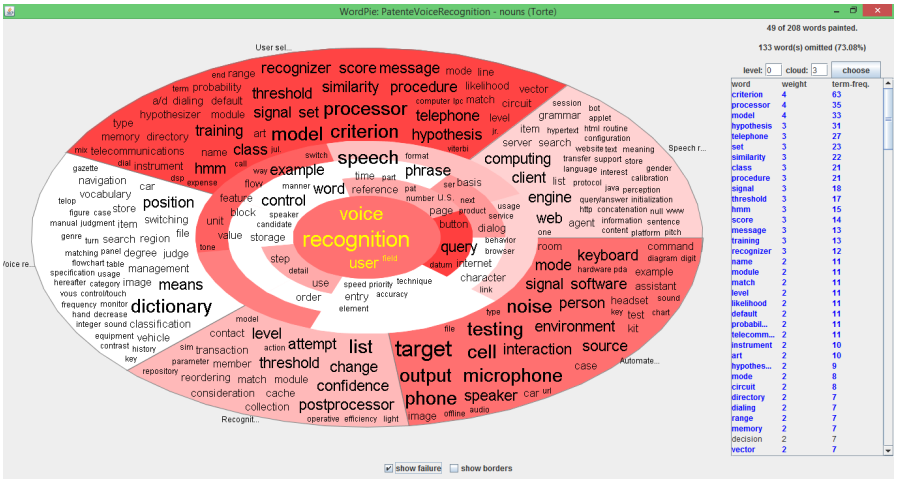


Abbildung 5.4.: Failure-Mode-Ansicht bei dem Vergleich der sechs Patente mit dem Stichwort „voice recognition“.

WordPie zustande. Da die Bände chronologisch aufeinander aufbauen wurden diese in chronologischer Reihenfolge angeordnet. Der WordPie ist bereits in Abbildung 4.2 zu sehen gewesen. In der Liste rechts sind die häufigst vorkommenden Wörter in der mittleren Word-Cloud zu sehen.

Wie erwartet kommen die Namen der Protagonisten, Harry, Ron, Hermoine und Dumbledore, durchschnittlich am Häufigsten vor. Ebenso stechen die wichtigen Personen, die nur in den einzelnen Romanen auftauchen, in den äußeren Word-Clouds hervor. In der mittleren Ebene tauchen aufgrund des geringen Platzes nur wenige große Wörter auf. Aber bei der genaueren Analyse mit Hilfe der Liste lassen sich auch hier passenden Wörter und finden, die zu den jeweiligen Romanen passen. So taucht das Wort „Moody“ in den Romanen vier bis sieben auf, weil die Person erst im vierten Band vorgestellt wird.

5.3. Extremfälle

Neben den erfolgreicheren Anwendungsfällen gibt es auch Extremfälle, die seltener vorkommen. Um diese zu erhalten müssen meist synthetisch erzeugte Anwendungsfälle kreiert werden. Dennoch müssen auch diese in der Implementierung berücksichtigt werden und sollen im folgenden vorgestellt werden.

5.3.1. Keine Gemeinsamkeiten

Wenn es keine gemeinsamen Wörter gibt, so sieht der Word-Cake ähnlich aus wie in Abbildung 4.1. Da keine Kombinationen zu Stande kommen, werden die inneren Word-Clouds nicht angezeigt, da diese nur leer wären und somit nur leeren Raum erzeugen würden. Es sind nur die „Kuchenstücke“ der äußersten Ebene zu sehen, da mindestens in einer Word-Cloud Wörter zu zeichnen sind.

5.3.2. Gleiche Dokumente

Bei gleichen Dokumenten werden alle relevanten Wörter in die innerste Ebene geschrieben, da jedes Wort in jedem Dokument auftaucht. Die restlichen Ebenen bleiben leer, da es kein Wort gibt, dass nicht auch in anderen Dokumenten auftauchen würde. Dadurch wird der gesamte Raum für die innerste Ebene benutzt. Abbildung 5.5 zeigt die Word-Cake für sechs mal den ersten Harry Potter Roman.

