Institut für Visualisierung und Interaktive Systeme
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diplomarbeit Nr. 3352

Interaktiver Ansatz für die visuelle Analyse von Textdokumenten basierend auf der Word-Cloud-Visualisierungstechnik

Simon Lange

Studiengang:	Informatik
Prüfer:	Prof. Dr. Thomas Ertl
Betreuer:	Steffen Lohmann, M. Sc. Dipl. Ling. Florian Heimerl
begonnen am:	14. Juli 2012
beendet am:	11. Januar 2013
CR-Klassifikation:	H.3.3, H.5.2, I.2.7

Kurzfassung

Das Konzept der Word-Cloud hat sich während des vergangenen Jahrzehnts im Internet etabliert und erfreut sich großer Popularität. Hinsichtlich der visuellen Analyse können Word-Clouds hilfreich sein, um dem Analysten einen ersten Eindruck vom Inhalt eines Textes zu vermitteln. Jedoch sind die Funktionalitäten herkömmlicher Word-Clouds stark beschränkt, da ihr primärer Fokus üblicherweise der Ästhetik gilt. Basierend auf der Word-Cloud-Idee wird in dieser Arbeit eine Visualisierung entwickelt, die eine Reihe interaktiver Funktionalitäten zur Unterstützung der visuellen Analyse von Textdokumenten anbietet. Hierbei werden Techniken der maschinellen Sprachverarbeitung mit Visualisierungs- und Interaktionsansätzen verknüpft, um einen komplett neuen Ansatz der visuellen Analyse zu ermöglichen. Um diesen Ansatz auf die Probe zu stellen sowie Feedback und Verbesserungsvorschläge zu erhalten, wurde im Anschluss an die Implementierung eine Nutzerstudie mit dem entstandenen Programm durchgeführt. Das Resultat dieser qualitativen Evaluation bestätigte die intuitive Bedienbarkeit und den Nutzen hinsichtlich bestimmter Aufgaben der visuellen Analyse von Textdokumenten. Unter anderem ist es möglich, ein beliebiges englischsprachiges Textkorpus mithilfe dieses Programms nach Wortarten oder Kategorien wie beispielsweise Personen oder Orten zu filtern und daraus eine Word-Cloud zu generieren. Darüber hinaus können explorativ Zusammenhänge zwischen einzelnen oder mehreren Wörtern ermittelt und in der Word-Cloud visualisiert werden. Neben diesen und vielen weiteren linguistischen Analysetechniken stehen dem Anwender eine Vielzahl interaktiver Einstellungsmöglichkeiten hinsichtlich der Word-Cloud zur Verfügung, die ihn bei einer visuellen Analyse unterstützen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis der Listings

Verzeichnis der Algorithmen

1. Einleitung

Das Konzept der Word-Cloud hat sich während des vergangenen Jahrzehnts im Internet etabliert und erfreut sich großer Popularität. Als Word-Cloud wird eine Visualisierungsmethode bezeichnet, bei der beliebige Schlagworte meist alphabetisch sortiert dargestellt werden. Hierbei korreliert die Schriftgroße der einzelnen Schlagworte mit der Häufigkeit ihrer Vorkommen innerhalb des zugrunde liegenden Datensatzes. Hinsichtlich der visuellen Analyse von Textdokumenten können Word-Clouds hilfreich sein, um einen ersten Eindruck vom Inhalt eines Textes zu vermitteln.

Neben den Worthäufigkeiten bieten Word-Cloud-Visualisierungen die Möglichkeit, weitere Informationen – beispielsweise Zusammenhänge zwischen Worten – darzustellen, welche den Analysten bei seiner Aufgabe unterstützen können. Um diese Informationen aus den zugrunde liegenden Textdokumenten zu extrahieren, werden Techniken der maschinellen Sprachverarbeitung benötigt.

In einigen Word-Cloud-Visualisierungen kommen bereits Techniken der maschinellen Sprachverarbeitung zum Einsatz, üblicherweise für die Bereinigung von Textkorpora. Die Integration solcher Techniken in die Word-Cloud-Visualisierung dieser Arbeit soll zahlreiche Funktionalitäten für die visuelle Analyse bereitstellen und den üblichen Verwendungszweck bei weitem übertreffen.

Eine visuelle Analyse von Textdokumenten ist von diversen Faktoren abhängig – etwa der Fachrichtung des Textkorpus und der speziellen Aufgabe des Analysten – weshalb Interaktionsmöglichkeiten eine hohe Priorität für diese Arbeit haben, um dem Analysten Flexibilität zu bieten.

Da die üblicherweise verwendeten Word-Cloud-Visualisierungen jedoch recht statisch sind, bieten sie neben einem Textüberblick kaum Funktionalitäten für eine interaktive visuelle Analyse von Textdokumenten. An eben dieser Stelle setzt die vorliegende Diplomarbeit an, um eine statische Word-Cloud in ein interaktives Werkzeug der visuellen Analyse zu verwandeln.

1.1. Aufgabe und Lösungsansatz

Das Ziel dieser Arbeit ist es, die begrenzten Möglichkeiten von Word-Cloud-Visualisierungen für die visuelle Analyse zu erweitern. Basierend auf der Word-Cloud-Idee, soll hierzu eine Visualisierung entwickelt werden, die eine Reihe interaktiver Funktionalitäten zur Unterstützung der visuellen Analyse von Textdokumenten anbietet. Hierbei sollen Techniken der

maschinellen Sprachverarbeitung mit Visualisierungs- und Interaktionsansätzen verknüpft werden und sich auf beliebige englischsprachige Texte anwenden lassen. Am Ende des Projekts soll eine Sprachverarbeitungspipeline für die Vorverarbeitung von Textdokumenten und eine um interaktive Funktionalitäten erweiterte Word-Cloud-Visualisierung entstehen.

Ein interaktiver Word-Cloud-Ansatz setzt eine intensive Vorverarbeitung der zugrunde liegenden Textkorpora voraus. Hierfür sollen zunächst bestehende Techniken der maschinellen Sprachverarbeitung sowie Visualisierungs- und Interaktionsansätze betrachtet und verglichen werden. Dabei sollen nützliche Konzepte und Ansätze als Inspiration für die Konzeption der Funktionalitäten dienen. Diese Funktionalitäten beinhalten unter anderem die Zusammenfassung verschiedener Wortformen sowie die Erkennung von Wortarten und Kategorien wie beispielsweise Personen oder Orte. Darüber hinaus sollen Zusammenhänge zwischen Worten in der Word-Cloud-Visualisierung dargestellt werden.

Um den Ansatz interaktiv zu gestalten, ist es notwendig, dass alle konzipierten Funktionalitäten die betreffenden Daten in angemessener Zeit bereitstellen, um dem Analysten ein effizientes Arbeiten zu ermöglichen. Die Herausforderung besteht folglich darin, einen sinnvollen Mittelweg zwischen einer intensiven Vorausberechnung aller Daten und einer spontanen Berechnung der Daten, sobald diese benötigt werden, zu finden. Da einige Funktionaliäten viel Rechenzeit und/oder Arbeitsspeicher benötigen und nur für spezielle Aufgaben einen Mehrwert bieten, ist eine optionale Verwendung hilfreich, um unnötige Verzögerungen zu vermeiden und eine Verwendung des Ansatzes auf Rechnern mit geringen Hardware-Ressourcen zu ermöglichen.

1.2. Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2: Die *Grundlagen* vermitteln das für das Verständnis der Arbeit benötigte Fachwissen. Darüber hinaus werden in diesem Kapitel verwandte Arbeiten vorgestellt und diskutiert.

Kapitel 3: Das *Konzept* wird in diesem Kapitel erarbeitet und erläutert. Grundlage für das Konzept stellen die Erkenntnisse aus den verwandten Arbeiten und die bisher existierenden Ansatze dar.

Kapitel 4: Der *eigene Ansatz* beginnt mit einem Überblick über das Programm. Bevor die Funktionalitäten des Programms erläutert und veranschaulicht werden, werden zunächst die wichtigsten Datenstrukturen vorgestellt. Anschließend werden die Probleme der Implementierung diskutiert.

Kapitel 5: Die *Evaluation* des Programms findet in Form einer Benutzerstudie statt. Zunächst werden die Vorbereitungen, Materialien, Aufgaben sowie der Ablauf der Studie erklärt, anschließend werden die Ergebnisse präsentiert und diskutiert.

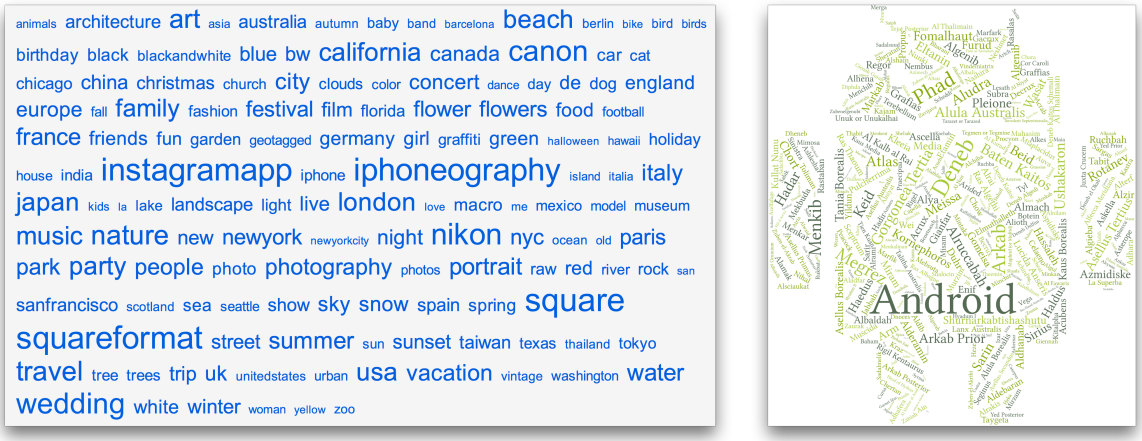
Kapitel 6: In Form einer *Diskussion* und eines *Ausblicks* werden die Ergebnisse der Arbeit zusammengefasst, diskutiert und Anknüpfungspunkte dargestellt.

2. Grundlagen

In diesem Kapitel werden die grundlegenden Konzepte von Word-Clouds erläutert. Anhand verwandter Arbeiten wird der aktuelle Stand existierender Word-Cloud-Visualisierungen und -Ansätze präsentiert und diskutiert. Anschließend werden relevante Techniken der natürlichen Sprachverarbeitung erläutert, welche im Rahmen dieser Arbeit Anwendung finden sollen.

2.1. Word-Clouds

Als Tag- oder Word-Cloud wird eine Visualisierungsmethode bezeichnet, bei der beliebige Schlagworte, genannt „Tags“, meist alphabetisch sortiert dargestellt werden. Hierbei korreliert die Schriftgröße der einzelnen Tags mit deren Häufigkeit. Auf diese Weise können die beiden Ordnungsdimensionen „alphabetische Sortierung“ und „Häufigkeit“ gleichzeitig dargestellt werden [sch12]. Neben der Worthäufigkeit existieren weitere Ansätze der Rangordnungsbestimmung, jedoch ist die Worthäufigkeit gut dafür geeignet [SA11] und wird aufgrund ihrer Einfachheit häufig verwendet. Bekannt wurde die Visualisierungsmethode durch die Foto-Netzgemeinschaft Flickr [fli] und andere soziale Netzwerke, in denen Tags



(a) „Die beliebtesten Tags aller Zeiten“ von Flickr [fli]

(b) Auf Ästhetik fokussierte Word-Cloud von Tagul [tag]

Abbildung 2.1.: Beispielhafte Word-Clouds

verwendet werden [VW08, LZT09]. Eine beispielhafte Word-Cloud der beliebtesten Tags aller Zeiten von Flickr, sowie eine auf Ästhetik fokussierte Word-Cloud von Tagul [tag] sind in Abbildung 2.1 dargestellt.

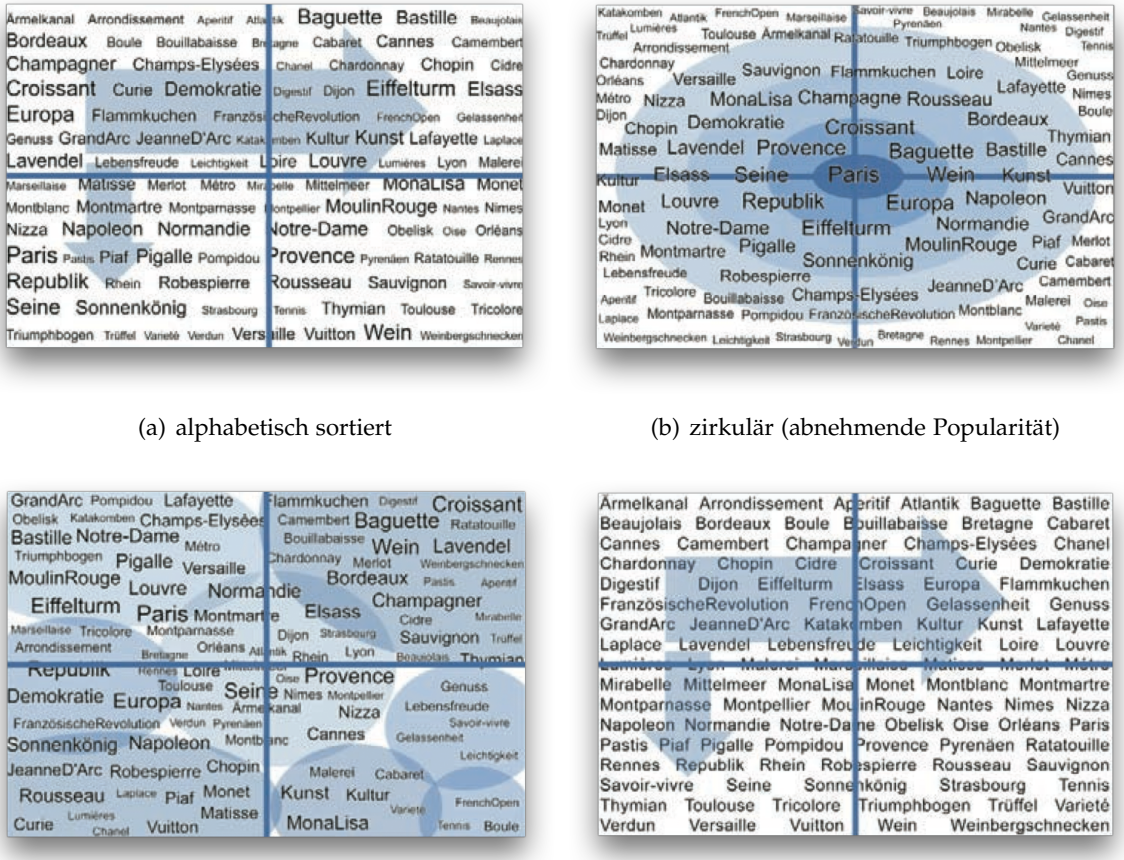
Word-Clouds kommen meist für folgende Aufgaben zum Einsatz: [RGMM07]

- „Searching“: Das Suchen eines bestimmten Begriffs (oder die Feststellung, dass der Begriff nicht vorhanden ist), oft als Mittel, zu den zugrunde liegenden Inhalten zu navigieren.
- „Browsing“: Das Surfen im Internet, oft ohne bestimmtes Ziel oder Thema im Auge.
- „Impression Formation or Gisting“: Die Betrachtung einer Word-Cloud mit dem Ziel, einen allgemeinen Eindruck eines zugrunde liegenden Datensatzes zu gewinnen. Dieser Eindruck sollte sowohl ein Bewusstsein für die häufigsten wie auch die weniger häufigen Themen beinhalten.
- „Recognition or Matching“: Die Erkennung, welche Informationen von verschiedenen Informationssätzen am Wahrscheinlichsten in einer Word-Cloud angezeigt werden, um beispielsweise zu entscheiden, welcher von zwei John Smiths derjenige ist, den man bei einer Konferenz getroffen hat, allein anhand ihrer persönlichen Word-Clouds.

Neben Größe und Position der dargestellten Wörter werden in manchen Word-Clouds weitere visuelle Eigenschaften beeinflusst. Bateman et al [BGN08] untersuchten neun visuelle Eigenschaften (Font Size, Font Weight, Colour, Intensity, Number of Pixels, Tag Width, Number of Characters, Tag Area, Position), welche in einer Word-Cloud variiert werden können, sowie deren Auswirkung. Ergebnis der Studie war, dass eine große Schriftgröße sowie eine hohe Schriftstärke (fett) und die Farbe der Wörter den größten Einfluss auf Benutzer haben. Der Position der Wörter wurde ebenfalls eine große Rolle zugeschrieben, was durch die Eye-Tracking-basierte Studie von Lohmann et al. [LZT09] ebenfalls untersucht und bestätigt werden konnte. Demnach wird die linke obere Ecke sowie der mittlere Bereich einer Word-Cloud am längsten fokussiert.

Darüber hinaus wurden in dieser Nutzerstudie unterschiedliche Layouts untersucht, welche in Abbildung 2.2 dargestellt sind. Um die Effizienz der Layouts zu ermitteln, hatten die Teilnehmer drei Aufgaben zu bearbeiten und sollten anschließend für jede Aufgabe angeben, welches der vier Layouts sie bevorzugten. Interessanterweise wurde bei jeder Aufgabe ein anderes Layout favorisiert, was eine starke Aufgabenabhängigkeit der Word-Cloud-Layouts nahelegt.

Im Gegensatz zu den durch eine Nutzergemeinschaft vergebenen Tags liegt der Fokus dieser Arbeit auf Tags, welche automatisch aus einem Textkorpus extrahiert werden. Da ein Textkorpus überwiegend aus Wörtern besteht und das Hauptinteresse für die visuelle Analyse auf den Wörtern liegt, wird im Folgenden die Begrifflichkeit Word-Cloud anstelle von Tag-Cloud verwendet.



(a) alphabetisch sortiert

(b) zirkulär (abnehmende Popularität)

(c) thematisch gruppiert

(d) alphabetisch sortiert, ohne Gewichtung

Abbildung 2.2.: Unterschiedliche Layouts einer Word-Cloud [LZT09]
Hierbei stellen die blauen Linien die Grenzen der Quandranten dar, während die blauen Pfeile und Kreise das Ordnungsprinzip verdeutlichen.

2.2. Verwandte Arbeiten

Im Folgenden werden bereits existierende Programme und verwandte Arbeiten aufgeführt, die wesentlich zur Konzeption dieser Arbeit beigetragen haben. Anschließend werden weitere Ansätze, welche ähnliche Ziele verfolgen, diskutiert.