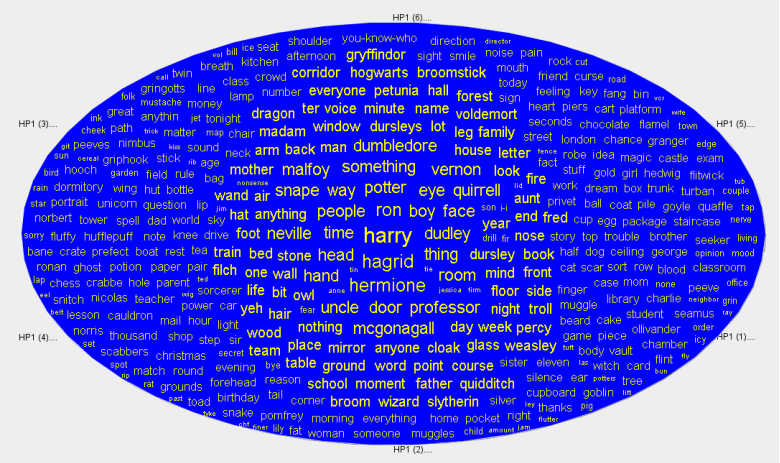


Abbildung 5.5.: Word-Cake für sechs Mal den ersten Harry Potter Roman.

6. Diskussion

Neben den Vorteilen von einigen erfolgreichen Anwendung haben Implementierungen auch Restriktionen, die gewisse Grenzen aufzeigen. Wie in den meisten Word-Clouds ist ein großes Manko, dass nicht alle Wörter angezeigt werden können. Jedoch soll das Bewusstsein über die fehlenden Wörter erhalten bleiben durch die zusätzliche Liste, die alle Wörter, die in diese Word-Cloud gehören, auflistet. Ebenfalls soll durch den Failure-Mode ermöglicht werden eine Übersicht über Anzahl der ausgelassen Wörter zu erhalten.

Ein weiteres Problem ist die Berechnung der Höhe der einzelnen Ebenen. Wenn bei bisherigen Stand eine äußerste Word-Cloud nur ein Wort und die restlichen keine enthalten würden, würde sich dennoch die Höhe der Word-Cloud sich nicht variieren. Dies müsste noch sensitiver an den Inhalt der Word-Clouds angepasst werden.

Ein Nachteil bei der Anordnung nach Ähnlichkeit ist, dass bei Dokumenten, die zu mehr als zwei weiteren Dokumenten eine hohen Ähnlichkeit haben, jede weitere Ähnlichkeit nicht mehr berücksichtigt wird. Allerdings birgt der Anordnungs-Algorithmus ebenso den Vorteil, dass die Dokumente in der Nachbarschaft viel miteinander gemeinsam haben und dadurch die gemeinsamen Wörter in den inneren Ebenen auftauchen. Als weiterer Nachteil können bei der Berechnung der Nachbarn der Dokumente mehrere zusammengehörige Verbände entstehen, die wenig miteinander gemein haben. Diese werden bei der Visualisierung allerdings nicht berücksichtigt, sodass es zu suboptimalen Ergebnissen kommen kann. Im Gegensatz dazu funktioniert die Anordnung nach Erscheinungsdatum, wie beispielsweise bei Romanen möglich ist, im Fall von Harry Potter ganz gut.