2.2.1. Wordle - Beautiful Word Clouds

Ein sehr bekanntes Beispiel für individuelle und kreative Word-Clouds nennt sich „Wordle - Beautiful Word Clouds“ [Fei11]. Wordle wurde im Juni 2008 von Jonathan Feinberg entwickelt und ist als Java-Applet frei verfügbar im Internet zu finden. Der Fokus des Applets liegt

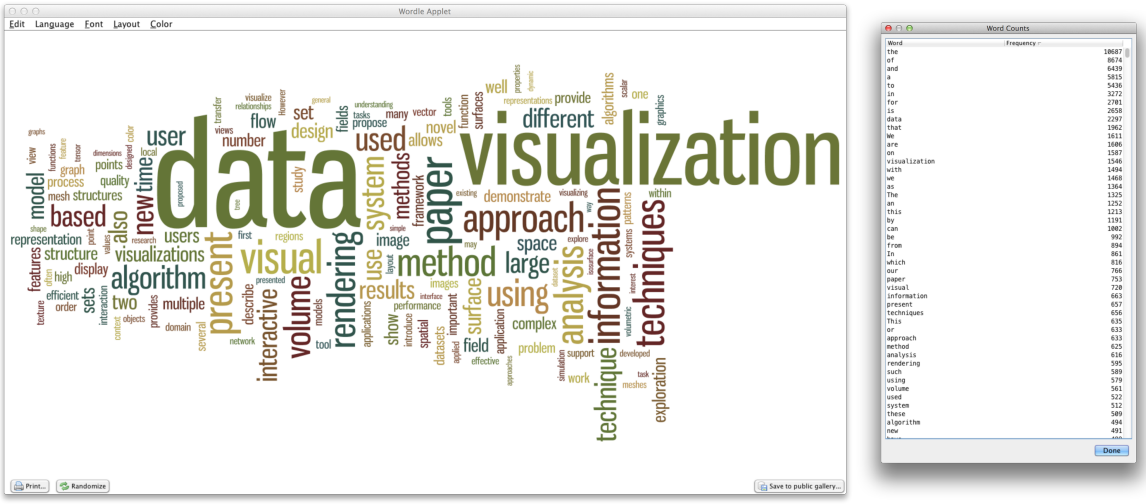


Abbildung 2.3.: Von Wordle erzeugte Word-Cloud mit Häufigkeitsverteilung [Fei11]

auf dem ästhetischen Layout [VWF09], welches zufällig generiert oder interaktiv angepasst werden kann. Hierbei können beispielsweise Farbe, Ausrichtung der Wörter im Raum und die Schriftart variiert werden. Hat der Benutzer einen Text eingegeben und ist mit dem gewählten Layout zufrieden, kann die erzeugte Word-Cloud an Freunde verschickt, ausgedruckt oder gespeichert werden. In Abbildung 2.3 ist eine von Wordle erzeugte Word-Cloud zu sehen, welcher eine Konkatenation von Abstracts aus Veröffentlichungen der VisWeek (Zeitraum: 1998-2011) [vis, HKBE12, S. 6] zugrunde liegt. Um den Text zu bereinigen, errät Wordle die Sprache des Textes und entfernt Stoppwörter, die hier als „common words“ bezeichnet werden und in Abschnitt 3.4 ausführlich beschrieben werden. Die Sprache der Stoppwörter kann ebenfalls manuell ausgewählt werden oder es kann gänzlich auf die Entfernung der Stoppwörter verzichtet werden. Darüber hinaus ist es möglich, die Worthäufigkeiten einzusehen, was ebenfalls in Abbildung 2.3 dargestellt ist. Die Anzahl der in der Word-Cloud angezeigten Wörter kann durch ein selbst definiertes Maximum begrenzt werden. Wordle bietet darüber hinaus einen Fortgeschrittenen-Modus an, bei welchem ein bereits verarbeiteter Text in Form von Worten mit zugehöriger Gewichtung und Farbe eingegeben und visualisiert werden kann. Die Texteingabe kann hierbei ebenfalls mittels POST-Request [FGM+99] erfolgen.

Ziel dieses Applets ist folglich weniger eine interaktive visuelle Analyse des eingegebenen Textes, als mehr die Erzeugung einer optisch ansprechenden jedoch statischen Word-Cloud. Dennoch stellt die Vorverarbeitung des Textes, insbesondere die Stoppwortentfernung, einen interessanten Aspekt dar.

2.2.2. Peter Holme's Word Stemmer

Ein ebenfalls interessanter Ansatz zur Word-Cloud-Generierung stammt von Peter Holme [Hol11]. Er macht sich den Porter Stemmer zunutze, der in Abschnitt 3.8 vorgestellt wird.

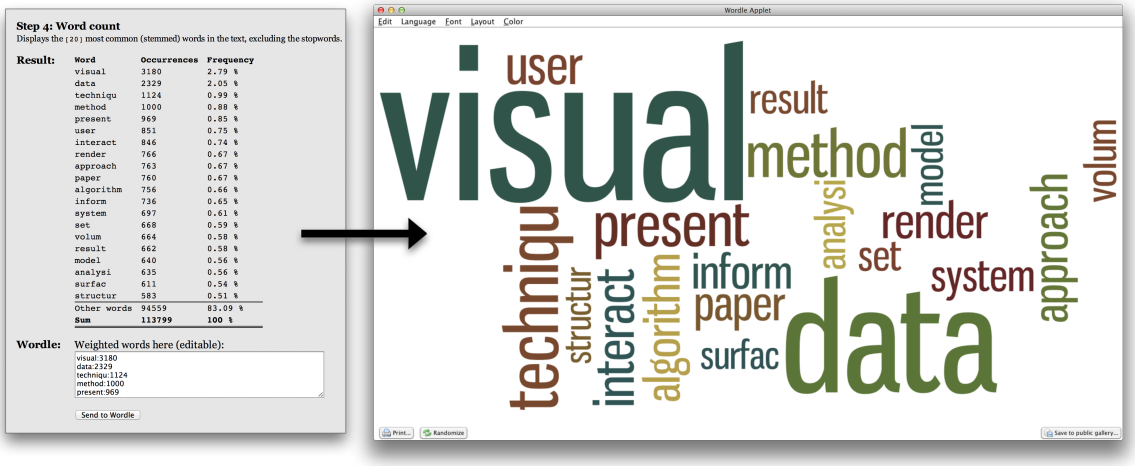


Abbildung 2.4.: Peter Holme's Word Stemmer, durch Wordle visualisiert [Hol11, Fei11]

Dadurch werden bestimmte Wortendungen abgeschnitten, was die Zusammenfassung verschiedener Wortformen (beispielsweise Singular- und Pluralform eines Substantivs) zur Folge hat. Sprachbasierte Stoppwortentfernung ist ebenfalls möglich. Eine Word-Cloud wird nicht generiert, hierfür wird die oben erwähnte POST-Request-Methode (siehe Abschnitt 2.2.1) von Wordle verwendet, wodurch sämtliche Gestaltungsmöglichkeiten des Layouts von Wordle zur Verfügung stehen. Nach der Verarbeitung des Textes werden die häufigsten Wörter zusammen mit ihren jeweiligen Häufigkeiten an Wordle gesendet, was in Abbildung 2.4 dargestellt wird. In der Abbildung ist darüber hinaus zu sehen, dass der Porter Stemmer abgeschnittene Wörter wie beispielsweise „analysi" produziert. Im Gegenzug werden Wörter wie „visual" und „visualization" zusammengefasst. Auf diese Weise werden die Verhältnisse der Worthäufigkeiten dahingehend verändert, dass die Einzelhäufigkeiten unterschiedlicher Wortformen des selben Wortstammes zusammengezählt und infolge dessen größer dargestellt werden.

Obgleich dieser Ansatz ebenfalls eine statische Word-Cloud generiert, stellt das Zusammenfassen ähnlicher Wortformen einen interessanten Verarbeitungsschritt dar, von welchem eine Word-Cloud durchaus profitieren kann.

2.2.3. Micro-Blog Analyzer

Lohmann et al. entwickelten 2012 an der Universität Stuttgart einen interaktiven Word-Cloud-Ansatz [LBSW12]. Da die zu analysierenden Datensätze sozialen Netzwerken wie beispielsweise Twitter [twi] entstammen, trägt das entstandene Programm den Namen „Micro-Blog Analyzer". Aufgabe des Programms ist einerseits die Visualisierung kookkurrenter Tags (siehe Abschnitt 3.9), andererseits die Darstellung temporaler sowie geolokaler Abhängigkeiten der Tagvorkommen. In Abbildung 2.5 ist die Hervorhebung der zu „tennis" kookkurrenten Tags zu sehen. Diese Hervorhebung zeichnet sich einerseits durch den

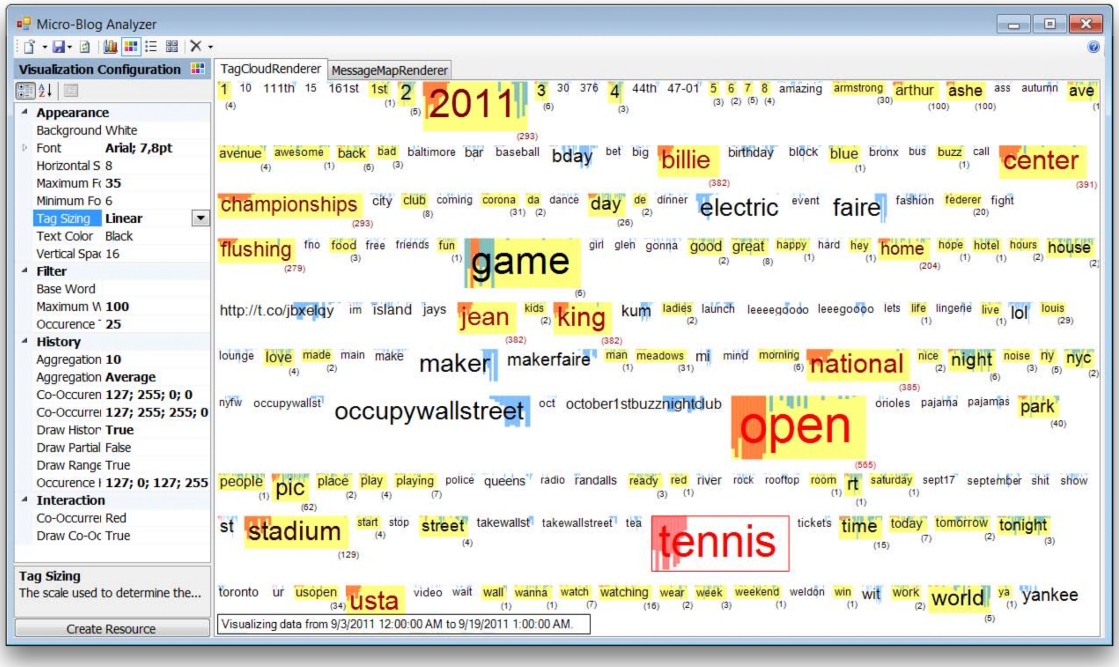


Abbildung 2.5.: Hervorhebung der zu „tennis“ kookkurrenten Tags [LBSW12]

gelben Hintergrund der kookkurrenten Tags aus, andererseits korreliert die Helligkeit der Schriftfarbe mit der Häufigkeit gemeinsamer Vorkommnisse.

Die Besonderheit dieses Ansatzes stellt das „co-occurrence highlighting“ dar, was für die visuelle Analyse von Textdokumenten mithilfe einer Word-Cloud ebenfalls von Nutzen sein kann.

2.2.4. Weitere Ansätze

Neben den bisher aufgeführten Ansätzen wurde die natürliche Sprachverarbeitung ebenfalls in anderen Arbeiten genutzt. Sowohl Stemming als auch die Entfernung der Stoppwörter kommt häufig bei einer Bereinigung von Textkorpora zum Einsatz [Ste06, DGWC10, CVW09]. Darüber hinaus wird beispielsweise in dem Ansatz von Collins et al [CVW09] das Stemming-verfahren erweitert, indem Paare aus Wort und Wortstamm gebildet werden (word,stem) und von dem jeweils häufigsten Paar eines Wortstammes dessen Wort verwendet wird. Durch dieses als „Reverse Stemming“ bezeichnete Verfahren wird sichergestellt, dass keine abgeschnittenen Wörter in der Word-Cloud dargestellt werden. Der Fokus dieses Ansatzes liegt auf dem Vergleich mehrerer Texte oder dem Vergleich mehrerer Versionen eines Textes, was sich von dem Fokus der vorliegenden Arbeit deutlich abhebt. Das Ergebnis des durch eine Word-Cloud visualisierten Textvergleichs ist in Abbildung 2.6 zu sehen.

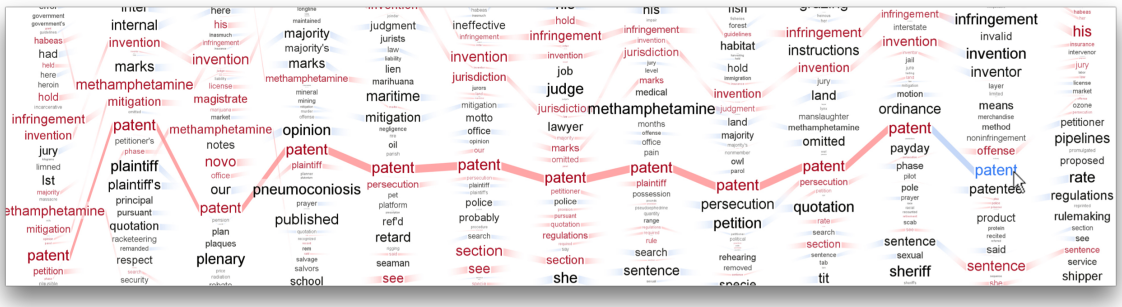


Abbildung 2.6.: Parallel Tag Clouds: Die Verbindungen eines Wortes zwischen den Texten werden visualisiert [CVW09]

In einigen Arbeiten liegt der Fokus auf der Optimierung des Layouts durch räumliche Arrangierung der Tags mit dem Ziel, den Leerraum möglichst gering zu halten oder die Tags nach bestimmten Kriterien zu sortieren oder zu gruppieren [KLKS10, Dav, CWL+10, PTT+12, RGMM07, KL07]. Abbildung 2.7 stellt eine Reduzierung des Leerraums innerhalb einer Word-Cloud dar, indem der Platz zwischen großen Wörtern für kleinere Wörter genutzt wird.

Diese Optimierungen verfolgen entweder die Absicht, die Ästhetik zu erhöhen [tag, Leu] (siehe Abbildung 2.8), was im Rahmen dieser Arbeit von geringer Priorität ist, oder die Informationsanalyse zu verbessern. Zu Letzterem wurden vergleichende Studien durchgeführt, um die Effizienz der Layouts für festgelegte Aufgaben herauszufinden [LZT09, PTT+12, SLT09]. Erwartungsgemäß ergaben die Studien, dass nicht ein bestes Layout existiert, sondern die Effizienz des Layouts von der Aufgabenstellung abhängig ist. Infolge dessen werden Word-Clouds ebenfalls mit aufgabenspezifischen Methoden wie Suchmechanismen oder Tabellen verglichen, worin die Word-Clouds jeweils das Nachsehen haben [SCH08, KHGW07, OC10, HK07].

Oosterman und Cockburn [OC10] stellten Word-Clouds herkömmlichen Tabellen gegenüber und untersuchten deren Effektivität hinsichtlich Such- und Wiedererkennungsaufgaben. Die Studie ergab deutliche Vorteile der Tabellen gegenüber den Word-Clouds, sowohl bezogen

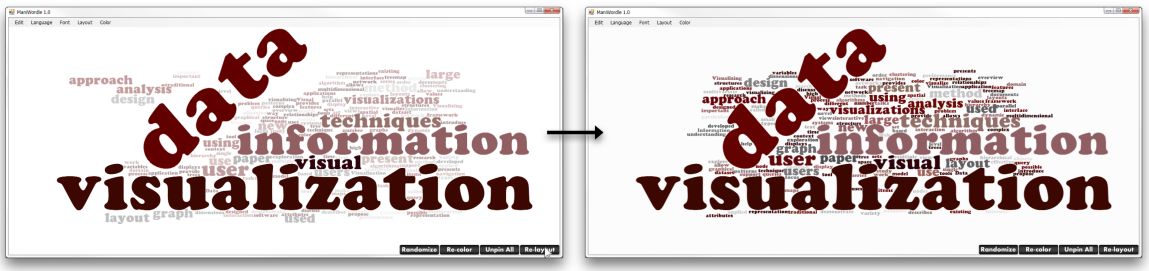


Abbildung 2.7.: Reduzierung des Leerraums innerhalb einer Word-Cloud [KLKS10]

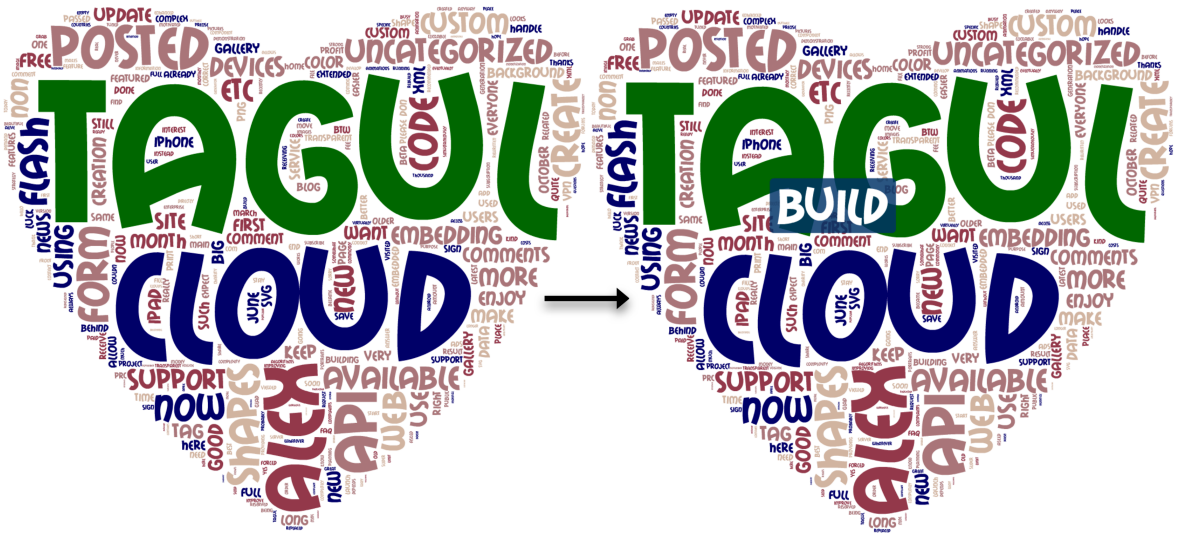


Abbildung 2.8.: Auf Ästhetik fokussierte Word-Cloud mit einer Interaktionsmöglichkeit: wird der Mauszeiger auf ein Wort (hier: „BUILD“) gefahren, so wird dieses horizontal ausgerichtet und hervorgehoben [tag]

auf die benötigte Zeit der Aufgabenbearbeitung als auch auf die Genauigkeit der Ergebnisse, was teilweise auf eine fehlende Interaktionsmöglichkeiten der Word-Clouds zurückgeführt wird.

Sinclair und Cardew-Hall [SCH08] gingen der Frage nach, wann der Einsatz von Word-Clouds sinnvoll ist, und verglichen dafür Word-Clouds mit einer datenbankbasierten Suche. Für die Beantwortung der Suchaufgaben wurde zumeist die herkömmliche Suche favorisiert. Darüber hinaus stellten Sinclair und Cardew-Hall fest, dass etwa die Hälfte der in der Datenbank gespeicherten Artikel nicht über die Word-Cloud zugänglich waren, was besonders für eine gezielte Suche erhebliche Einschränkungen darstellt. Durch eine Benutzerbefragung im Anschluss an die Studie stellte sich heraus, dass einige Teilnehmer dennoch in der Word-Cloud einen großen Vorteil sehen, andere eher in der herkömmlichen Suchfunktion, was den Gedanken nahelegt, dass eine Kombination einer herkömmlichen Suchfunktion mit einer Word-Cloud beide Interessensgruppen zufriedenstellen könnte.

Obwohl Word-Clouds verglichen mit aufgabenspezifischen Methoden offensichtlich eingeschränkt sind, wird häufig ein großer Vorteil darin gesehen, dass eine Word-Cloud einen Überblick über die zugrundeliegenden Daten bieten kann [SCH08], was in einer Studie schwer messbar ist. Einen weiteren, schwer messbaren Vorteil von Word-Clouds bezeichnet Mathes [Mat04] als „serendipity“, was für ein zufälliges Auffinden von etwas Gutem oder Nützlichem ohne die explizite Suche danach steht [ser12].