6.1. Expertengespräch

Durch eine Evaluation mit Personen, die im Gebiet der Visualisierung forschen, sollen weitere Einsichten erhalten werden. Dabei wurde die Implementierung vom WordPie mit der des Vergleichslayouts verglichen. Beide wurden den Probanden auf einem 15,6" LCD Bildschirm mit der Auflösung 1366 px × 768 px gezeigt. Vor der Einführung in die neue Anwendung wurden die Probenden nach Alter und

Geschlecht gefragt und mit einem Ishihara-Sehtest eine Rot-Grün-Schwäche ausgeschlossen. Anschließend sollten die Probanden ihr Vorwissen zu den Themen „Word-Cloud“, „Textanalyse“ und „Harry Potter“ auf einer Skala zwischen 1 und 10 selbst einschätzen. Anhand des Anwendungsfalls „Voice Recognition“ und den drei einfachen Fragen wurden die Visualisierungen den Probanden erklärt und vorgestellt. Danach wurde zur Visualisierung des WordPies und dem Vergleichslayout mit dem Anwendungsfalls „Harry Potter“ gewechselt und mit jeweils sechs Fragen die Möglichkeiten der Visualisierung gezeigt. Der Fragebogen ist im Anhang A.2 zu finden. In Klammern dahinter sind mögliche richtige Antworten, da beispielsweise Frage A1 nicht eindeutig ist. Um die Abhängigkeit der Reihenfolge zu umgehen wurden bei der Hälfte der Teilnehmer zuerst das Vergleichslayout und danach der WordPie gezeigt. Während der Befragung konnten die Probanden jederzeit eigene Fragen stellen oder Anmerkungen machen. Im anschließenden Interview sollten die positiven und negativen Eindrücke genannt werden. Des weiteren sollte noch eine bevorzugte Visualisierung und fehlende Funktionen genannt werden.

6.1.1. Aufgaben

Zu jeder Visualisierung wurden sechs Fragen gestellt, die einen Überblick über die Möglichkeiten der Visualisierung geben sollen. Zunächst soll das allgemeine Verständnis mit der Frage nach einem gemeinsamen und einem dokumentspezifischen Wort überprüft werden. Da nicht nur die Gemeinsamkeiten von allen Dokumenten interessant sein können, spielt eine Frage auf die Analyse von Teilmengen aller Dokumente an. Abschließend soll das Vorkommen eines Wortes nur in einem Dokument überprüft und die Wahrnehmung über die Anzahl der ausgelassenen Wörter getestet werden.

6.1.2. Auswertung

Von den sechs Probanden waren fünf männlich und eine weiblich, deren Alter sich zwischen 27 und 32 Jahren befanden. Alle Probanden konnten die sechs Zahlen des Ishihara-Sehtests vorlesen. Das Vorwissen zum Thema Word-Cloud hielt sich hauptsächlich im mittleren Bereich auf mit einem Durchschnitt von 5. Zum Thema Textanalyse verschob sich der Mittelwert mit $5,67$ nur minimal ins bessere. Während das Vorwissen zu Harry Potter mit durchschnittlich $6,17$ recht gut war.

Die mittlere Ebene der Visualisierung war zunächst gewöhnungsbedürftig. Nach den ersten Fragen war die Funktionsweise ersichtlich und der Umgang wurde schneller. Auch die Verwendung des Tooltips benötigte einige Gewöhnung und

war durch eine kurze Anzeigezeit nur mühsam zum Ablesen geeignet. Die Verwendung der Wortliste und die Suche nach den passenden Dokument-Kombinationen war schnell gelernt. Letztendlich konnten alle Probanden alle Fragen richtig beantworten. Im anschließenden Interview wurden folgende Vor- und Nachteile genannt.