Gambette und Véronis [GV10] machen sich eine erweiterte Variante der Word-Cloud zunutze, um einen Überblick über einen Text darzustellen. Neben der durch die Häufigkeit der Wortvorkommen definierten Schriftgröße werden in dieser als „Tree Cloud“ bezeichneten Visualisierung semantische Beziehungen dargestellt. Diese Beziehungen basieren auf dem

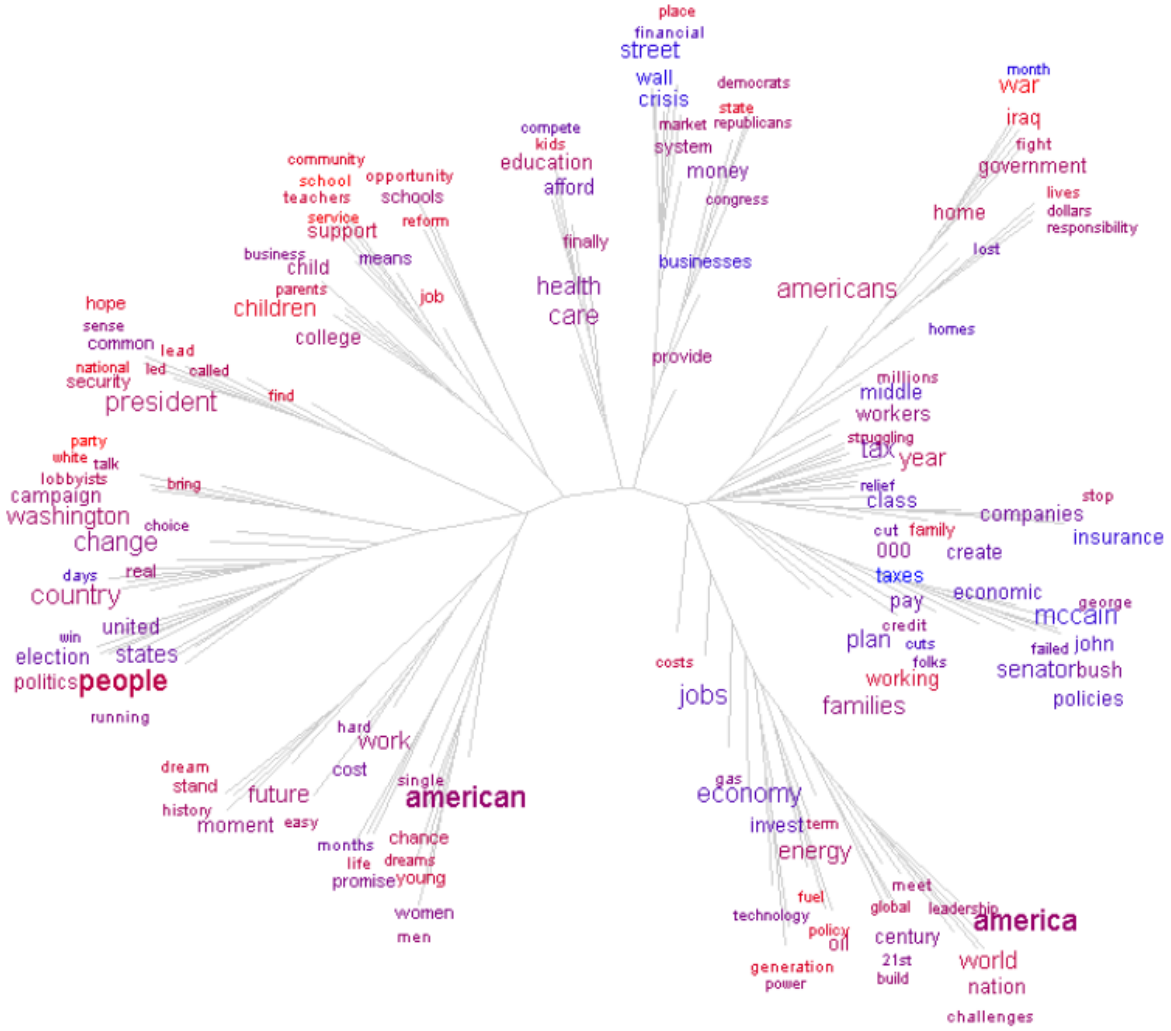


Abbildung 2.9.: Tree Cloud mit 50 Wörtern aus 138 Präsidentschaftsreden von Barack Obama

Prinzip der in Abschnitt 2.2.3 erwähnten Kookkurrenzen (siehe auch Abschnitt 2.3.5). Jedoch wird für die Berechnung der auf Kookkurrenzen aufbauenden Distanzmatrix nicht ein Satz (oder Abschnitt) als Rahmen gewählt, sondern ein sogenanntes „sliding window“. Dieses Fenster hat standardmäßig eine Größe von 30 Wörtern und wird mit der Schrittweite von einem Wort durch den zu analysierenden Text geschoben. Alle 30 Wörter innerhalb dieses Fensters sind zueinander kookkurrent. Darüber hinaus wird den Wörtern in der Tree Cloud eine Farbe zugeordnet, die mit der Chronologie korreliert. In Abbildung 2.9 ist eine beispielhafte Tree Cloud zu sehen.

Dieser Ansatz produziert wie viele der bisher vorgestellten Ansätze eine statische Word-Cloud. Die Gruppierung thematisch verwandter Wörter stellt einen interessanten Aspekt dar und kann die Grundlage für ein „clustered layout“ bilden. Jedoch wird mit einem Blick auf die in Abbildung 2.9 dargestellte Tree Cloud ersichtlich, dass keine Zusammenfassung

unterschiedlicher Wortformen (siehe Abschnitt 2.3.4) zum Einsatz kommt, weshalb Wörter wie „america“, „american“, „americans“ getrennt voneinander in der Tree Cloud dargestellt sind.

2.2.5. Diskussion

In den vorgestellten Arbeiten wurde deutlich, dass bereits einige Ansätze existieren, die auf der Idee der Word-Cloud-Visualisierungstechnik aufbauen. Verschiedenste Ansätze versuchen, sich die Vorteile einer Word-Cloud zunutze zu machen. Diese bestehen hauptsächlich darin, dass viele Benutzer gerne mit Word-Clouds arbeiten und dabei Abstriche hinsichtlich der Präzision und Geschwindigkeit (bezogen auf die Aufgabenbearbeitung und verglichen mit anderen Methoden) in Kauf nehmen.

Obwohl die jeweiligen Fokussierungen und Aufgabengebiete der vorgestellten Ansätze teils deutlich von der Aufgabenstellung dieser Arbeit abweichen oder lediglich einen Teil der gewünschten Funktionalitäten bieten, können sie hilfreiche Ideen für die Konzipierung dieser Arbeit liefern.

Die größte Einschränkung vieler der vorgestellten Arbeiten stellen die fehlenden Interaktionsmöglichkeiten dar. Da der Platz einer Word-Cloud auf den des darstellenden Mediums begrenzt ist, kann ein entsprechend umfangreiches Textkorpus niemals mit einer einzigen statischen Word-Cloud vollständig repräsentiert werden. Auch die Tatsache, dass Textkorpora aus unterschiedlichen Fachrichtungen nicht auf dieselbe Art und Weise analysiert werden können, da der Fokus jeweils ein anderer ist, schmälert die Erfolgsaussichten einer einzigen Darstellung erheblich. Der Schlüssel zu diesem Problem liegt folglich in Interaktionsmöglichkeiten und der Kombination mehrerer Ansätze.

Da das Layout der Word-Cloud nicht nur eine Geschmacksfrage ist, sondern auch von der Aufgabenstellung abhängt, bietet es sich an, mehrere Layouts zur Verfügung zu stellen. Auf diese Weise hat der Anwender die Möglichkeit, das Layout dynamisch der Aufgabe anzupassen.

Bei dem direkten Vergleich einer Word-Cloud mit Suchmechanismen, wie sie beispielsweise in Datenbanksystemen zum Einsatz kommen, wird der Nachteil der Word-Cloud deutlich. Jedoch kann dieser Nachteil leicht durch eine integrierte Suchfunktion kompensiert werden.

Hinsichtlich der Informationsanzeige sollte die Word-Cloud nicht ausschließlich auf die traditionelle Visualisierungsmethode beschränkt, sondern um eine separate Informationsanzeige erweitert sein, um neben dem groben Überblick auch exakte Detailinformationen bereitzustellen, wie es beispielsweise in Wordle angeboten wird.

In einigen der oben erwähnten Arbeiten werden interessante und nützliche Funktionen verwendet, welche auch für eine visuelle Analyse von Textdokumenten einen Mehrwert bieten können und infolge dessen integriert werden sollen.

Viele Word-Cloud-Visualisierungen bedienen sich einiger Techniken aus dem Bereich der natürlichen Sprachverarbeitung, meist um Textkorpora zu bereinigen. Jedoch hat die natürliche Sprachverarbeitung weitere Techniken zu bieten, welche für die visuelle Analyse von Textdokumenten von großem Nutzen sein können. Um diese Techniken sinnvoll in das Konzept einarbeiten zu können, ist ein Einblick in die natürliche Sprachverarbeitung, verbunden mit Erläuterungen der für diese Arbeit relevanten Techniken, hilfreich.

2.3. Natural Language Processing (NLP)

Aufgabe der natürlichen Sprachverarbeitung ist die Analyse und Repräsentation menschlicher Sprache durch einen Computer [Lido1]. Wegen des Ziels, menschliche Sprache zu verstehen und dadurch Programmiersprachen zukünftig weitestgehend überflüssig zu machen, fällt NLP in den Bereich der künstlichen Intelligenz [Rou11a]. Um die Ambiguitäten der menschlichen Sprache (beispielsweise kann mit dem Wort „Apple“ die Frucht oder auch der Konzern gemeint sein) korrekt auflösen zu können, kommt maschinelles Lernen zum Einsatz. Unter maschinellem Lernen wird die Fähigkeit eines Computers zum Wissenserwerb verstanden, ohne die explizite Programmierung dieses Wissens [Rou11b, NS07]. Als Beispiel soll ein E-Mail-Programm dienen, das unerwünschte E-Mails automatisch erkennt. Der Benutzer markiert die unerwünschten E-Mails, woraufhin das Programm deren Inhalte analysiert und seine Filterregeln daran anpasst [Die98]. Auf diese Weise kann das E-Mail-Programm zukünftige E-Mails selbstständig als unerwünscht einordnen, sofern sie einen ähnlichen Inhalt aufweisen. Um das E-Mail-Programm automatisch zu trainieren, besteht die Möglichkeit, dass die Hersteller des Programms beispielhafte E-Mails bereitstellen, anhand derer das Programm unterscheiden lernt, welche E-Mails unerwünscht sind.

2.3.1. Stoppwörter

Unter Stoppwörtern werden Wörter verstanden, die für die Analyse eines Textes keine relevanten Informationen enthalten und infolge dessen nicht in einer Word-Cloud dargestellt werden sollen [Fei10, S. 48 f.]. Beispielhafte Stoppwörter sind „der“, „die“, „das“ sowie alle unbestimmten Artikelformen. Da Stoppwörter von der Sprache des Textes abhängen, muss für jede Sprache eine eigene Liste von Stoppwörtern geführt werden, weil Stoppwörter der Sprache A relevante Wörter der Sprache B sein können. Der deutsche Artikel „die“ darf beispielsweise in englischen Texten nicht ignoriert werden.

2.3.2. Erkennung der Wortarten und Eigennamen

Ein wichtiges Gebiet der natürlichen Sprachverarbeitung stellt die Erkennung von Wortarten und Eigennamen dar. Bei der Eigennamenerkennung (named entity recognition) handelt es sich um die Kategorisierung von Wörtern. Beispielhafte Kategorien sind etwa Personen oder Organisationen.

Die in dieser Arbeit angewandten Methoden der Wortart- und Eigennamenerkennung sind sich ähnlich und basieren auf maschinellem Lernen (siehe Abschnitt 2.3). Dazu wird ein getaggtes, sprachabhängiges Textkorpus benötigt. Anhand dieses annotierten Textkorpus, in welchem jedem Wort seine Wortart sowie Kategorie zugewiesen ist, und mithilfe von Wahrscheinlichkeitsmodellen wie beispielsweise Entscheidungsbäumen oder Hidden Markov Modellen [Sch94, Sch95, TKMS03, Fin07] werden den Wörtern jeweils die bestpassenden Wortarten und Kategorien zugewiesen. Das Textkorpus ist nötig, um Ambiguitäten mithilfe des Kontextes besser auflösen zu können. Beispielsweise kann mit „Essen“ je nach Kontext eine Mahlzeit oder die Stadt gemeint sein. [FGM05, Bau07, NS07, MPR00]

2.3.3. Multiwörter

Als Multiwörtern werden im Rahmen dieser Arbeit zusammenhängende, durch ein Trennzeichen getrennte Wörter wie beispielsweise „New York“ bezeichnet. In einer Word-Cloud ist eine getrennte Darstellung der Wörter „New“ und „York“ wenig hilfreich für die visuelle Analyse eines Textes. Ein sehr simples Verfahren, um Multiwörter zu erkennen, vergleicht im Text vorkommende Wortkombinationen. Werden jeweils zwei aufeinanderfolgende Wörter verglichen, so wird diese Wortkombination als $2-$Gramm bezeichnet, allgemein wird eine solche Wortkombination abhängig von der Anzahl (n) ihrer Komponenten als $n-$Gramm bezeichnet [AA11]. „New York“ würde folglich als $2-$Gramm erkannt werden, während „New York City Bank“ unter den $4-$Grammen zu finden wäre. Nachteil dieser $n-$Gramme ist neben der Übergenerierung die nicht berücksichtigte Semantik. So werden beispielsweise auch Kombinationen aus Stoppwörtern (siehe Abschnitt 2.3.1) und ähnliche irrelevante Wortkombinationen als Multiwort erkannt. Um nur relevante Multiwörter in der Word-Cloud darzustellen, ist es unabdingbar, diese Multiwörter einer semantischen Kontrolle zu unterziehen. Beispielsweise haben relevante Multiwörter stets die Wortart Substantiv und optimalerweise werden sie als Eigenname kategorisiert (siehe Abschnitt 2.3.2). [SBB+02]

2.3.4. Zusammenfassung unterschiedlicher Wortformen

Mit unterschiedlichen Wortformen sind Modifizierungen einer Wortgrundform, wie etwa die Pluralbildung, gemeint [MS99, S. 83]. Um unterschiedliche Wortformen zusammenzufassen, existieren zwei populäre Ansätze. Zum einen ein meist regelbasiertes Verfahren, das als „stemming“ bezeichnet wird [Por01], zum anderen ein wörterbuchbasierter Ansatz, der „Lemmatisierung“ genannt wird [MS99, S. 132]. Das sogenannte Lemma stellt eine bestimmte Grundform des Wortes dar. Ein bekanntes Beispiel für einen Stemmingalgorithmus stellt der „Porter Stemmer“ dar [Por97]. Hierbei werden Wörter durch vordefinierte Regeln reduziert, indem die Wortendungen abgeschnitten werden. Abhängig von der Komplexität der Wortbildung der jeweiligen Sprache kann dieses Verfahren gute Ergebnisse erzielen, für beispielsweise die deutsche Sprache ist solch ein regelbasierter Ansatz jedoch ungeeignet. Wie der Name bereits andeutet, kommt für wörterbuchbasierte Verfahren ein Wörterbuch zum Einsatz, das alle Wortformen mit der jeweiligen Grundform beinhaltet. Auf diese Weise können auch komplexere Wortbildungen und Ausnahmen abgebildet werden [Lew05,

S. 106 ff.]. Einen Nachteil dieses Verfahrens stellt das Wörterbuch dar, das abhängig von der Sprache sehr groß werden kann, was sich einerseits in der Zeit für das Auffinden einer Grundform niederschlägt, andererseits wird mehr Speicher benötigt.

Die Zusammenfassung unterschiedlicher Wortformen bietet den Vorteil, dass ähnliche Wörter zusammengefasst werden, was jedoch den Nachteil mit sich bringt, dass potentiell auch Wortformen zusammengefasst werden, die nicht zusammengefasst werden sollten.

2.3.5. Kookkurrenz

Bei „co-occurrence“, zu deutsch Kookkurrenz, handelt es sich um gemeinsames Auftreten, welches im Rahmen dieser Arbeit stets auf Wörter bezogen ist. Zwei Wörter treten dann gemeinsam auf, wenn beide Wörter innerhalb eines gewählten Rahmens mindestens einmal vorkommen. Als Rahmen sind sowohl Sätze als auch Abschnitte denkbar. Als Beispiel für gemeinsam auftretende Wörter sollen nachfolgende zwei Sätze dienen:

Heute regnet es. Morgen soll es nicht regnen.

In diesem Beispiel tritt das lemmatisierte Wort „regnen“ mit jedem anderen Wort der zwei Sätze gemeinsam auf. Das Wort „heute“ hingegen tritt nur gemeinsam mit den Wörtern „regnen“ und „es“ auf. Wird anstelle der Satzebene die Absatzebene als Rahmen betrachtet, so ist in diesem Beispiel jedes Wort zu jedem anderen kookkurrent. Darüber hinaus kann für die Kookkurrenz die Distanz der kookkurrenten Wörter innerhalb des gewählten Rahmens zueinander in Betracht gezogen werden. [WW05, MS99, S. 554 ff.]

2.3.6. NLP-Framework

Um die eigene Implementierung von Techniken aus der maschinellen Sprachverarbeitung auf ein Minimum zu reduzieren, soll ein existierendes NLP-Framework eingebunden werden. Da im Internet zahlreiche NLP-Frameworks zur Verfügung stehen, die jeweils Vor- und Nachteile mit sich bringen, sei an dieser Stelle lediglich auf eine Übersichtsseite verwiesen [Kol12] , auf welcher einige NLP-Frameworks und -Tools aufgelistet werden. Aufgrund bisheriger Erfahrungen wurde das ausgereifte CoreNLP-Framework der Universität Stanford [cor] für die vorliegende Arbeit vorgegeben. Weitere Gründe für diese Vorgabe stellen die Erweiterbarkeit des Frameworks sowie die Tatsache, dass das Framework in Java geschrieben wurde und daher optimale Anbindungen verspricht, dar. Darüber hinaus vereint das CoreNLP-Framework diverse NLP-Werkzeuge, von welchen an dieser Stelle lediglich die verwendeten vorgestellt werden. Mit etwa 259 MB ist dieses Framework vergleichsweise schwergewichtig, was der Bandbreite an Funktionalitäten geschuldet ist.

Die Aufgabe des CoreNLP-Frameworks besteht darin, reinen Text zu annotieren. Für die Weiterverwendung sind folgende in Abbildung 2.10 farblich hervorgehobene Annotationen von Bedeutung:

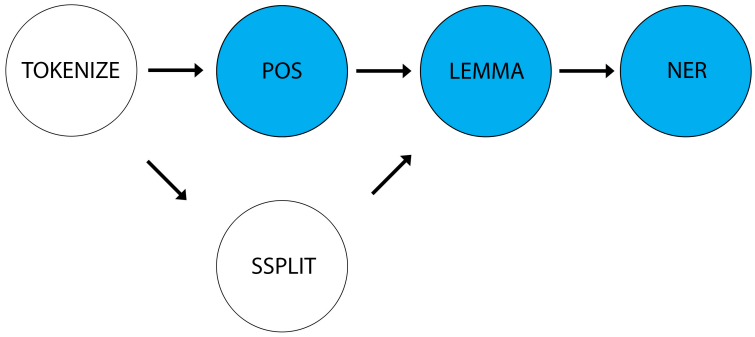


Abbildung 2.10.: Annotationsabhängigkeiten des Stanford CoreNLP-Frameworks

- Part-Of-Speech (POS) (siehe Abschnitt 2.3.2)
- Lemmatization (LEMMA) (siehe Abschnitt 2.3.4)
- Named Entity Recognition (NER) (siehe Abschnitt 2.3.2)

Wie bereits diskutiert, bietet die Lemmatisierung gewisse Vorteile gegenüber Stemming und wird standardmäßig mit dem Framework bereitgestellt, ohne zusätzliche Datenbanken oder Pakete zu benötigen. Wortarten („Part-Of-Speech“) sind zum einen hilfreich, um Multiwörter zu bestimmen, zum anderen können sie bei der Textanalyse einen Mehrwert bieten. Auch die Erkennung der Kategorien („Named Entitiy Recognition“) ist wichtig, um den Text besser analysieren zu können. Die einzelnen Annotationen weisen bestimmte Abhängigkeiten auf, die in Abbildung 2.10 dargestellt sind und im Folgenden erläutert werden. Jede Annotation baut auf dem Tokenizer („TOKENIZE“) auf. Bei diesem Vorgang wird der gesamte Text in logisch zusammenhängende Einheiten („token“) wie beispielsweise Wörter oder Satzzeichen zerlegt [AA11, MS99, S. 124 f.]. Anschließend kann der Prozess „SSPLIT“ gestartet werden, bei dem mehrere „tokens“ zu Sätzen zusammengefasst werden [MS99, S. 134 ff.]. Der darauf folgende Prozess „POS“ ist dafür zuständig, jedem Wort seine Wortart zuzuordnen. Anschließend kann die Lemmatisierung erfolgen. Darauf aufbauend folgt als letzter Schritt die Erkennung der Kategorien.

Das CoreNLP-Framework stellt somit die wichtigsten Werkzeuge zur Verfügung, lässt jedoch eine Unterstützung für einige der vorgestellten Techniken der natürlichen Sprachverarbeitung vermissen.

3. Konzept

Im diesem Kapitel wird das Konzept der vorliegenden Arbeit erarbeitet und erläutert. Grundlage für das Konzept stellen die Erkenntnisse aus den verwandten Arbeiten und die bisher existierenden Ansätze dar. Darüber hinaus werden in das Konzept interessante und potentiell nützliche Techniken der natürlichen Sprachverarbeitung integriert.

3.1. Layout

Wie aus dem vorherigen Kapitel hervorgeht, ist das Layout für die Word-Cloud von wesentlicher Bedeutung. Im Rahmen dieser Arbeit liegt der Fokus der Layoutgestaltung jedoch primär auf der Funktionalität und weniger auf der optischen Perfektion. Da Funktionalität und Effektivität des Layouts stark von dem jeweiligen Aufgabenbereich abhängig sind, sollen mehrere Layouts angeboten werden, um maximale Flexibilität zu gewährleisten [LZT09]. Für ein schnelles Lokalisieren eines gesuchten Wortes ist beispielsweise ein alphabetisch sortiertes Layout dienlich. Ist aufgabenbedingt nach der Häufigkeit gefragt, so eignet sich das alphabetische Layout weniger, in diesem Fall ist ein nach Häufigkeit sortiertes Layout die bessere Wahl. Dieses häufigkeitsbasierte Layout kann entweder zirkulär oder in Listenform dargestellt werden. Sind Häufigkeiten direkt miteinander zu vergleichen, so hat die listenförmige Darstellung gegenüber der zirkulären den Vorteil, dass zwischen benachbarten Wörtern stets die minimale Häufigkeitsdifferenz garantiert ist. Ein weiteres wünschenswertes Layout ist das „clustered layout“ [ASM10], bei welchem die Wörter nach gewissen Kriterien, wie etwa der Kookkurrenz (siehe Abschnitt 2.3.5), räumlich gruppiert werden, um Zusammenhänge zu visualisieren. Eine beispielhafte Darstellung eines solchen Layouts (jedoch thematisch sortiert) ist in Abbildung 2.2 zu sehen und konnte besonders bei der Suche nach thematisch verwandten Wörtern überzeugen.

3.2. Suche

Besonders im Hinblick auf die Textanalyse ist es notwenig, eine Suchfunktion zu integrieren [SCH08]. Einerseits soll die Suche das Auffinden des gesuchten Wortes in der Word-Cloud erleichtern. Andererseits sollen die Informationen zu dem gesuchten Wort angezeigt werden, unabhängig davon, ob das Wort Bestandteil der Word-Cloud ist. Auf diese Weise können die Vorteile der Word-Cloud und die Vorteile einer herkömmlichen Suchfunktion kombiniert und dadurch Synergieeffekte genutzt werden.

Werden unterschiedliche Wortformen zusammengefasst (siehe Abschnitt 2.3.4), so soll die Suche für sämtliche auftretenden Wortformen ein Ergebnis liefern, auch wenn die gesuchte Wortform nicht der häufigsten und somit repräsentativen entspricht.

3.3. Informationsanzeige

Einen wichtigen Punkt des Programms stellt die Informationsanzeige dar. In Anlehnung an beispielsweise die Häufigkeitsanzeige des in Abschnitt 2.2.1 vorgestellten Wordle sollen neben der Visualisierung durch die Word-Cloud Informationen zu Wörtern und Zusammenhängen direkt einsehbar sein. Um eine Verhältnismäßigkeit der Häufigkeiten herstellen zu können, soll stets die Gesamtzahl der Wörter sowie die minimale und maximale Häufigkeit angezeigt werden. Über die Häufigkeit hinaus sollen jegliche Eigenschaften der Wörter, wie beispielsweise Kategorie oder Wortart, und ebenso die Zusammenhänge zwischen Wörtern, etwa die Anzahl gemeinsam geteilter Sätze, dargestellt werden. Bei der Visualisierung der Informationen ist darauf zu achten, welcher Teil der Informationen immer sichtbar ist und welcher Teil nur bei Bedarf einsehbar sein soll.

3.4. Entfernung der Stoppwörter

Analog zu den vorgestellten Arbeiten soll auch in der vorliegenden Arbeit eine Entfernung der Stoppwörter möglich sein. Da sich diese Arbeit auf englische Texte beschränkt, werden keine sprachabhängigen Stoppwortlisten benötigt. Es genügt folglich eine einzige Liste der englischen Stoppwörter. Jedoch soll die Stoppwortliste nicht in das Programm integriert, sondern potentiell austausch- und erweiterbar konzipiert werden. Im Anschluss an die Verarbeitung des Textes soll für jedes Wort des Textes überprüft werden, ob es in der Stoppwortliste enthalten ist, und gegebenenfalls ignoriert werden.

Da eine vorgegebene Liste nicht für jeden Anwendungsfall das optimale Ergebnis erzielen kann, muss die Liste dynamisch anpassbar sein. Darüber hinaus soll die Möglichkeit bestehen, die Entfernung der Stoppwörter zu deaktivieren, ohne die Liste zu leeren.

3.5. Erkennung der Wortarten

Da in einigen Anwendungsfällen spezielle Wortarten von Interesse sind (beispielsweise können bei der Auswertungen von Interviews speziell Adjektive interessant sein) soll das Programm eine Zuordnung der Wörter zu ihren Wortarten unterstützen. Darüber hinaus sollen die einzelnen Wortarten sowohl separat als auch kombiniert gefiltert werden können, um Wortarten gezielt ein- oder ausblenden zu können. Diese Funktionalität wurde bisher in keiner existierenden Arbeit in dieser Form verwendet, verspricht jedoch einen potentiellen Mehrwert für die Textanalyse.

3.6. Erkennung von Kategorien

Analog zu den Wortarten sollen die Kategorien ebenso einzeln sowie kombiniert filterbar sein, um die Anzeige einschränken zu können. Darüber hinaus sollen die Kategorien, welche das CoreNLP-Framework verwendet, durch eine Zuordnungstabelle auf frei wählbare Kategorien abgebildet werden. Auf diese Weise kann die Benutzeroberfläche flexibel gestaltet werden und ist nicht an die Vorgaben des Frameworks gebunden.

3.7. Erkennung von Multiwörtern

In der Word-Cloud sollen die Wörter „New“ und „York“ nicht getrennt voneinander angezeigt werden. Dies unterstützen einige der vorgestellten Arbeiten, indem Multiwörter durch eine Tilde getrennt (New∼York) manuell eingegeben werden können. Da die Wörter im Rahmen dieser Arbeit jedoch automatisch aus dem Textkorpus extrahiert werden, sollen die Multiwörter ebenfalls ohne Benutzerinteraktion erkannt werden. Kommen die Komponenten, beispielsweise „new“, in einem anderem Kontext innerhalb des Textkorpus vor, so soll sowohl diese Komponente als eigenständiges Wort als auch das Multiwort in der Word-Cloud auftauchen. Da die Darstellung von Multiwörtern in der Word-Cloud jedoch auch störend für die Analyse sein kann, soll diese Funktion deaktivierbar konzipiert werden.

3.8. Zusammenfassung unterschiedlicher Wortformen

Besonders im Hinblick auf die Darstellungsform „Word-Cloud“ ist eine Zusammenfassung unterschiedlicher Wortformen sinnvoll (siehe Abschnitt 2.3.4). Der Anwender will in der Word-Cloud beispielsweise das Wort „algorithm“ sehen, jedoch nicht zusätzlich das Wort „algorithms“. Ähnlich verhält es sich bei unterschiedlich konjugierten Verbformen und der Veränderung anderer Wortarten. Um keine abgeschnittenen Wörter darzustellen, soll (ähnlich dem in Abschnitt 2.2.4 erläuterten Verfahren des „Reverse Stemming“) die häufigste Wortform als Repräsentant der zusammengefassten Wortformen in der Word-Cloud dargestellt werden.

Da die Zusammenfassung unterschiedlicher Wortformen unweigerlich auch unerwünschte Ambiguitäten mit sich bringt und somit Einschränkungen für spezielle Aufgaben darstellt, sollte diese Funktion deaktivierbar konzipiert werden.

3.9. Kookkurrenz

Wie in Abschnitt 2.3.5 erläutert, sind als Rahmen der Kookkurrenz sowohl Sätze als auch Abschnitte denkbar. Infolge dessen soll dieser Rahmen konfigurierbar konzipiert werden. Die Metrik der Distanz zweier Wörter innerhalb eines Rahmens zueinander stellt einen

interessanten Aspekt dar. Jedoch basiert die Word-Cloud auf Häufigkeiten, weshalb dieser Distanzfaktor für die visuelle Analyse eher hinderlich ist, da die dargestellte Schriftgröße des Wortes in diesem Fall einen Rückschluss auf dessen Häufigkeit verhindert.

Das Programm soll die Kookkurrenzen sowohl zu einem einzelnen Wort wie auch zu mehreren Wörtern gleichzeitig darstellen können. Um dem Anwender zu verdeutlichen, ob in der Word-Cloud Kookkurrenzen dargestellt werden, müssen die für die Kookkurrenz ausgewählten Wörter stets in der Benutzeroberfläche sichtbar sein.

3.10. Eingabe

Eine notwendige Funktion für die Word-Cloud stellt die Eingabe des Textkorpus dar, für die es verschiedene Möglichkeiten geben soll. Eine Möglichkeit Text einzugeben soll darin bestehen, direkt in ein Textfeld zu schreiben, oder Text aus der Zwischenablage darin einzufügen. Des Weiteren soll ein Dialog zur Verfügung stehen, um Textdateien im Dateisystem auswählen zu können, ohne deren Inhalt in die Zwischenablage zu kopieren. Wurde ein Textdokument auf diese Weise ausgewählt, so soll die Word-Cloud nicht umgehend generiert werden, damit die Möglichkeit besteht, den geladenen Text vorher anzupassen. Nachdem ein zu verarbeitender Text gewählt wurde, ist eine Schätzung der bevorstehenden Verarbeitungsdauer hilfreich, auch wenn diese aufgrund der vielen dem Programm unbekannten Faktoren wie beispielsweise dem Prozessortakt und der Komplexität des Textes, lediglich eine grobe Richtlinie darstellt.

3.11. Persistenz

Da die Verarbeitung des Textkorpus ein sehr zeit- und rechenaufwändiger Prozess ist, soll die Möglichkeit bestehen, die berechneten Ergebnisse zu speichern. Optionen und Einstellungen bezüglich der Benutzeroberfläche sollen ebenfalls persistiert werden können, um den Bedienkomfort zu erhöhen. Hierfür bietet Java diverse Möglichkeiten an, die abhängig von den zu speichernden Daten zu wählen sind. Die Stoppwortliste besteht beispielsweise aus einzelnen Wörtern, und soll optional durch einen Texteditor bearbeitbar sein (Abschnitt 3.4). In diesem Fall bietet sich eine simple Textdatei an, da alle Objekte vom selben Datentyp sind und keine Hierarchie benötigt wird. Anders verhält es sich bei der Konfigurationsdatei für die Einstellungen. Da die Einstellungen über das Programm angepasst werden, ist es nicht notwendig, menschliche Lesbarkeit zu verwenden, was eine binäre Repräsentation nahelegt. Binär gespeicherte Dateien haben den großen Vorteil, dass sie schneller gelesen und geschrieben werden können als beispielsweise Textdateien, für die jeweils sogenannte „reader“ und „writer“ benötigt werden. Darüber hinaus können in Java komplette Klassen als Binärdatei gespeichert werden, was nach dem Einlesen der Datei einen weiteren Verarbeitungsschritt einspart. Für die Ergebnisse des verarbeiteten Textkorpus ist ebenfalls eine Binärdatei sinnvoll, da diese direkt von dem Programm interpretiert werden müssen und keine Bearbeitung durch externe Editoren vorgesehen ist.

4. Eigener Ansatz

Zunächst soll ein grober Überblick über das Programm vermittelt werden, um die nachfolgend dargestellten Funktionalitäten gut einsortieren zu können. Anschließend wird ein kurzer Ablauf des Programms, angefangen bei einem Text bis hin zur Generierung der interaktiven Word-Cloud, vorgestellt. Um die Funktionalitäten vollständig erklären zu können, ist die Vorstellung der wichtigsten Datenstrukturen des Programms nötig. Anschließend werden die Funktionen und Eigenschaften des Programms anhand der Benutzeroberfläche erläutert und veranschaulicht. Eine Diskussion der aufgetretenen Implementierungsprobleme schließt das Kapitel ab.

4.1. Überblick

Der in Abbildung 4.1 dargestellte, schematische Ablauf des Programms wird im nachfolgenden Abschnitt näher erläutert, soll jedoch an dieser Stelle zum Überblick des Programms beitragen. Nachdem ein Textkorpus ausgewählt wurde und die im Konzept dargestellten Verarbeitungsschritte durchlaufen hat, kann das Hauptfenster des Programms dargestellt werden. Dieses Fenster enthält neben diversen Informationen einige Einstellungs- sowie Interaktionsmöglichkeiten. Abbildung 4.2 soll einen möglichst umfassenden Überblick über das Hauptfenster bieten, um eine gute Orientierung zu ermöglichen. Welche Funktionen sich hinter den Menüeinträgen und Oberflächenelementen verbergen, werden im Anschluss an den Ablauf im Detail erläutert.

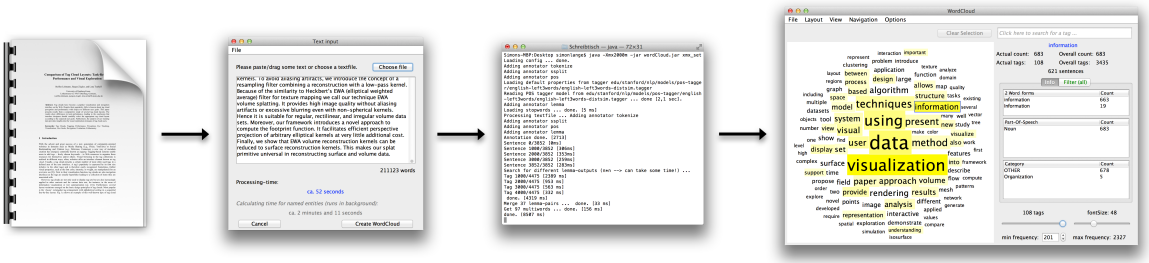


Abbildung 4.1.: Schematischer Programmablauf:
Eingabe, Verarbeitung und Darstellung eines Textkorpus

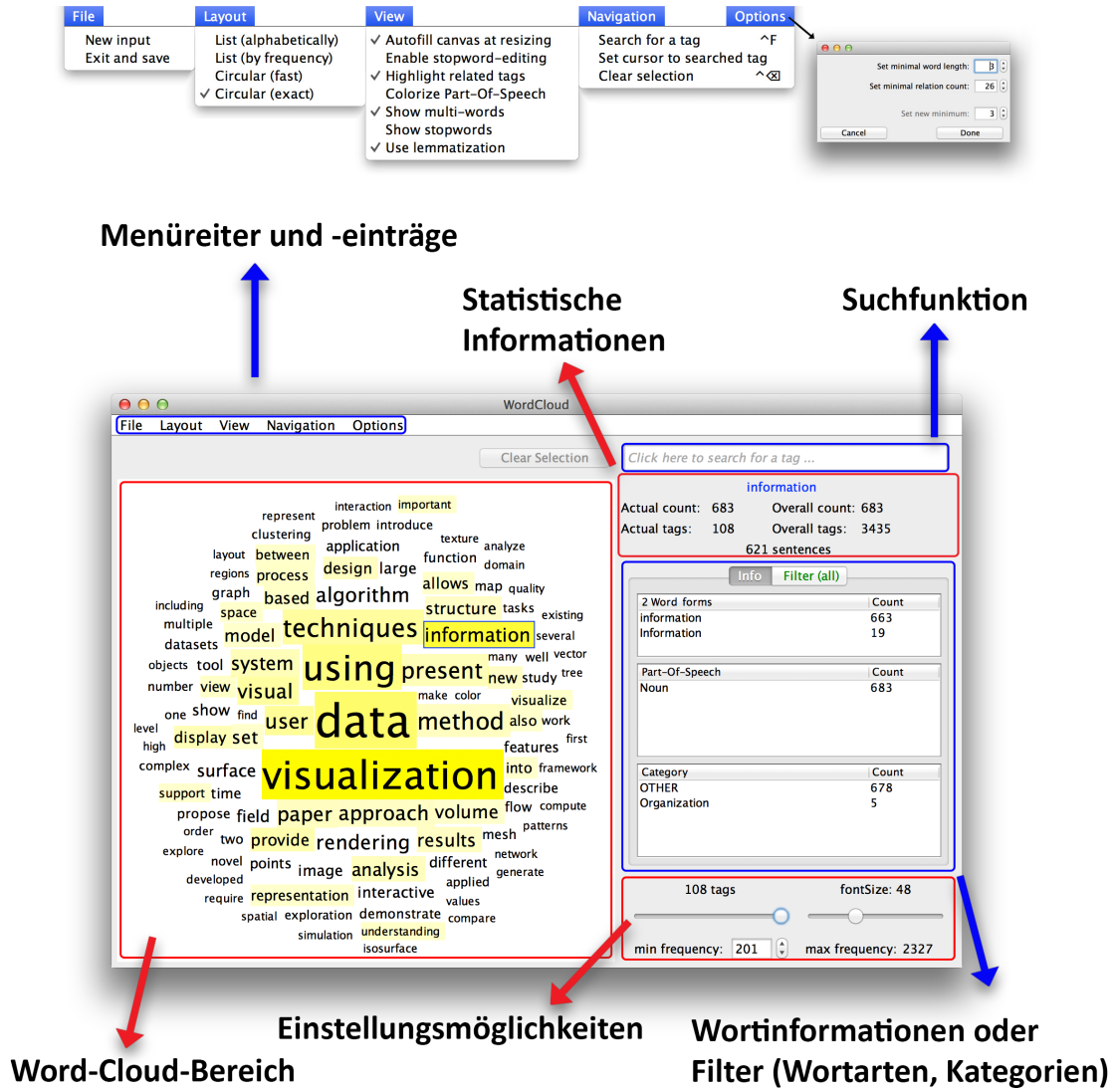


Abbildung 4.2.: Übersicht des Hauptfensters

4.2. Ablauf

In Abbildung 4.1 ist ein schematischer Ablauf des Programms dargestellt. Zunächst wird für die Verwendung des Programms ein Text benötigt. Sollen mehrere Texte als Datengrundlage dienen (beispielsweise einzelne Nachrichten), so enthält das Programm die Hilfsklasse „MergeTxtFiles“, die hierfür Unterstützung anbietet. Aufgabe dieser Hilfsklasse ist es, die relevanten Informationen aus den einzelnen Textdateien in einer einzigen Textdatei zu konkatenieren. Steht die Textdatei bereit, kann das Programm geöffnet, die Textdatei im Eingabefenster ausgewählt und die Generierung der Word-Cloud gestartet werden. Daraufhin öffnet sich das Hauptfenster des Programms, während im Hintergrund diverse Vorgänge ab-

laufen. Zunächst muss der Text verarbeitet werden. Hierfür kommt das CoreNLP-Framework zum Einsatz, welches den Text zunächst ohne Berücksichtigung der Kategorien annotiert, also mit zusätzlichen Informationen wie beispielsweise den Wortarten versieht. Anschließend wird der annotierte Text Satz für Satz und innerhalb eines Satzes Wort für Wort durchlaufen und in die entsprechenden Datenstrukturen gespeichert. Nach dieser Verarbeitung wird umgehend die Word-Cloud generiert und dargestellt. Während das Programm bereits größtenteils verwendbar ist, werden den einzelnen Wörtern im Hintergrund mithilfe des CoreNLP-Frameworks ihre Kategorien zugewiesen. Dieser Vorgang dauert meist länger als die ursprüngliche Annotierung[1] und läuft deshalb im Hintergrund ab. Sobald dieser Vorgang abgeschlossen ist, werden sowohl die zugrunde liegenden Daten als auch die Oberfläche der Word-Cloud aktualisiert, woraufhin dem Anwender nun der volle Funktionsumfang zur Verfügung steht, welcher in den nachfolgenden Abschnitten vorgestellt wird.

4.3. Datenstruktur

Um die Funktionalitäten und Eigenschaften des Programms erläutern zu können, ist ein kurzer Einblick in die wichtigsten Datenstrukturen sinnvoll. Diese werden nach dem Bottom-Up-Prinzip vorgestellt, angefangen bei den grundlegenden Klassen bis hin zu den komplexeren. Da die Klassen meist umfangreich sind, werden nur die für den Leser relevanten Eigenschaften daraus erläutert. Die vollständigen Listings der wichtigen Klassen sind im Anhang A.1 zu finden.