Jeder Teilnehmer hatte eine Suchfunktion gewünscht, da bei den vielen angezeigten Wörtern die Suche nach einem bestimmten Wort mühsam ist. Ebenfalls sollte die Sortierung der Liste auch auf alphabetisch umgestellt werden können, um spezielle Wörter in der Liste zu suchen. Des weiteren wurde oft nach der Möglichkeit gefragt das Wort im Text anzuzeigen, sodass der Kontext und der Sinn des Wortes nachgeschlagen werden kann, da bei manchen Wörtern nicht klar ist, ob es eine Verunreinigung des Textes oder ein Eigenname ist. Auch das dazugehörige Anzeigen von Koreferenzen wurde von zwei Probanden als nützliches Funktion vorgeschlagen. Durch den fließenden Übergang in den mittleren Ebene ist die Suche nach einem passenden Wort, dass zu einer Kombination aus Dokumenten gehört, schwierig oder unmöglich, da kein Wort daraus angezeigt wird. Darum wurde als weitere Ergänzung zur Interaktion eine Möglichkeit gewünscht worden, dass sich mehrere Dokumente außen markieren ließen und anschließend die von denen gemeinsamen Wörter angezeigt werden. Dennoch macht die Anordnung der Wörter in der mittleren Ebene Sinn und ist nachvollziehbar.

Als eindeutigen Nachteil ist den Probanden der fehlende Hinweis auf redundante Vorkommen eines Wortes in anderen Dokumenten aufgefallen. Das auch Wörter wie „elf“, welche auch in nicht-benachbarten Dokumenten auftauchen, in allen vorkommenden Romanen hervorgehoben werden. Ein weiterer Nachteil, der allerdings nur einem aufgefallen ist, ist die mögliche Dominanz eines Dokuments bei einem Wort, die sich auch nicht durch den Mittelwert vollständig umgehen lässt. So tauchte ein Wort auf, dass in zwei Dokumenten vorkommt, oft in einem Dokument und nur einmal im anderen. Dies könnte umgangen werden indem eine Mindestanzahl, die ein Wort vorkommen muss, setzt und dadurch werden Wörter, die nur selten vorkommen ignoriert.

Im Vergleichslayout hat der Versatz zwischen den Kombinationen der Dokumente gefehlt und war dadurch deutlich weniger intuitiv zu handhaben. Allerdings birgt es mehr Platz für Wörter und ist dadurch besser geeignet, wenn viele Wörter parallel durchsucht werden sollen, allerdings wirkt die Visualisierung dadurch schnell überladen.

Beim WordPie fiel den Nutzern positiv auf, dass gemeinsame Wörter leicht zu erkennen sind und dadurch ein kompakter Überblick über die verglichenen Dokumente zustande kommt. Im Vergleich zwischen den beiden vorgestellten Visualisierungen wurde immer der WordPie bevorzugt. Nach der Eingewöhnung ist er intuitiver, da die mittlere Ebene passend versetzt ist, im Gegensatz zum Ver-

gleichslayout. Gleichzeitig wirkt der WordPie ästhetisch ansprechender und die gemeinsamen Wörter sind „auf einen Blick“ zu erkennen.

7. Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Konzept zum Vergleich vorkommender Wörter mehrerer Dokumente entwickelt. Für die einzelnen Dokumente wurden Word-Clouds radial angeordnet. Zum Zentrum hin wurden alle gemeinsamen Wörter von immer mehr benachbarte Dokumenten in eine neue Word-Cloud separiert. Im Zentrum befindet sich nur noch eine einzige Word-Cloud, die alle Wörter enthält, die in allen Dokumenten auftauchen. Durch diese Anordnung werden schnell die wichtigsten Wörter, die alle Texte verbinden, ersichtlich. Gleichzeitig lassen sich auch die Wörter analysieren, die die jeweiligen Dokumente von den benachbarten unterscheiden. Auch Teilmengen von benachbarten Dokumenten können von Interesse sein und mit diesem Ansatz untersucht werden. Durch die radiale Anordnung kann der Nutzer intuitiv nach der passenden Word-Cloud suchen.

Aus dieser Idee haben sich zwei Konzepte entwickelt, wobei das eine aufgrund von zu vielem leeren Raum nicht weiter verfolgt wurde. Das zweite Konzept, WordPie, ordnet die einzelnen Word-Clouds in überlappenden Kreisstücken an. Dadurch entstehen die einzelnen Ebenen in konzentrischen Kreisen. Um die Platzfindung für die Wörter kulanter zu gestalten, wurden alle Ebenen, außer die äußerste und die innerste, zu einer Ebene verbunden, sodass der WordPie in drei Kategorien eingeteilt ist.