4.3.1. Tag

Ein grundlegendes Element dieses Programms ist die Klasse „Tag“, deren relevante Attribute in Listing A.1 auf Seite 89 zu sehen sind. In dieser Klasse werden alle wortspezifischen Informationen repräsentiert und im Folgenden näher erläutert. Grundsätzlich repräsentiert ein Tag eine Gruppe von Wörtern, die eine gemeinsame Grundform besitzen (siehe Abschnitt 2.3.4). Neben den verschiedenen Wortformen wird in einem Tag die häufigste Wortform gespeichert, da sie den Repräsentanten des Tags für die Word-Cloud darstellt. Darüber hinaus werden die Häufigkeiten der Wortformen sowie die daraus resultierende Schriftgröße gespeichert. Linguistische Informationen, beispielsweise darüber, ob es sich bei dem Wort um ein Stoppwort (oder Multiwort) handelt, werden ebenfalls repräsentiert. Für die Wortarten und Kategorien wird jeweils eine Liste mit den jeweiligen Häufigkeiten angelegt. Als letztes erwähnenswertes Attribut enthält jedes Tag eine Liste aller Sätze, in welchen mindestens eine der Wortformen vorkommt.

[1] für exakte Zeitmessungen siehe Abschnitt 4.5.6 auf Seite 66

4.3.2. Label

Die Klasse „Label“ stellt eine Erweiterung der Java Swing Komponente „JLabel“ dar. Neben den geerbten Eigenschaften wie beispielsweise „text“ wurde das Attribut „tag“ hinzugefügt, was einen Tag beinhaltet. Darüber hinaus verfügt das Label über sogenannte „MouseListener“, die dafür zuständig sind, auf folgende Mausereignisse zu reagieren:

- Der Mauszeiger wird über das Label gefahren (siehe Abschnitt 4.4.2)
- Der Mauszeiger verlässt das Label (siehe Abschnitt 4.4.2)
- Das Label wird angeklickt (siehe Abschnitt 4.4.5)

Labels stellen die Bausteine der Word-Cloud dar. Jedes darzustellende Tag wird einem Label zugeordnet und in der Word-Cloud entsprechend dem ausgewählten Layout visualisiert. Unkomplizierter wäre es, anstelle der Konstruktion aus Tags und Labels alle Tags direkt als Labels zu speichern. Bei dieser Variante stellte sich jedoch schnell heraus, dass Labels deutlich mehr Speicherplatz benötigen und die Anwendung dadurch schlechter skalieren würde. Da in der Word-Cloud immer nur ein gewisser Teil der Tags angezeigt wird, spart diese späte Zuordnung zwischen Labels und Tags einen Mehraufwand, der unweigerlich durch die nicht dargestellten, im Speicher liegenden Labels entstehen würde.

4.3.3. SortedLabels

In der Klasse „SortedLabels“ werden alle textbezogenen Daten gehalten, welche in Listing A.2 auf Seite 89 zu sehen sind. Die Klasse wird entweder unmittelbar nach der Verarbeitung eines Textkorpus mit Daten gefüllt oder aus einer binär gespeicherten Textdatei wiederhergestellt. Um einen Text binär zu speichern, wird eben diese Klasse persistiert. Um eine erneute Verarbeitung des Textkorpus zu ermöglichen, muss dieser gespeichert werden. Darüber hinaus werden alle Sätze als Liste gespeichert, um die Berechnung von Kookkurrenzen zu ermöglichen. Das wichtigste Attribut dieser Klasse stellt die Liste aller Tags dar, die entweder alphabetisch oder nach deren Häufigkeit sortiert ist.

4.3.4. Externe Konfiguration

Wird das Programm gestartet, so wird zunächst die externe Konfigurationsdatei ausgelesen, die in Listing 4.1 zu sehen ist. Existiert diese Datei nicht, wird eine vorkonfigurierte Datei neu

Listing 4.1 config.xml

```
<minimum>1</minimum>
<noNamedEntities>1</noNamedEntities>
<XMX-Parameter>2048</XMX-Parameter>
<splitNewlines>0</splitNewlines>
```

erstellt. In dieser Datei sind vier Informationen enthalten, welche das Programm wesentlich beeinflussen.

Hinter dem Begriff „minimum“ verbirgt sich eine untere Schranke, die bei den Optionen (siehe Abschnitt 4.4.11) genauer erklärt wird. „noNamedEntities“ gibt dem Anwender die Option, gänzlich auf Kategorien zu verzichten. Dies ist besonders dann hilfreich, wenn nur wenig Arbeitsspeicher zur Verfügung steht oder keine Kategorien benötigt werden. Da das CoreNLP-Framework sehr umfangreich ist (ca. 259 MB) und während der Verarbeitung von Textkorpora zusätzlichen Speicherplatz benötigt, muss der Java-Umgebung entsprechend viel Speicherplatz zur Verfügung gestellt werden. Dies erfolgt über den Parameter „xmx“, gefolgt von einer in Megabyte angegebenen Zahl. Standardmäßig eingestellt sind 2048 MB, was für Texte bis zu einer Größe von etwa 1 Million Wörter ausreicht[2]. „splitNewlines“ ermöglicht, den Rahmen der Kookkurrenzen auf Abschnitte einzustellen (siehe Abschnitt 4.4.9). Wurden diese vier Werte vom Programm ausgelesen, so startet sich das Programm erneut. Grund für den Neustart des Programms ist die Speicherangabe, die nur per Befehlszeile angegeben werden kann (VM-Argumente sind nicht Bestandteil einer „runnable jar“-Datei). Der Neustart ist für den Anwender nicht sichtbar, da das Eingabefenster beim ersten Start nicht angezeigt wird. Das Programm wird, so nicht anders konfiguriert, mit folgender Befehlszeile aufgerufen:

„java -Xmx2000m -jar wordCloud.jar xmx_set“

Der letzte Parameter „xmx_set“ gibt an, dass sich das Programm nicht erneut aufrufen soll. Wird das Programm nicht per Doppelklick, sondern direkt über die Befehlszeile gestartet, so kann eben diese Zeile dafür verwendet werden und es muss nicht der Umweg über die Konfigurationsdatei gegangen werden. Dem ambitionierten Anwender wird aufgrund der sonst fehlenden Konsolenausgabe der Weg über die Befehlszeile empfohlen.

4.3.5. Interne Konfiguration

Die Klasse der internen Konfiguration („Config“) ist für die Verwendung dieses Programms von großer Bedeutsamkeit. In dieser Klasse werden alle Konfigurationseinstellungen vorgenommen, die nicht Bestandteil der externen Konfigurationsdatei sind (siehe Listing 4.1). Um den Verlust getroffener Einstellungen bei einem Neustart des Programms zu vermeiden, wird die interne Konfiguration unmittelbar vor dem Beenden in die Datei „internal_config.data“ persistiert. Falls diese Datei nicht existiert, wird eine neue Datei mit Standardeinstellungen generiert. Unmittelbar nach dem Programmstart wird die Datei ausgelesen. In der Konfiguration werden beispielsweise Dateinamen festgelegt, Farben definiert, Schriftarten eingestellt und Texte für spezielle Situationen (zum Beispiel „No tags to display!“) hinterlegt. Darüber hinaus wird die Zuordnung zwischen den vom CoreNLP-Framework festgelegten und den in der Word-Cloud angezeigten Kategorien definiert („NP“ wird beispielsweise als „Noun“ angezeigt). Das vollständige Listing der Klasse A.3 auf Seite 90 ist im Anhang zu finden.

[2]Für genauere Abschätzung siehe Tabelle 4.1 auf Seite 66

4.4. Benutzeroberfläche

Anhand der Benutzeroberfläche werden im Folgenden die umgesetzten Funktionen beschrieben und veranschaulicht.

4.4.1. Überblick

Abbildung 4.3 soll einen Überblick über die Benutzeroberfläche geben. Da es für dieses Hauptfenster nicht möglich ist, alle Programmelemente in einer einzigen Abbildung darzustellen, werden für die Beschreibungen der Elemente jeweils Teile der WordCloud herausgegriffen oder entsprechend hervorgehoben. Da die gesamte Arbeit unter Mac OS X entstanden ist, können die Abbildungen von der tatsächlichen Darstellung auf dem jeweiligen Betriebssystem geringfügig abweichen. Wenn nicht anders angegeben, dient als Textgrundlage stets eine Konkatenation von Abstracts aus Veröffentlichungen der VisWeek (Zeitraum: 1998-2011) [vis, HKBE12, S. 6].

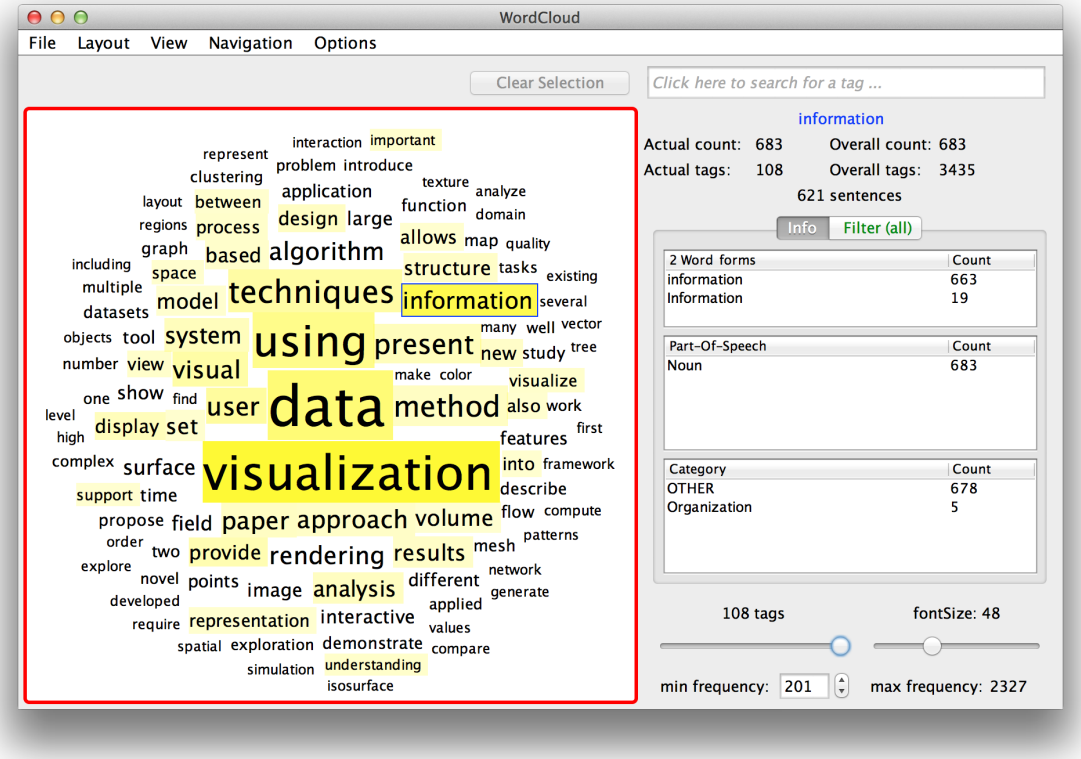


Abbildung 4.3.: Word-Cloud-Bereich

4.4.2. Word-Cloud-Bereich

In Abbildung 4.3 ist auf der linken Seite der rot umrandete Word-Cloud-Bereich zu sehen. Da in dieser Arbeit eine Word-Cloud als Visualisierungstechnik dienen soll, nimmt sie folglich die größte Fläche in Anspruch.

Word-Cloud-Generierung

Für die Erzeugung der Word-Cloud wird wie folgt vorgegangen: Zunächst müssen die darzustellenden Tags aus der Klasse „sortedLabels“ extrahiert werden. Abhängig von gewählter Reihenfolge und Anzahl der darzustellenden Tags (Filter und Auswahl werden an dieser Stelle nicht näher betrachtet) wird eine entsprechende Untermenge von Tags zurückgegeben. Anhand des häufigsten und seltensten Tags dieser Untermenge wird die Größe aller darzustellenden Tags berechnet. Hierbei wird dem seltensten Tag die minimale Schriftgröße zugewiesen, dem häufigsten Tag die maximale, welche indirekt über die Benutzeroberfläche variiert werden kann (siehe Abschnitt 4.4.2). Das Verhältnis zwischen maximaler und minimaler Schriftgröße ist in der internen Konfigurationsdatei hinterlegt und beträgt standardmäßig $0,75$ also $3:1$. Um die Schriftgröße eines darzustellenden Tags zu bestimmen, kommt folgende Formel zum Einsatz:

$$s_{\text{tag}} = \left\lfloor s_{\max} - \frac{p \cdot s_{\max} \cdot (f_{\max} - f_{\text{tag}})}{f_{\max} - f_{\min}} \right\rfloor \tag{4.1}$$

Hierbei steht s für die Schriftgröße, f für die Häufigkeit und p für das Verhältnis zwischen maximaler und minimaler Schriftgröße. Für den Fall, dass maximale und minimale Häufigkeit gleich groß sind, wird der Nenner ignoriert. Die berechnete Schriftgröße wird für jedes Tag in dessen Eigenschaft „fontsize“ gespeichert. Anschließend wird eine Platzierung der Tags für das gewählte Layout simuliert. Simulation und tatsächliche Platzierung unterscheiden sich darin, ob die Labels gezeichnet werden. Konnte allen darzustellenden Tags ein Platz zugewiesen werden, so wird aus jedem Tag ein Label generiert, platziert und in der Word-Cloud dargestellt.

Anzahl der Tags begrenzen

Eines der Steuerelemente, welches in Abbildung 4.3 in der rechten unteren Ecke zu sehen ist, regelt die Anzahl der angezeigten Tags. Da es sich bei der Anzahl der Tags um eine natürliche Zahl handelt, bei der es nicht auf eine exakte Ermittlung ankommt, kann sie mittels eines Schiebereglers eingestellt werden. Der Wertebereich des Schiebereglers beginnt bei dem Wert 1 und ist nach oben hin durch die maximale Anzahl darstellbarer Tags begrenzt, die (abhängig von dem gewählten Layout) für eine konstant große Fläche sehr unterschiedlich sein kann (siehe Abbildung 4.16 auf Seite 50). Abhängig von der Einstellung, ob die optimale Anzahl darstellbarer Tags berechnet werden soll oder nicht (siehe Abschnitt 4.4.9), stellt diese obere Grenze die korrekte Beschränkung oder nur eine Annäherung dar (dies wird farblich

Abbildung 4.4.: Anzahl der Tags begrenzen

hervorgehoben, siehe Abbildung 4.19 auf Seite 54). In Abbildung 4.4 ist eine Begrenzung auf etwa die Hälfte der darstellbaren Tags veranschaulicht.

Schriftgröße der Labels anpassen

Ein weiteres Steuerelement, welches ebenfalls in der rechten unteren Ecke in Abbildung 4.3 zu sehen ist, steuert die Schriftgröße der angezeigten Labels. Auch bei diesem Element kommt ein Schieberegler zum Einsatz, da die Schriftgröße ebenfalls als natürliche Zahl repräsentiert wird. Der eingestellte Bereich für die Schriftgröße umfasst Werte zwischen 20 und 100, was für gängige Bildschirmauflösungen ausreichend Flexibilität bietet. Hierbei ist zu beachten, dass die Schriftgröße, die per Schieberegler ausgewählt werden kann, nicht der tatsächlichen Maximalgröße entspricht, da für die Berechnung der Schriftgröße noch das Verhältnis zwischen kleinster und größter Schriftgröße einbezogen wird (siehe Gleichung 4.1). Bei dem voreingestellten Verhältnis von $0,75$ entspricht Schriftgröße 20 also den Labelschriftgrößen 5 bis 15, während Schriftgröße 100 Labelschriftgrößen im Bereich von 25 bis 75 produziert. Standardmäßig ist Schriftgröße 48 voreingestellt, was den Labelschriftgrößenbereich 12 bis 36 abdeckt und eine gute Lesbarkeit bietet. Die angezeigten Werte richten sich stets nach der Maximalgröße, unabhängig von diesem Verhältnis, um einen konstanten Wertebereich (20 bis 100) zu repräsentieren. Da die Schriftgröße von der Bildschirmauflösung und dem persönlichen Geschmack des Anwenders abhängt, kann sie über den Schieberegler interaktiv geändert werden. Ein Vergleich der Schriftgrößen mit daraus resultierenden Anzahlen an dargestellten Tags ist in Abbildung 4.5 dargestellt.

(a) Schriftgröße: 20 (1257 *Tags*) (b) Schriftgröße: 48 (263 *Tags*) (c) Schriftgröße: 100 (40 *Tags*)

Abbildung 4.5.: Schriftgrößenvergleich mit der Anzahl dargestellter Tags (Vollbildmodus)

Mindesthäufigkeit festlegen

Über das letzte direkte Steuerelement „min frequency“, welches ebenfalls in der rechten unteren Ecke in Abbildung 4.3 zu finden ist, kann die Mindesthäufigkeit festgelegt werden. Diese Funktion verhindert die Darstellung von Tags mit kleinerer Häufigkeit als der angegebenen Mindesthäufigkeit. Dem Steuerelement liegt die Java Swing Komponente „JSpinner“ zugrunde. Da der Wertebereich der auftretenden Häufigkeiten nicht stetig ist, muss das Modell der Komponente für jede Änderung der Word-Cloud angepasst werden. Die durch das Textfeld der Komponente eingegebene Zahl wird auf eine auftretende Häufigkeit mit minimaler Differenz abgebildet. Alternativ können die beiden mit je einem Pfeil versehenen

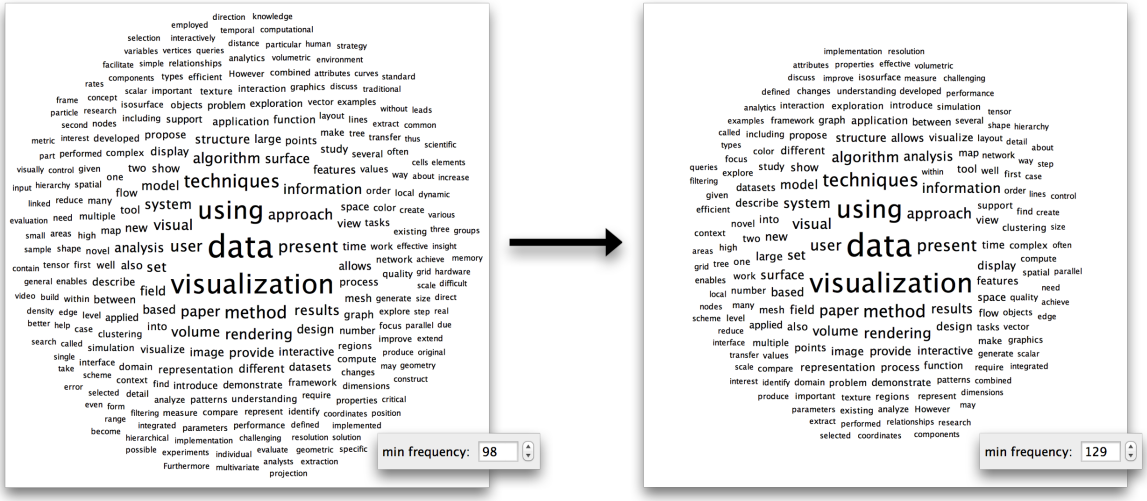


Abbildung 4.6.: Mindesthäufigkeit

Schaltflächen verwendet werden, um den Wert schrittweise zu erhöhen beziehungsweise zu verringern. In Abbildung 4.6 ist eine Erhöhung der Mindesthäufigkeit veranschaulicht.