Zur genaueren Analyse werden nur die vorkommenden Nomen gezählt. Mit einer Stopwortliste werden zusätzlich Wörter entfernt, die ohne Kontext keinen informativen Gehalt besitzen oder die für das entsprechende Thema keine Relevanz haben.

Durch interaktive Elemente kann zu jedem Wort die Termfrequenz und die dazugehörigen Dokumente eingesehen werden. Durch eine zusätzliche Liste am Rand können Wörter, die keinen Platz im WordPie gefunden haben, durchsucht werden. Zwei Eingabefelder ermöglichen es spezielle Word-Clouds zu selektieren und in der Liste zu analysieren. Der Failure-Mode ermöglicht es die Anzahl der übersprungenen Wörter als rote Einfärbung der Word-Clouds zu visualisieren.

Durch ein Expertengespräch wurden weitere Vor- und Nachteile der Visualisierung erschlossen. Nach einer Eingewöhnungszeit ist der WordPie benutzerfreund-

lich und hebt die wichtigsten Wörter hervor.

Für die weitere Entwicklung des WordPies ließe eine Suchfunktion leichter Wörter finden, da das in dieser Anordnung eher mühsam ist ein spezielles Wort zu finden. Ebenfalls könnten durch eine solche Suchfunktion Wörter gefunden werden, die aufgrund von Platzproblemen nicht gezeichnet werden konnten. Eine weitere auch im Expertengespräch hilfreiche Ergänzung ist eine Volltextsuche, sodass alle Stellen in denen das Wort vorkommt durchgelesen werden können. Die entstehenden Redundanzen durch gleiche Wörter in nicht benachbarten Dokumenten sollten durch die Interaktion angezeigt werden, sodass kein falscher Eindruck entsteht. Einem Experten kam die Idee, den Ansatz in 3D zu erweitern und dadurch mehr Kombinationen von benachbarten Dokumenten zu ermöglichen.

Literaturverzeichnis

[BGN08] Bateman, Scott ; Gutwin, Carl ; Nacenta, Miguel: Seeing things in the clouds: the effect of visual features on tag cloud selections. In: *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia* ACM, 2008, S. 193–202

[BLB+] Burch, Michael ; Lohmann, Steffen ; Beck, Fabian ; Rodriguez, Nils ; Di Silvestro, Lorenzo ; Weiskopf, Daniel: RadCloud: Visualizing Multiple Texts with Merged Word Clouds.

[BLPW13] Burch, Michael ; Lohmann, Steffen ; Pompe, Daniel ; Weiskopf, Daniel: Prefix tag clouds. In: *Information Visualisation (IV), 2013 17th International Conference* IEEE, 2013, S. 45–50

[BRW13] Burch, Michael ; Raschke, Michael ; Weiskopf, Daniel: Exploring Spatio-Temporal Data Modeled as Dynamic Weighted Relations. In: *KIK@ KI*, 2013, S. 36–43

[CCP09] Collins, Christopher ; Carpendale, Sheelagh ; Penn, Gerald: Docuburst: Visualizing document content using language structure. In: *Computer Graphics Forum* Bd. 28 Wiley Online Library, 2009, S. 1039–1046

[CR84] Cressie, Noel ; Read, Timothy R.: Multinomial goodness-of-fit tests. In: *Journal of the Royal Statistical Society, Series B* 46 (1984), Nr. 3, S. 440–464

[CS95] Coupland, Douglas ; Sakellaropoulou, Christianna: *Microserfs*. Flamingo London, 1995

[CSBT09] Chen, Ya-Xi ; Santamaría, Rodrigo ; Butz, Andreas ; Therón, Roberto: Tagclusters: Semantic aggregation of collaborative tags beyond tagclouds. In: *Smart Graphics* Springer, 2009, S. 56–67

[CVW09] Collins, Christopher ; Viégas, Fernanda B. ; Wattenberg, Martin: Parallel tag clouds to explore and analyze faceted text corpora. In: *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on* IEEE, 2009, S. 91–98

[DCCW08] Dork, Marian ; Carpendale, Sheelagh ; Collins, Christopher ; Williamson, Carey: Visgets: Coordinated visualizations for web-based information exploration and discovery. In: *Visualization and Computer Graphics, IEEE Transactions on* 14 (2008), Nr. 6, S. 1205–1212

[DG92] Deleuze, Gilles ; Guattari, Félix: Tausend Plateaus: Kapitalismus und Schizophrenie. In: *Aufl., Berlin* (1992)

[Dun93] Dunning, Ted: Accurate methods for the statistics of surprise and coincidence. In: *Computational linguistics* 19 (1993), Nr. 1, S. 61–74

[Fei09] Feinberg, Jonathan: *Wordle-Beautiful Word Clouds*. 2009

[FFM+08] Fujimura, Ko ; Fujimura, Shigeru ; Matsubayashi, Tatsushi ; Yamada, Takeshi ; Okuda, Hidenori: Topigraphy: visualization for large-scale tag clouds. In: *Proceedings of the 17th international conference on World Wide Web* ACM, 2008, S. 1087–1088

[GV10] Gambette, Philippe ; Véronis, Jean: Visualising a text with a tree cloud. In: *Classification as a Tool for Research*. Springer, 2010, S. 561–569

[Har11] Harris, Jacob: *Word cloud considered harmful*. http://www.niemanlab.org/2011/10/word-clouds-considered-harmful/, October 2011. – zuletzt gesehen: 2014-10-08

[HR08] Hearst, Marti A. ; Rosner, Daniela: Tag clouds: Data analysis tool or social signaller? In: *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual* IEEE, 2008, S. 160–160

[LRKC10] Lee, Bongshin ; Riche, Nathalie H. ; Karlson, Amy K. ; Carpendale, Sheelagh: Sparkclouds: Visualizing trends in tag clouds. In: *Visualization and Computer Graphics, IEEE Transactions on* 16 (2010), Nr. 6, S. 1182–1189

[LZT09] Lohmann, Steffen ; Ziegler, Jürgen ; Tetzlaff, Lena: Comparison of tag cloud layouts: Task-related performance and visual exploration. In: *Human-Computer Interaction–INTERACT 2009*. Springer, 2009, S. 392–404

[RGMM07] Rivadeneira, A W. ; Gruen, Daniel M. ; Muller, Michael J. ; Millen, David R.: Getting our head in the clouds: toward evaluation studies of tagclouds. In: *Proceedings of the SIGCHI conference on Human factors in computing systems* ACM, 2007, S. 995–998

[SM83] Salton, Gerard ; McGill, Michael J.: Introduction to modern information retrieval. (1983)

[Ste07] STEFANER, Moritz: Visual tools for the socio-semantic web. In: *Master's Thesis, University of Applied* (2007)

[TKMS03] TOUTANOVA, Kristina ; KLEIN, Dan ; MANNING, Christopher D. ; SINGER, Yoram: Feature-rich part-of-speech tagging with a cyclic dependency network. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* Association for Computational Linguistics, 2003, S. 173–180

[TM00] TOUTANOVA, Kristina ; MANNING, Christopher D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13* Association for Computational Linguistics, 2000, S. 63–70

[woh] WOHNUNGBOERSE.NET: *Wohnungsboerse.net*. http://www.wohnungsboerse.net/searches/index/marketing_type:miete/object_type:1/state:1/cities:972, . – zuletzt gesehen: 2014-10-08

[Wor12] WORDPRESS.ORG: *WP-Cumulus*. http://wordpress.org/plugins/wp-cumulus/, November 2012. – zuletzt gesehen: 2014-10-08