Auswahl eines InfoLabels

Für eine direkte Interaktion mit der Word-Cloud muss der Mauszeiger über ein Label gefahren werden. Dieses Mausereignis wird von dem jeweiligen Label erkannt (siehe Abschnitt 4.3.2) und entsprechend darauf reagiert. Unabhängig von allen Einstellungen wird das entsprechende Label, im Folgenden als Infolabel bezeichnet, durch einen Rahmen hervorgehoben, was im oberen Bereich der Abbildung 4.7 veranschaulicht wird (ohne Hervorhebung der Kookkurrenzen). Da der Rahmen eine Breite von mindestens einem Pixel hat, würde die Schrift des Infolabels um eben diese Breite verschoben werden. Um dieses unschöne Verhalten zu verhindern, wird das Infolabel gleichzeitig um die Rahmenbreite nach links verschoben, damit der Text, nachdem beide Transformationen beendet wurden, seine ursprüngliche Position einnimmt. Verlässt der Mauszeiger das Infolabel, so muss diese Positionsverschiebung wieder rückgängig gemacht und das Infolabel gelöscht werden. Um eine mehrmalige Verschiebung desselben Labels zu verhindern (der Mauszeiger kann ein Label auch verlassen ohne ein Mausereignis auszulösen, beispielsweise durch einen Programmwechsel) kommt die Eigenschaft „moved“ eines Labels zum Einsatz, die anzeigt, ob das Label bereits verschoben wurde oder nicht. Neben der Hervorhebung durch einen Rahmen setzt das Mausereignis noch weitere Änderungen in Gang, die teilweise von getroffenen Einstellungen abhängen und an entsprechender Stelle erläutert werden (Informationsbereich und Hervorhebung der Kookkurrenzen). Eine dieser Änderungen betrifft die statistischen Informationen zu dem Infolabel und ist ebenfalls in Abbildung 4.7 dargestellt. Es ist deutlich zu erkennen, dass auf der linken Seite der Abbildung Informationen angezeigt werden, obwohl kein Infolabel exisitert. Dies sind die Anzahl der in der Word-Cloud dargestellten Tags („Actual tags“) sowie die Gesamtzahl der aus dem Text extrahierten Tags („Overall tags“). Darunter wird die Gesamtzahl der Sätze angezeigt, die der Text enthält, sofern kein Infolabel existiert. Existiert ein Infolabel, so wird die Anzahl der Sätze angezeigt, in denen

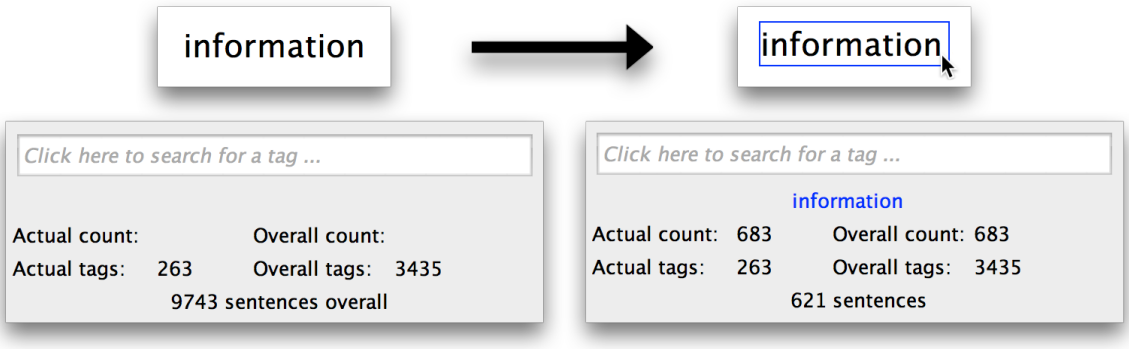


Abbildung 4.7.: Auswahl eines InfoLabels

dieses vorkommt[3]. Wurden Labels ausgewählt (siehe Abschnitt 4.4.5), so wird die Anzahl der gemeinsamen Sätze angezeigt. Wird der Mauszeiger nun über ein Label gefahren, so wird die häufigste Wortform des Infolabels in blauer Farbe angezeigt. Handelt es sich bei dem Infolabel um ein Stoppwort, so wird es in roter Farbe dargestellt. Die beiden übrigen Informationen betreffen die Häufigkeit des Infolabels. „Overall count“ gibt die absolute Häufigkeit an, die sich auf das gesamte Textkorpus bezieht. Mit „Actual count“ wird die relative Häufigkeit des Labels bezeichnet, welche durch den Einsatz von Auswahl (siehe Abschnitt 4.4.5) und Filtern (siehe Abschnitt 4.4.6) beeinflusst wird.

4.4.3. Informationsbereich

Der Informationsbereich wird in Abbildung 4.8 dargestellt und ist in drei tabellenartig aufgebaute Bereiche unterteilt. Jeder Bereich besitzt zwei Spalten, wobei die zweite Spalte stets die Anzahl des Objektes der ersten Spalte darstellt. Da jede Zeile (Überschriftzeilen ausgenommen) folglich eine Anzahl beinhaltet, werden die Zeilen absteigend nach dieser Anzahl sortiert, wodurch sichergestellt werden kann, dass das häufigste Element einer Tabelle an erster Stelle steht. Je nach Fenstergröße ist es möglich, dass der Tabelleninhalt nicht komplett dargestellt werden kann. Aus diesem Grund kann jede Tabelle nach unten gefahren werden[4]. Der oberste Bereich enthält Informationen über die unterschiedlichen Vorkommen eines Tags, die in der Benutzeroberfläche als „Word forms“ bezeichnet werden. Bei aktivierten Multiwörtern werden in diesem Bereich ebenfalls die Multiwörter aufgelistet,

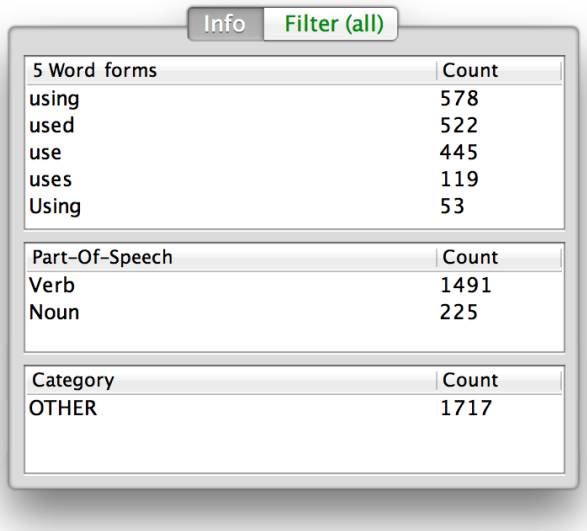


Abbildung 4.8.: Informationsbereich mit Infolabel „using“

[3]Voraussetzung dafür ist die Hervorhebung der Kookkurrenzen (siehe Abschnitt 4.4.9)
[4]jede Tabelle ist in ein „JScrollPane“ (javax.swing) eingebettet

in denen das Tag auftritt (siehe Abschnitt 4.4.9). Die Summe der einzelnen Vorkommen bildet die Gesamtanzahl des Tags. Der mittlere und untere Bereich funktioniert jeweils nach dem gleichen Prinzip und listet die unterschiedlichen Wortarten beziehungsweise Kategorien auf, die den Wortvorkommen während der Verarbeitung der Textkorpora zugewiesen wurden. Wie in der Abbildung zu erkennen ist, sind die Wortarten auf das gesamte Tag bezogen, nicht auf jede einzelne Wortform separat. Dies hat den Vorteil, dass die Datenstruktur (siehe Abschnitt 4.3.1) keine große Komplexität annimmt, was sich positiv auf Speicherverbrauch und Rechenzeit auswirkt. Jedoch hat diese Repräsentation den Nachteil, dass kein Rückschluss zwischen Wortform und Wortart möglich sein muss. Da das gesamte Tag in der Word-Cloud durch ein einziges Label repräsentiert wird, macht diese Zusammenfassung der Wortarten durchaus Sinn. Um die Wortformen getrennt zu betrachten, steht es dem Anwender frei, die Lemmatisierung zu deaktivieren (siehe Abschnitt 4.4.9). Die Anzahl der Kategorie (1717) und die aufsummierten Anzahlen der Wortformen (1717) weichen von den aufsummierten Anzahlen der Wortarten (1716) ab, da die fehlende Wortart durch ein gesetztes Minimum (mit einem Wert ≥ 2) entfernt wurde (siehe Abschnitt 4.4.11).

4.4.4. Suchfunktion

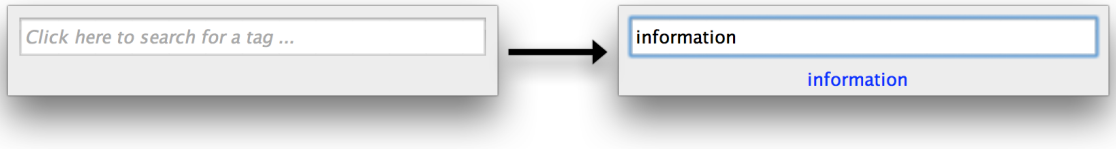


Abbildung 4.9.: Suchfunktion: links inaktiv, rechts aktiv

Um ein bestimmtes Wort zu suchen oder Informationen darüber zu erhalten, steht dem Anwender eine Suchfunktion zur Verfügung, die in Abbildung 4.9 dargestellt ist. Bei Inaktivität des Suchfeldes wird dessen Inhalt mit einem Platzhaltertext gefüllt, welcher sobald das Suchfeld aktiv wird, wieder entfernt wird. Mit der Taste „Escape“ kann der Inhalt des aktiven Suchfeldes gelöscht werden. Sobald ein Text in das Suchfeld eingegeben wird, werden sämtliche Tags nach dem eingegebenen Text durchsucht. Dazu wird sowohl das Lemma eines Tags als auch die Liste der Wortformen während der Eingabe durchsucht. Kann keine Übereinstimmung gefunden werden (Groß- und Kleinschreibung wird hierbei nicht beachtet) so bleiben die Informationsbereiche leer (siehe Abschnitte 4.4.2 und 4.4.3). Kann eine Übereinstimmung gefunden werden, so wird das Tag des aktuellen Infolabels durch das gefundene Tag ersetzt und in den Informationsbereichen angezeigt. Ist das gesuchte Tag darüber hinaus Bestandteil der Word-Cloud, so wird das Mausereignis des entsprechenden Labels simuliert, um dieses hervorzuheben (siehe Abschnitt 4.4.2). Abhängig von der getroffenen Einstellung wird der Mauszeiger auf dieses Label verschoben (siehe Abschnitt 4.4.10). Um die Suche nach einem Multiwort (beispielsweise „New York City“) zu erleichtern, existieren drei Möglichkeiten, die einzelnen Bestandteile des Multiwortes einzugeben:

1. Zusammen („newyorkcity“)

2. durch Leerzeichen getrennt („new york city“)
3. eine Kombination aus 1. und 2. („newyork city“ oder „new yorkcity“)

Neben der beschriebenen Suche steht dem Anwender eine weitere Funktion zur Verfügung: Wird in dem aktiven Suchfeld die Taste „Return“ gedrückt, so wird das gesuchte Tag (bei positiver Übereinstimmung) ausgewählt (siehe Abschnitt 4.4.5). Um auf diese Weise mehrere Tags gleichzeitig auswählen zu können, besteht die Möglichkeit, diese, mit Leerzeichen oder Komma getrennt, in das Suchfeld einzugeben und der Auswahl hinzuzufügen. In diesem Fall bleiben die Informationsbereiche leer, da mehrere Wörter gesucht werden und die Suche während der Eingabe für einzelne Tags konzipiert ist.

4.4.5. Auswahl

Eine der wichtigsten Interaktionsmöglichkeit mit der Word-Cloud stellt die Auswahl dar, die in Abbildung 4.10 zu sehen ist. Ein Weg, der Auswahl Tags per Suchfeld hinzuzufügen, wurde in Abschnitt 4.4.4 bereits erläutert. Einen weiteren Weg stellt das Anklicken eines Labels dar. Hierbei ist zu unterscheiden, ob das angeklickte Label bereits Teil der Auswahl ist oder nicht. Ist das Label kein Bestandteil der Auswahl, so wird es der Auswahl hinzugefügt. Ist das Label aber Bestandteil der Auswahl und wird angeklickt, so wird es aus der Auswahl entfernt. In jedem Fall wird eine neue Word-Cloud generiert, welche ebenfalls in Abbildung 4.10 dargestellt ist. Welche Tags in den jeweiligen Word-Clouds angezeigt werden, hängt von den Kookkurrenzen ab, welche in Abschnitt 4.4.9 beschrieben werden. Sobald eine Auswahl existiert, wird die Schaltfläche „Clear Selection“ anklickbar, wodurch die gesamte

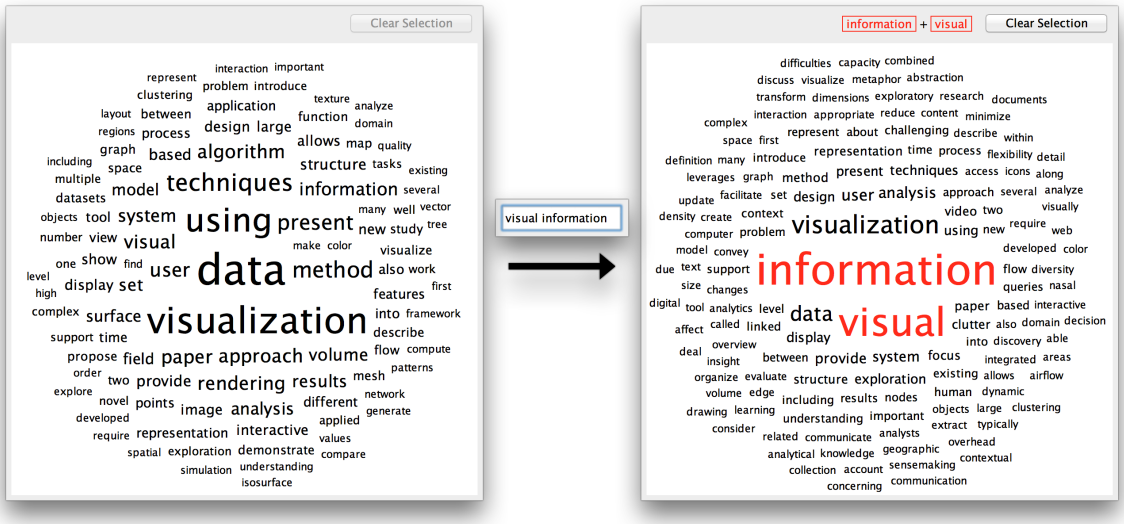


Abbildung 4.10.: Auswahl der Tags „information“ und „visual“

Auswahl geleert werden kann. Die Auswahl kann ebenso mithilfe des Kurzbefehls „Control+Backspace“ geleert werden. Um einzelne Tags aus der Auswahl zu entfernen, sind die ausgewählten Tags unmittelbar über der Word-Cloud aufgelistet und können per Mausklick aus der Auswahl entfernt werden. Diese Auflistung hat darüber hinaus den Vorteil, dass stets zu sehen ist, welche Tags ausgewählt wurden, denn abhängig von den gewählten Filtern (siehe Abschnitt 4.4.6) müssen die ausgewählten Tags nicht zwangsläufig Element der Word-Cloud sein. Da die Funktionalitäten und Ergebnisse der Word-Cloud stark von der Auswahl abhängig sind, werden die ausgewählten Labels stets durch eine rote Schrift hervorgehoben, um Missverständnissen vorzubeugen.

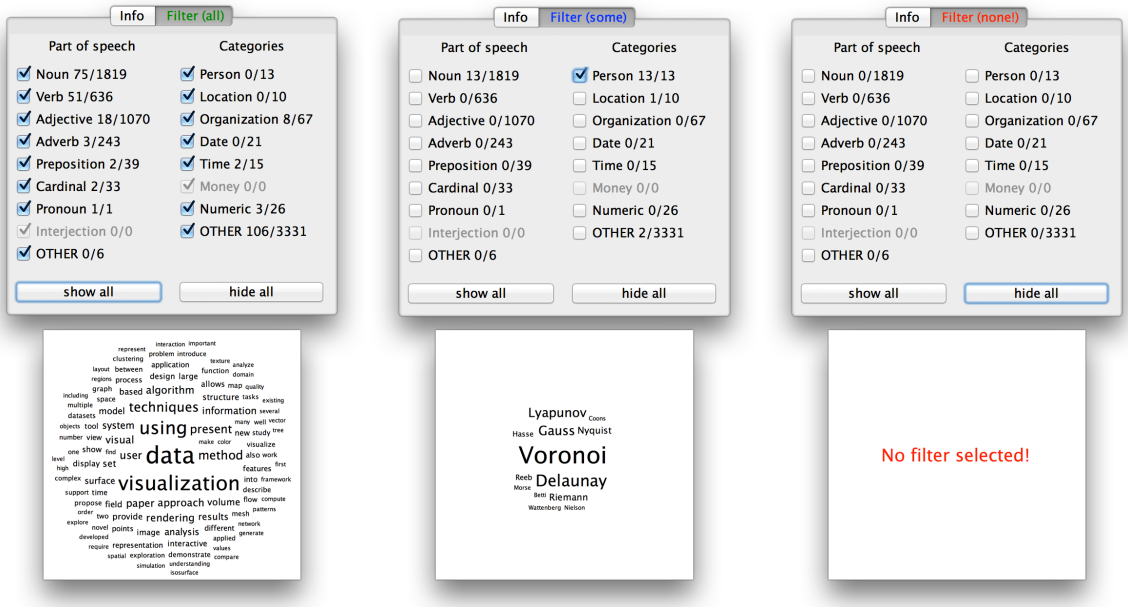
4.4.6. Filterfunktion

Die Filterfunktion ist einer der großen Vorteile für die Word-Cloud, welcher durch die linguistische Verarbeitung von Textkorpora ermöglicht wird. Bei den Filtern ist zu beachten, dass es sich um additive[5] Filter handelt. Werden also die Filter „Verb“ und „Noun“ ausgewählt (also deren Checkboxen aktiviert) und alle anderen Filter deaktiviert, so erscheinen in der Word-Cloud ausschließlich Verben und Substantive. Aus der Gesamtmenge der Tags wird pro Filter die jeweilige Untermenge (beispielsweise Substantive) extrahiert und mit den Untermengen anderer aktiver Filter vereinigt, was schließlich als Grundlage für die Erstellung der Word-Cloud dient. Standardmäßig sind alle Filter ausgewählt, um in der initialen Word-Cloud einen guten Überblick zu ermöglichen. Eine Filterung erfolgt bei jeglicher Änderung der Filterkonstellation.

Wie in Abbildung 4.11 zu sehen ist, sind die Filter in zwei Gruppen unterteilt. Auf der linken Seite befinden sich die Filter bezüglich der Wortarten, während auf der rechten Seite entsprechend die Filter bezüglich der Kategorien dargestellt sind. Jede der beiden Filtergruppen deckt die Menge aller Tags ab, da diejenigen Tags, die keinem Wortart- beziehungsweise Kategorie-Filter zugeordnet werden können, jeweils dem Filter „OTHER“ zugeordnet werden. Da die Filter additiv arbeiten, hat eine Änderung der Kategorie-Filter keinerlei Auswirkung, sofern alle Wortart-Filter aktiviert wurden, und vice versa. Unterhalb der Filter stehen dem Anwender zwei Schaltflächen zur Verfügung, um alle Filter gleichzeitig zu aktivieren („show all“) oder zu deaktivieren („hide all“). Um die Auswirkungen der Filter zu verdeutlichen, sind die daraus resultierenden Word-Clouds ebenfalls in Abbildung 4.11 zu sehen. Der in der Abbildung oben abgebildete Reiter „Filter“ enthält abhängig von der Filterkonstellation eine entsprechende Information („all“, „some“ oder „none!“) und auch die Schriftfarbe wird angepasst, um den Anwender auf die Filterkonstellation aufmerksam zu machen, besonders wenn der Reiter nicht aktiv ist.

Hinter jedem Filter sind zwei durch einen Schrägstrich getrennte Zahlen abgebildet. Die hintere Zahl bezieht sich auf die Menge aller Tags und stellt die Anzahl der Tags dar, welchen dieser Filter zugeordnet wurde. Die vordere Zahl bezieht sich hingegen lediglich auf die in der Word-Cloud dargestellten Tags. Anhand der Abbildung 4.11 kann diese

[5] in Anlehnung an die additive Farbmischung [Daho6]



(a) alle Filter sind aktiv (b) ein Filter ist aktiv (c) kein Filter ist aktiv

Abbildung 4.11.: Filterkonstellationen mit resultierenden Word-Clouds

Zahlendarstellung leicht nachvollzogen werden. Als Beispiel soll die Kategorie „Person“ dienen. In (a) sind alle Filter aktiv und insgesamt wurden in dem Text 13 Personen erkannt, von welchen jedoch keine in der Word-Cloud dargestellt wird, da deren Häufigkeiten zu gering für die Größe der Word-Cloud sind. Folglich steht hinter Person 0/13. In (b) ist lediglich der Filter „Person“ aktiv und die Größe der Word-Cloud lässt es zu, dass alle 13 Personen dargestellt werden können, weshalb hinter Person diesmal 13/13 zu sehen ist. Da in (c) kein Filter aktiv ist und dementsprechend auch keines der Tags in der Word-Cloud dargestellt wird, steht hinter Person wieder 0/13. Trifft ein bestimmter Filter auf kein Tag zu (aus der Menge aller Tags), im Beispiel der Filter „Money“, so wird dieser Filter in hellem Grau dargestellt, um zu verdeutlichen, dass er keinerlei Einfluß auf die Word-Cloud hat.