[Zel05] ZELDMAN, Jeffrey: *Tag clouds are the new mullets*. http://www.zeldman.com/daily/0405d.shtml, April 2005. – zuletzt gesehen: 2014-10-08

A. Anhang

A.1. Bestimmung der Reihenfolge

Listing A.1: Bestimmung der Reihhenfolge der Quellen und Berücksichtigung der Abhängigkeit der direkten Nachbarn.

```
1  /**
2       * Berechnet die passende Reihenfolge für die übergebenen Vektoren.
3       * Die erste Quelle ist immer alle[0].
4       *
5       * @param alle
6       *            Die zu sortierenden Vektoren
7       * @return Gibt die sortierten Vektoren zurück.
8       * @Laufzeit: worst case O(Dokumente^3)
9       */
10     public Vector<Vector<Integer>> berechneReihenfolge(
11             Vector<Vector<Integer>> alle) {
12         Vector<Vector<Integer>> toRet = new Vector<Vector<Integer>>();
13
14         // Für 3 oder weniger Dokumente lässt sich keine Reihenfolge festlegen.
15         if (alle.size() < 4)
16             return alle;
17
18         // Boolean-Matrix, zum Markieren, welche Vektoren nebeneinander liegen
19         boolean[][] zusammengehoerige = new boolean[alle.size()][alle.size()];
20
21         int maxI = 0, maxJ = 0;
22         double max;
23         while (true) {
24
25             // Größtes Element suchen (O(n^2))
26             max = getGroesstesElement();
27             maxI = max.i;
28             maxJ = max.j;
29
30             // Wenn keins gefunden, dann fertig
31             if (max == -1.0) {
32                 break;
33             }
34
35             // Werte auf -1 setzen und in zusammengehörige-Matrix schreiben
36             abhaenge[maxI][maxJ] = -1.0;
37             abhaenge[maxJ][maxI] = -1.0;
38             zusammengehoerige[maxI][maxJ] = true;
39             zusammengehoerige[maxJ][maxI] = true;
40
41             // Falls ein Vektor schon zwei Nachbarn hat,
42             // dann alle weiteren Nachbarn entfernen
43             // Zeile
44             for (int j = 0; j < abhaenge.length; j++) {
45                 // falls eine weitere ein weiteres zusammengehöriges existiert
46                 if (zusammengehoerige[maxI][j] && j != maxJ) {
47                     // Abhänge zwischen den getrennten Nachbarn löschen
48                     abhaenge[j][maxJ] = -1.0;
49                     abhaenge[maxJ][j] = -1.0;
50                     // Abhänge in der gesamten Zeile und Spalte löschen
51                     for (int j2 = 0; j2 < abhaenge.length; j2++) {
52                         abhaenge[maxI][j2] = -1.0;
53                         abhaenge[j2][maxI] = -1.0;
54                     }
55                     break;
56                 }
57             }
```

```
58              // Spalte
59              for (int i = 0; i < abhaenge.length; i++) {
60                  // falls eine weitere ein weiteres zusammengehöriges existiert
61                  if (zusammengehoerige[i][maxJ] && i != maxI) {
62                      // Abhänge zwischen den getrennten Nachbarn löschen
63                      abhaenge[i][maxI] = -1.0;
64                      abhaenge[maxI][i] = -1.0;
65                      // Abhänge in der gesamten Zeile und Spalte löschen
66                      for (int i2 = 0; i2 < abhaenge.length; i2++) {
67                          abhaenge[i2][maxJ] = -1.0;
68                          abhaenge[maxJ][i2] = -1.0;
69                      }
70                      break;
71                  }
72              }
73          }
74
75          // Vektoren zusammenfügen
76          int i = 0;
77          boolean nichtsGefunden = false;
78          // O.B.d.A. erstes Element an den Anfang hängen
79          toRet.add(alle.get(i));
80          // Bis alle Vektoren hinzugefügt wurden
81          while (toRet.size() != alle.size()) {
82              for (int j = 0; j < alle.size(); j++) {
83                  nichtsGefunden = true;
84                  // wenn ein Nachbar gefunden wurde
85                  if (zusammengehoerige[i][j]) {
86                      // Nachbar anhängen
87                      toRet.add(alle.get(j));
88                      // Zusammengehörigkeit löschen, um wiederholte Treffer zu vermeiden
89                      zusammengehoerige[i][j] = false;
90                      zusammengehoerige[j][i] = false;
91                      // In neuer Spalte suchen
92                      i = j;
93                      nichtsGefunden = false;
94                      break;
95                  }
96              }
97              // Sollte nichts mehr in der Spalte zu finden sein in der nächsten Zeile suchen (Kann bei getrennten Abhängigkeiten auftreten)
98              if (nichtsGefunden) {
99                  // Falls das Dokument mit keinem Ähnlichkeit besitzt
100                 if (!toRet.contains(alle.get(i))) {
101                     toRet.add(alle.get(i));
102                 }
103                 i = (i + 1) % alle.size();
104             }
105         }
106         return toRet;
107     }
```