Da eine sinnvolle Änderung der Filterkonstellation auch eine Änderung der Menge an Tags, die der Word-Cloud zugrunde liegt, nach sich zieht, hat der Anwender neben der vorgestellten Aktivierung und Deaktivierung von Filtern eine weitere Möglichkeit, mit den Filtern zu arbeiten. Diese Möglichkeit besteht darin, sich eine Vorschau der Filter anzeigen zu lassen, und bietet sich insbesondere dann an, wenn der Anwender unsicher ist, welche Filter zielführend sind. Um von dieser Filtervorschau Gebrauch zu machen, muss der Mauszeiger über den entsprechenden Filter gefahren werden, was in Abbildung 4.12 veranschaulicht wird. In dieser Abbildung ist zu sehen, dass einerseits die statistischen Informationsfelder (zwischen dem Suchfeld und den Filtern) Informationen über den entsprechenden Filter anzeigen, andererseits diverse Tags in der Word-Cloud hervorgehoben werden. Bei der Hervorhebung der Tags, auf die der Filter zutrifft (hier: Adjektive), wird zwischen zwei

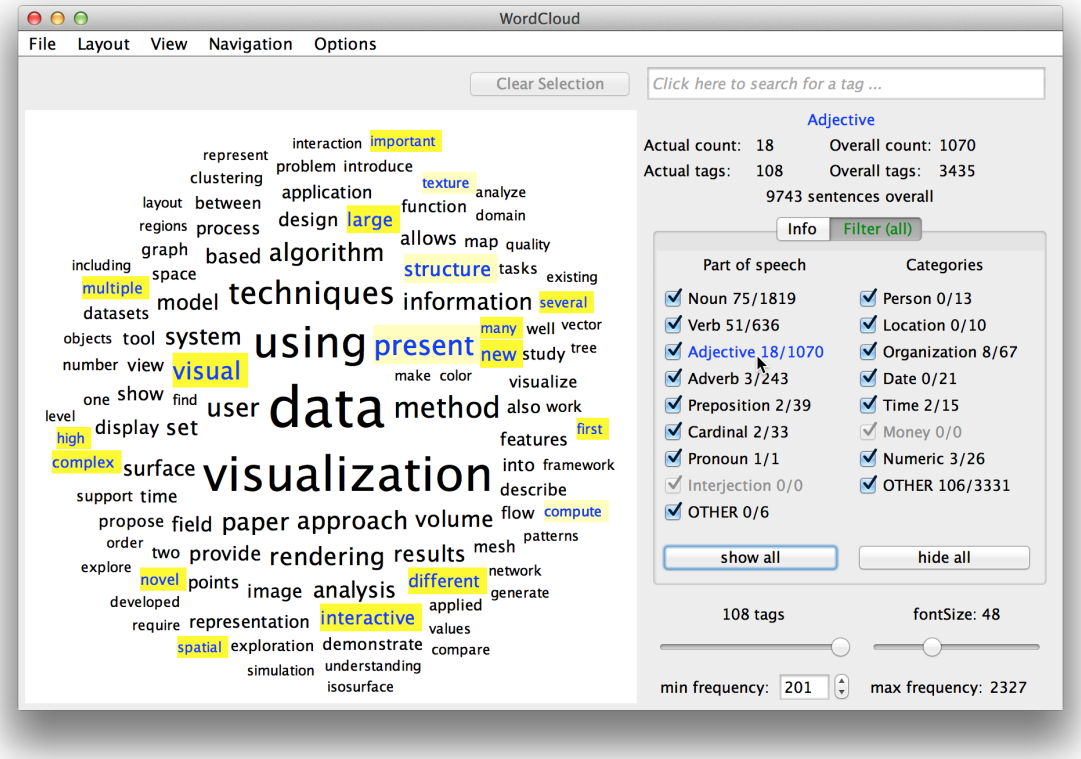


Abbildung 4.12.: Filtervorschau für Adjektive

Stufen unterschieden. Die erste Stufe ist für Tags konzipiert, die zwar durch die Verarbeitung des Textkorpus als die entsprechende Wortart beziehungsweise Kategorie erkannt wurden, also in welchen „Adjective“ in der Liste der Wortarten enthalten ist, jedoch mindestens eine weitere Wortart häufiger in der Liste zu finden ist. Als Beispiel hierfür soll das Tag „texture“ dienen, welches im Text neun mal als Adjektiv und 221 mal als Substantiv erkannt wird. Da Substantiv die häufigste Wortart von „texture“ ist, wird es bei dem Filter „Adjective“ mit der ersten Stufe, also hellem Gelb, hervorgehoben. Die zweite Hervorhebungsstufe, ein vollständig gesättigtes Gelb, kommt dann zum Einsatz, wenn das Tag den Filter als häufigste Wortart beziehungsweise Kategorie hat, wie es beispielsweise bei dem Tag „visual“ der Fall ist. Da die Vorschaufunktion der Filter lediglich die Menge der in der Word-Cloud dargestellten Tags als Grundlage hat, sind die hervorgehobenen Tags unter Umständen unvollständig. Im Gegenzug ist die Berechnung der hervorzuhebenden Tags nicht von der Gesamtanzahl an Tags abhängig, sondern lediglich von der Anzahl der in der Word-Cloud dargestellten Tags, was bei einer Skalierung des Textumfangs einen enormen Vorteil bietet.

4.4.7. Der Menüreiter „File“

Im Folgenden werden die Menüreiter von links nach rechts und innerhalb eines Menüreiters von oben nach unten vorgestellt.

Wie in Abbildung 4.13 zu sehen ist, beherbergt der Menüreiter Datei („File“) die Menüeinträge „New input“ und „Exit and save“. „New input“ öffnet das Eingabefenster, welches in Abbildung 4.14 dargestellt ist und im nachfolgenden Abschnitt detailliert beschrieben wird. Dieses Eingabefenster kann über die Schaltfläche „Cancel“ geschlossen werden. Wird jedoch eine neue Word-Cloud erstellt, so schließt sich das bestehende Hauptfenster automatisch. Jegliche Ergebnisse der Verarbeitung von Textkorpora sowie eventuell geänderte Einstellungen werden in diesem Fall verworfen. Um diese Einstellungsänderungen und Ergebnisse zu speichern, muss der zweite Menüeintrag („Exit and save“) ausgewählt werden. Hierbei wird sowohl die interne Konfigurationsdatei persistiert (siehe Abschnitt 4.3.5) als auch die Klasse „SortedLabels“ als binäre Textdatei gespeichert. Anschließend wird das Programm beendet.

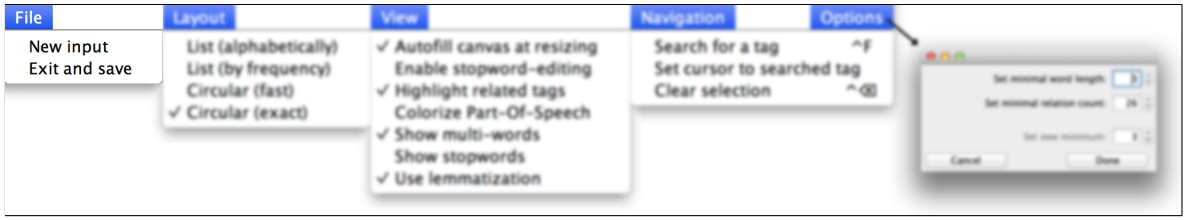


Abbildung 4.13.: Der Menüreiter „File“

Eingabefenster

Das Eingabefenster wird bei jedem Programmstart aufgerufen. Über den Menüreiter „File“ steht einerseits die Funktion „Open textfile“ zur Verfügung, welche einen Dateiauswahldialog öffnet, der ausschließlich Textdateien akzeptiert. Andererseits besteht die Möglichkeit, das Programm mit „Exit“ zu beenden (falls das Hauptfenster im Hintergrund geöffnet ist, wird lediglich das Eingabefenster geschlossen). Um einen Text auszuwählen, hat der Anwender mehrere Möglichkeiten. Die Schaltfläche „Choose file“ öffnet ebenfalls den eben erwähnten Dateiauswahldialog. In die Textfläche kann Text eingegeben oder eingefügt werden. Sie ermöglicht darüber hinaus eine Anpassung des eingefügten oder geladenen Textes. Eine weitere Möglichkeit, Text hinzuzufügen, besteht darin, eine Textdatei mit gedrückter Maustaste in das Textfeld zu ziehen und die Maustaste loszulassen. Diese Funktion besitzt den Vorteil, dass sie nicht auf Textdateien beschränkt ist. Hat ein Anwender bereits einen Text verarbeiten lassen und gespeichert, so kann diese binär gespeicherte Datei direkt in das Textfeld gezogen werden. Anhand der Dateiendung „.data“ wird erkannt, dass es sich um eine binär gespeicherte Textdatei handelt. Auf diese Weise muss der Text nicht erneut verarbeitet werden, was Zeitersparnis bedeutet. Enthält das Textfeld, wie im rechten Bild zu sehen ist, einen Text, so wird umgehend die Anzahl der Wörter zusammen mit einer groben

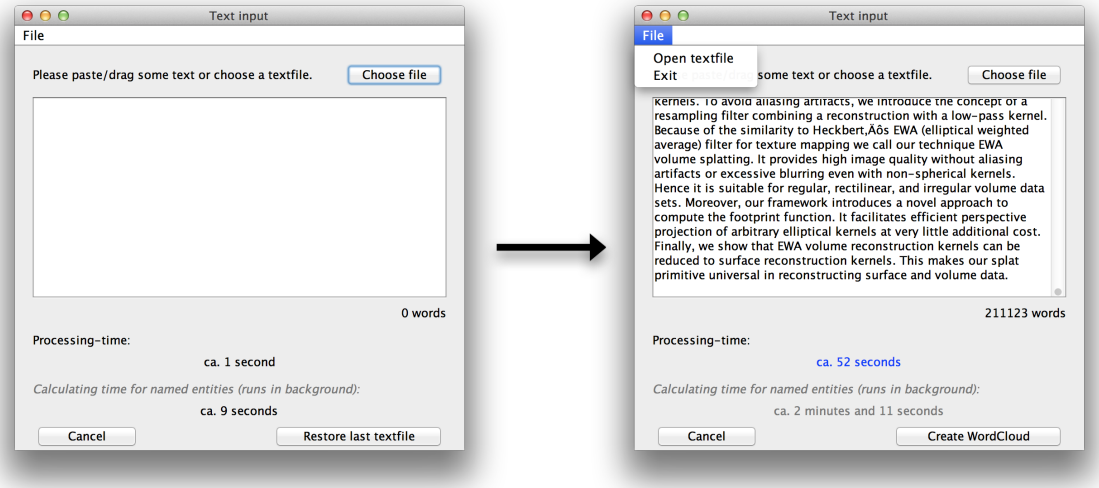


Abbildung 4.14.: Eingabefenster

Schätzung der Berechnungsdauer angezeigt. Dass im linken Bild ebenfalls eine Schätzung angezeigt wird, hat eine Optimierung des Arbeitsablaufes als Hintergrund. Unabhängig von der Textlänge muss das CoreNLP-Framework Modelle laden und durch annotierte Texte trainiert werden (siehe Abschnitt 3.5). Diese beiden Vorgänge dauern insgesamt etwa 10 Sekunden, abhängig von der Rechenleistung. Da das Hinzufügen des Textes ohnehin einige Sekunden in Anspruch nimmt, werden diese beiden Vorgänge im Hintergrund angestoßen und die Anzeige sekündlich aktualisiert. Ist der erste Vorgang beendet, wird die Berechnungsdauer („Processing-time“) blau dargestellt, um sich von dem anderen Text abzuheben. Ist auch der zweite Vorgang abgeschlossen wird, die Berechnungsdauer für die Kategorisierung kursiv und grau dargestellt, ihre Priorität ist eher untergeordnet, da der Vorgang im Hintergrund läuft. Konnte ein Vorgang nicht abgeschlossen werden, so wird dieser an entsprechender Stelle erneut angestoßen, weshalb der Anwender nicht zum Warten gezwungen wird. Die Betätigung der Schaltfläche „Cancel“ in der linken unteren Ecke schließt das Eingabefenster. Die Schaltfläche „Restore last textfile“ in der rechten unteren Ecke öffnet das Hauptfenster und schließt ebenfalls das Eingabefenster. Wenn das Textfeld keinen Text enthält, wird versucht, die zuletzt gespeicherte Textdatei wiederherzustellen. Schlägt dies fehl, was der Fall ist, wenn die entsprechende Datei an dem erwarteten Ort nicht vorhanden ist oder einen anderen Namen trägt, so erscheint eine leere Word-Cloud. Sobald mindestens ein Wort im Textfeld steht, ändert sich die Beschriftung der Schaltfläche auf „Create WordCloud“ und in diesem Fall wird der Verarbeitungsprozess des Textkorpus angestoßen und anschließend die Word-Cloud generiert.

4.4.8. Der Menüreiter „Layout“

Unter dem nächsten Menüreiter „Layout“ kann die gewünschte Word-Cloud-Darstellung ausgewählt werden, was in Abbildung 4.15 zu sehen ist. Es handelt sich dabei um zwei Listendarstellungen und eine zirkuläre Form mit zwei Varianten. Eine Übersicht der unterschiedlichen Layouts ist in Abbildung 4.16 dargestellt, wobei Layout (a), die alphabetisch sortierte Liste, als Standardlayout eingestellt wurde.

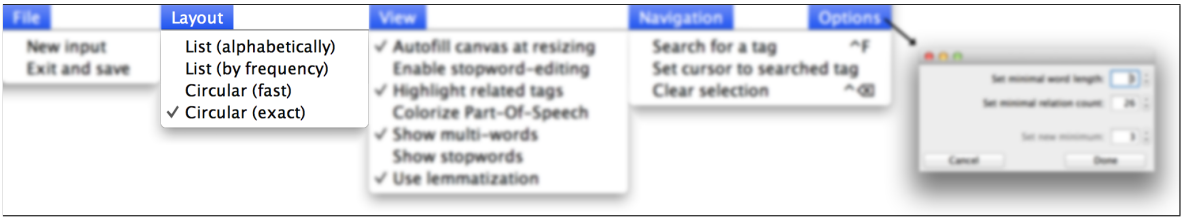


Abbildung 4.15.: Der Menüreiter „Layout“

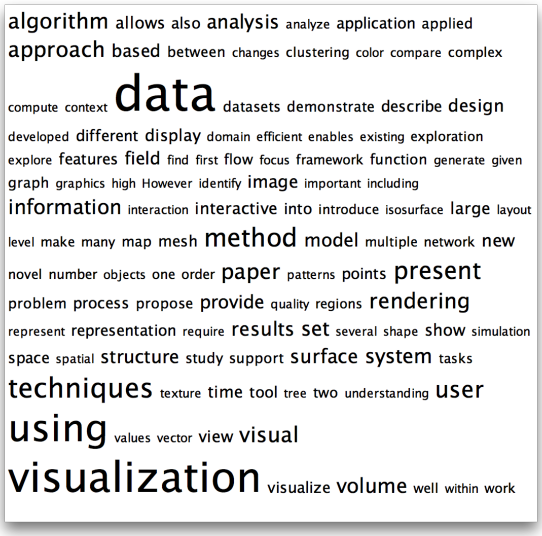
Alphabetisch sortierte Liste

Die Darstellungsform der alphabetisch sortierten Liste (siehe Abbildung 4.16 (a)) kommt den Word- oder Tag-Clouds, die einem Internetnutzer im Alltag begegnen können[6], sehr nahe und ist deshalb standardmäßig voreingestellt. Aufgrund der alphabetischen Sortierung können gesuchte Wörter sehr schnell gefunden werden, was einen Vorteil gegenüber den anderen Layouts darstellt. Die Platzierung der Labels erfolgt zeilenweise. Für jedes Label wird überprüft, ob in der aktuellen Zeile noch ausreichend viel Platz zur Verfügung steht. Ist dies der Fall, so wird das Label der Zeile hinzugefügt. Passt das Label nicht mehr in die aktuelle Zeile, so wird die aktuelle Zeile fest platziert und eine neue Zeile begonnen, die bereits das Label beinhaltet, das nicht mehr in die vorherige Zeile passt. Die Höhe einer Zeile wird durch die maximale Höhe ihrer Labels definiert. Dieses Verfahren wird solange fortgesetzt, bis entweder keine Labels mehr zu platzieren sind oder die Höhe der aktuellen Zeile die verfügbare Zeilenhöhe überschreitet. Besonders bei dem alphabetisch sortierten Layout kann auf diese Weise viel Platz verloren gehen, wenn benachbarte Labels sehr unterschiedliche Höhen besitzen.

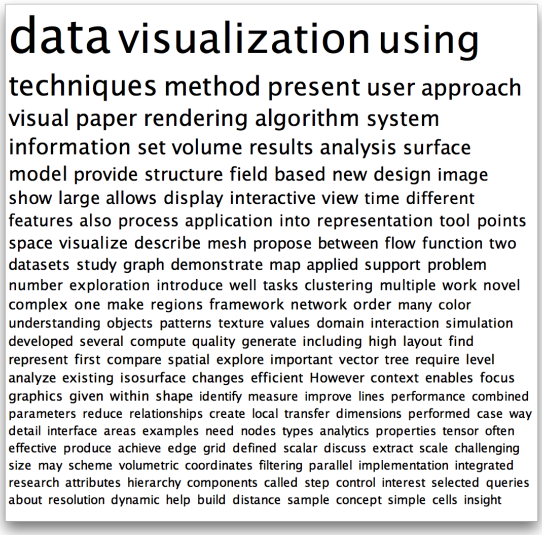
Nach Häufigkeit sortierte Liste

Die nach Häufigkeit sortierte Darstellungsform (siehe Abbildung 4.16 (b)) bietet gegenüber der alphabetischen gewisse Vorteile. Die Platzierung der Labels erfolgt analog zu der oben beschriebenen Platzierung der alphabetisch sortierten Labels. Aufgrund der Sortierung

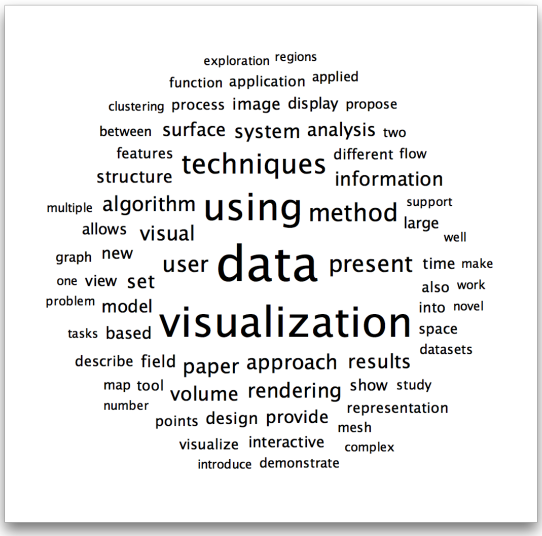
[6] http://www.iphone-ticker.de, unter „Schlagwörter“, http://www.duden.de, unter „Häufig gesucht“, http://texblog.org, unter „Tag Cloud“



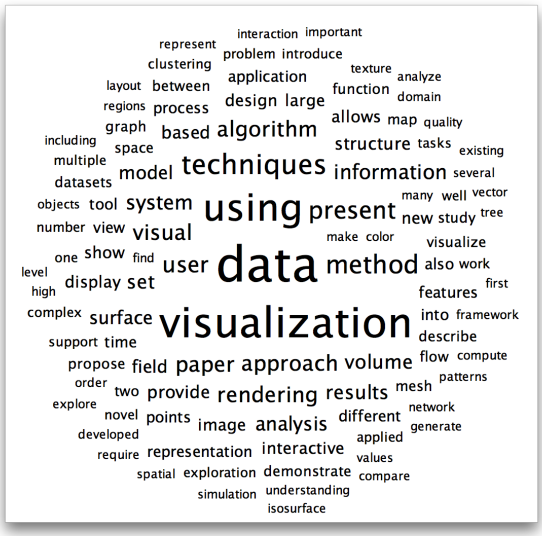
(a) List (alphabetically), 119 *Tags*



(b) List (by frequency), 186 *Tags*



(c) Circular (fast), 73 *Tags*



(d) Circular (exact), 108 *Tags*

Abbildung 4.16.: Übersicht der verschiedenen Layouts mit jeweiliger Taganzahl

nach Häufigkeit ist für die zeilenweise Darstellung sichergestellt, dass benachbarte Labels minimale Höhenunterschiede aufweisen. Diese Tatsache hat den Effekt, dass jede Zeilenhöhe stets minimal ist, was zur Folge hat, dass maximal viele Zeilen und somit auch maximal viele Tags darstellbar sind. In Abbildung 4.16 ist deutlich zu erkennen, dass das nach Häufigkeit sortierte Layout (b) verglichen mit den drei anderen Layouts bei gleicher Fläche die meisten Tags darstellen kann. Darüber hinaus ist bei dieser Sortierung sichergestellt, dass jedes Tag, das rechts oder unterhalb des aktuellen Tags dargestellt wird, nicht häufiger auftauchen kann als das aktuelle. Diese Information kann bei den übrigen drei Layouts lediglich anhand der dargestellten Größe geschätzt werden.