A.2. Fragebogen Expertengespräch

Bachelorarbeit Informatik | Expertengespräch-Word-Cloud | 01.10.2014

Alter:

Geschlecht:

Ishihara:

Vorwissen zum Thema Word/Tag-Clouds:

O O O O O O O O O O

1 2 3 4 5 6 7 8 9 10

Kein Ahnung gute Kenntnisse

Vorwissen zum Thema Textanalyse:

O O O O O O O O O O

1 2 3 4 5 6 7 8 9 10

Keine Ahnung gute Kenntnisse

Vertrautheit mit Harry Potter:

O O O O O O O O O O

1 2 3 4 5 6 7 8 9 10

Nie gelesen/gesehen alle gelesen

Gruppe:

Bachelorarbeit Informatik | Expertengespräch-Word-Cloud | 01.10.2014

A) Radial:

Welches Wort kommt in allen Romanen vor? (Harry)

Welches Wort kommt am häufigsten nur im 3. Roman (HP3) vor? (pettigrew)

Zu welchen Romanen gehört das Wort "elf"? (4,5,6,7)

Welches Wort kommt durchschnittlich am häufigsten in den Romanen 1 und 2 vor? (Justin)

Wie viele Wörter kommen in der innersten Word-Cloud vor und wie viele werden angezeigt? (980, 47)

Wie oft kommt „Harry" in Roman 1 vor? (1306)

B) Vergleich:

Welches Wort kommt am Seltensten in allen Romanen vor? (fighting)

Zu welchen Romanen gehört das Wort "griphook"? (1,7)

Welches Wort kommt am häufigsten nur im 2. Roman (HP2) vor? (lockhart)

Welches Wort kommt durchschnittlich am häufigsten in den Romanen 3, 4 und 5 vor? (boggart)

Wie viele Wörter kommen nur im ersten Roman (HP1) vor und wie viele werden davon angezeigt? (948, 59)

Wie oft kommt „Ron" in Roman 4 (HP4) vor? (1042)

Notizen:

Gruppe:

Bachelorarbeit Informatik | Expertengespräch-Word-Cloud | 01.10.2014

Einführung:

Zu welchen Dokumenten gehört „Control"?

Wie oft kommt Recognition vor?

Wie oft kommt „user" in „voice Recognition Apparatus" vor?

Lösung Ishihara:

3, 15, 74, 6, 45, 5

Gruppe:

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben.
Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet.
Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens.
Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht.
Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Unterschrift:

Stuttgart, 13.10.2014

Declaration

I hereby declare that the work presented in this thesis is entirely my own.
I did not use any other sources and references that the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations.
Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before.
The electronic copy is consistent with all submitted copies.

Signature:

Stuttgart, 13.10.2014