Zirkuläre Darstellung

Die zirkuläre Platzierung der Labels hat folgenden, in Abbildung 4.17 veranschaulichten Ablauf: Das häufigste Label wird mittig platziert. In absteigender Reihenfolge werden die übrigen Labels nun einzeln um das erste Label herum positioniert. Dabei kann ein Label nur dann platziert werden, wenn dessen Rahmen keinen anderen Rahmen bisher platzierter Labels schneidet (auch bekannt als „bounding box“ [SHH99]). Um dies sicherzustellen, muss jeder Punkt auf jeder Kante des Rahmens auf Kollisionen überprüft werden. Um diese Kollisionsüberprüfung möglichst speicher- und laufzeitschonend zu bewerkstelligen, kommt hierfür eine zweidimensionale Matrix in der Größe des Darstellungsbereiches zum Einsatz, die binäre Werte enthält. Anfangs werden die Werte der Matrix mit *falsch* initialisiert. Wurde keine Kollision entdeckt, so kann das Label platziert werden und die Werte der Matrix an der Position des Labels werden auf *wahr* gesetzt. Sobald ein Punkt mit einem anderen Label kollidiert (also *wahr* in der Matrix steht), muss eine neue Position für das Label gesucht werden. Die Positionssuche verläuft in konzentrischen Kreisen im Uhrzeigersinn um den Mittelpunkt herum. Konnte auf einem kompletten Kreis keine kollisionsfreie Position für

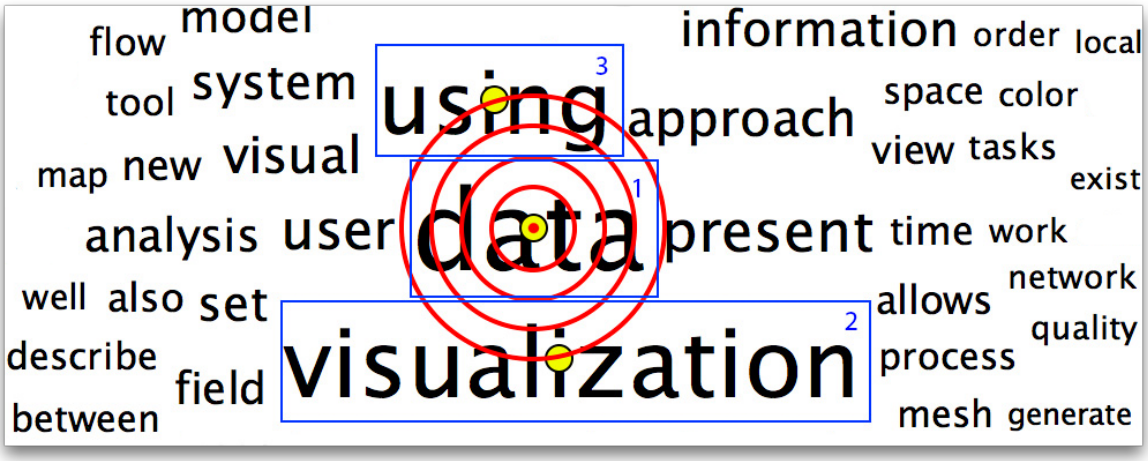


Abbildung 4.17.: Zirkuläre Darstellung: Die Rahmen der Labels sind blau dargestellt, deren Mittelpunkt gelb und die Positionssuche rot.

das Label gefunden werden, so wird der Radius des Kreises erhöht. Sobald entweder keine Labels mehr zu platzieren sind oder der Radius die zur Verfügung stehende Breite oder Höhe des Darstellungsbereiches überschreitet, wird der Algorithmus beendet.

Der Unterschied zwischen den beiden zirkulären Darstellung besteht in der Anzahl ihrer Platzierungssimulationen. Bei der schnellen Variante „Circular (fast)“ wird eine einzige Platzierungssimulation durchgeführt und die dabei ermittelte Anzahl an Labels wird anschließend platziert. Die exakte Variante „Circular (exact)“ führt mehrere Platzierungssimulationen durch und platziert die Labels, sobald die optimale Anzahl an Labels (analog zu den Berechnungen in Abschnitt 4.4.9) ermittelt wurde. Wie in Abbildung 4.16 unter (c) und (d) zu sehen ist, kann dies großen Einfluss auf die Anzahl der platzierten Labels haben.

4.4.9. Der Menüreiter „View“

Unter dem Menüreiter Ansicht („View“) können diverse Einstellungen vorgenommen werden. In Abbildung 4.18 ist zu sehen, wie diese Einstellungen standardmäßig vorkonfiguriert sind.

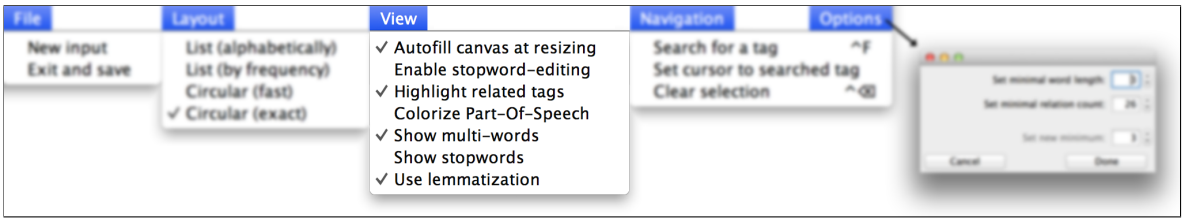


Abbildung 4.18.: Der Menüreiter „View“

Arbeitsfläche bei Größenänderung automatisch füllen

Bei der Berechnung der optimalen Anzahl an Labels ist folgende Problematik zu beachten: Die Größe der einzelnen Labels richtet sich nach den Extremwerten der Häufigkeiten darstellbarer Labels (höchste und niedrigste Häufigkeit). Deshalb müssen die Labelgrößen für jede Mengenänderung angepasst werden. Durch die Anpassung der Labelgrößen kann sich wiederum die Mächtigkeit der darstellbaren Labels ändern. Dies gilt insbesondere für das alphabetisch sortierte Layout, in dem die Position eines Labels, welches der aktuellen Darstellung hinzugefügt werden soll, vorher nicht feststeht. Zusammenfassend lässt sich über dieses Verhalten folgende Aussage treffen: Nur bei einer vollständigen Platzierung aller darzustellenden Labels wird diesen Labels ihre korrekte Schriftgröße zugewiesen. Sobald auch nur ein Label nicht platziert werden kann, sind alle Labelgrößen neu zu berechnen, um deren Korrektheit zu gewährleisten. Auf diese Weise kann die optimale Anzahl an darstellbaren Labels iterativ ermittelt werden, angefangen bei der Mächtigkeit aller Labels (oder einem festgelegten Maximum, siehe Abschnitt 4.4). Um den optimalen Wert zu finden, müssen alle Werte überprüft werden, bis hin zum ersten Wert, für den alle Labels platziert werden können. Bei diesem Ansatz entspricht die Anzahl der Iterationen im schlechtesten

Fall der Anzahl der darzustellenden Labels (falls das häufigste Label nicht dargestellt werden kann). Wird der Algorithmus in umgekehrter Reihenfolge durchlaufen, so tritt der schlechteste Fall ein, wenn alle Labels dargestellt werden können. Demzufolge weist der Algorithmus an sich (ohne Betrachtung der Platzierung) eine lineare Komplexität auf. Die Berechnung kann jedoch nach dem Prinzip der binären Suche optimiert werden. Algorithmus 4.1 veranschaulicht den Ablauf dieser optimierten Berechnung.

Algorithmus 4.1 Berechnung der optimalen Anzahl an Labels

$maxCount \leftarrow |sortedLabels|$
$step \leftarrow maxCount$
while $step > 0$ **do**
  $step \leftarrow \begin{cases} 0 & step = 1 \\ \left\lfloor \frac{step+1}{2} \right\rfloor & \text{otherwise} \end{cases}$
  if $maxCount - (\text{count of placeable labels}) > 0$ **then**
    $maxCount \leftarrow maxCount - step$
  else
    $maxCount \leftarrow maxCount + step$
  end if
end while
$maxCount \leftarrow (\text{count of placeable labels})$

Da die Schrittweite in jedem Schleifendurchlauf halbiert wird, ergibt sich für die Anzahl der Iterationen folgende Gleichung:

$$i = \begin{cases} 1 & |L_d| < 2 \\ \lceil \log_2 |L_d| \rceil + 1 & |L_d| \geq 2 \end{cases}$$

Wobei i für die Anzahl der Schleifendurchläufe und L_d für die darzustellenden Labels steht. Obwohl der optimierte Algorithmus an sich eine logarithmische Komplexität aufweist, ist der Rechenaufwand erheblich höher als ein einmaliger Durchlauf. Abhängig von der gegebenen Rechenleistung kann dieser Mehraufwand (für mehrmaliges Durchlaufen der Schleife) für den Benutzer zu wahrnehmbaren Verzögerungen führen. Aus dem Grund ist diese exakte Berechnung optional. Alternativ kann eine beliebige Anzahl an darzustellenden Labels gewählt werden, die jedoch entweder die Arbeitsfläche nicht optimal ausfüllt oder aber inkorrekte Labelgrößen anzeigen kann. Für das alphabetisch sortierte Layout können darüber hinaus bei Verkleinerung der Arbeitsfläche inkorrekte Labels angezeigt werden, da nicht die seltensten Labels weggelassen werden, sondern die alphabetisch kleinsten. Um auf diesen Umstand hinzuweisen, wird die Anzeige der Labelanzahl rot gefärbt, zusammen mit einer entsprechenden Kurzinfo, wie es in Abbildung 4.19 zu sehen ist. Generell wird die Aktivierung der automatischen Füllung des Arbeitsbereiches („Autofill canvas at resizing“) dringend empfohlen und ist deshalb auch standardmäßig voreingestellt.

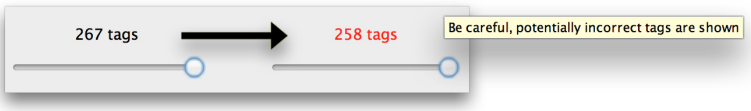


Abbildung 4.19.: Warnhinweis bei potentiell inkorrekter Darstellung

Stoppwortbereich aktivieren

Wie in Abschnitt 3.4 beschrieben, stellt die Entfernung der Stoppwörter einen wichtigen Schritt in der Textvorverarbeitung dar. Da es sich im Rahmen dieser Arbeit jedoch um einen interaktiven Ansatz handelt, sollten diese Anpassungen ebenfalls möglichst interaktiv gestaltet werden. Infolge dessen werden die Stoppwörter nicht destruktiv entfernt, sondern lediglich als Stoppwort markiert (siehe Abschnitt 4.3.1). Die Stoppwortliste ist als reine Textdatei mit geringem Aufwand durch jeden beliebigen Texteditor anpassbar, jedoch ist dieses Vorgehen während des Einsatzes des Programms nicht sehr praktikabel. Aus dem Grund steht dem Anwender ein eigener Bereich für diese Bearbeitung zur Verfügung, welcher in Abbildung 4.20 dargestellt ist. Da die Stoppwörter für ein Textkorpus nur einmalig angepasst werden müssen, wird der Bereich nicht generell angezeigt. Über den Menüpunkt „Enable stopword-editing“ (siehe Abbildung 4.18) kann der Bereich ein- beziehungsweise ausgeblendet werden. Sobald Text in das Textfeld eingegeben wird, wird dieser mit der

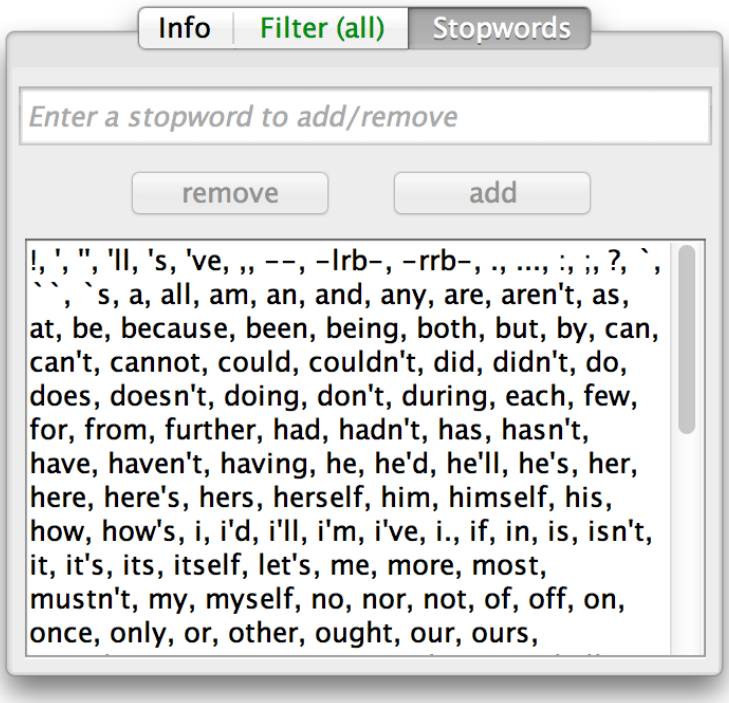


Abbildung 4.20.: Stoppwortbereich

Stoppwortliste abgeglichen. Abhängig davon, ob der Text in der Liste enthalten ist, wird eine der Schaltflächen „add“ oder „remove“ aktiv. Wird die entsprechende Schaltfläche betätigt, so wird der Text der Liste hinzugefügt beziehungsweise aus dieser entfernt. Darüber hinaus werden die Stoppworteigenschaften aller Tags aktualisiert. Da eine solche Änderung Einfluss auf die Word-Cloud haben kann, wird diese anschließend neu gezeichnet. In dem Listenbereich ist die gesamte Stoppwortliste zu sehen, um eine Überprüfung der getätigten Änderung zu ermöglichen und einen Überblick über die Stoppwörter zu bieten.

Kookkurrenzen hervorheben

Um die Kookkurrenzen eines Wortes zu berechnen, wird wie folgt vorgegangen: Jedes Wort hat Informationen über die Sätze, in denen es auftritt, und jeder Satz hat Informationen über die darin vorkommenden Wörter zusammen mit deren Häufigkeit. Sollen nun die Kookkurrenzen eines ausgewählten Wortes bestimmt werden, so werden zunächst alle Sätze, in denen das Wort auftritt, überprüft. Bei dieser Überprüfung werden alle anderen Wörter in den betreffenden Sätzen gezählt. Somit sind nach der Überprüfung alle Wörter, die mindestens einmal gezählt wurden, kookkurrent zu dem ausgewählten Wort. Ähnlich verhält es sich, wenn mehrere Wörter ausgewählt wurden. In dem Fall wird jedoch zunächst die Schnittmenge der Satzlisten der ausgewählten Wörter gebildet und diese dann überprüft.

Kookkurrenzen stellen einen wichtigen Teil der interaktiven Word-Cloud dar. Um dem Anwender einen Überblick über die Kookkurrenzen zu ermöglichen, kann die Menüoption „Highlight related tags“ (siehe Abbildung 4.18) aktiviert werden. Dieser Überblick ist in Abbildung 4.21 veranschaulicht. Wird der Mauszeiger auf ein Label gefahren, so werden umgehend die Kookkurrenzen für dieses Label berechnet. Für die Berechnung dienen je-

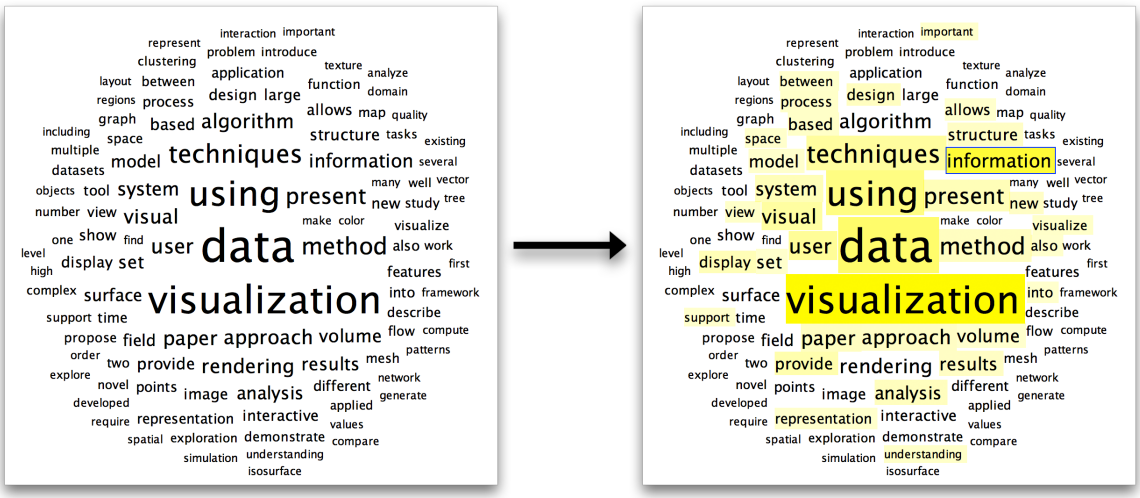


Abbildung 4.21.: Hervorhebung der Kookkurrenzen (hier: Kookkurrenzen zu „information“)

doch nicht alle Tags als Grundlage, sondern nur die in der Word-Cloud dargestellten. Auf diese Weise bleibt die Hervorhebung auch bei einer extrem hohen Anzahl an Tags noch verwendbar, da der Rechenaufwand nicht von der Skalierung abhängt. In der Abbildung ist bei genauer Betrachtung zu sehen, dass die Sättigungsstufen der gelben Hervorhebungen von Label zu Label unterschiedlich sind. Die Sättigungsstufe korreliert mit der Häufigkeit des kookkurrenten Tags. Ist die Sättigung niedrig (wie beispielsweise bei „important“, oben in Abbildung 4.21), so tauchen die beiden Tags in nur wenigen Sätzen (Analoges gilt für Abschnitte) gemeinsam auf, bei hoher Sättigung haben die Tags eine entsprechend größere Anzahl an Sätzen gemein. Analog zu der Größenberechnung für Labels (siehe Abschnitt 4.4.8) wird die Sättigung anhand der minimalen und maximalen Häufigkeit gemeinsamer Auftreten bestimmt. Das eigentliche Label (hier: „information“) wird für diese Berechnung allerdings außer Acht gelassen, da es ohnehin in jedem gemeinsamen Satz auftaucht und somit den Bereich der Sättigungsstufen unnötig stauchen würde.

Wortarten hervorheben

Unter dem Menüpunkt „Colorize Part-Of-Speech“ (siehe Abbildung 4.18) kann eine farbliche Hervorhebung der Wortarten aktiviert beziehungsweise deaktiviert werden. In Abbildung 4.22 ist diese farbliche Hervorhebung dargestellt. Die Farben der Wortarten können in der internen Konfiguration definiert werden. Als Farblegende dienen die Wortartfilter (siehe Abschnitt 4.4.6), welche bei Aktivierung der Hervorhebung die jeweilige Farbe annehmen, was ebenfalls in Abbildung 4.22 veranschaulicht ist. Jedem Label der Word-Cloud wird eine eindeutige Farbe anhand der jeweiligen Wortart zugewiesen. Da Labels mehrere Wortarten besitzen können, wird stets die häufigste davon ausgewählt, bei gleicher Anzahl wird die erste davon verwendet um die Konsistenz zu wahren (die Reihenfolge spielt dabei keine Rolle, solange sie nicht verändert wird). Eine Ausnahme bilden ausgewählte Labels, deren Farbe nicht von der Hervorhebung beeinflusst wird.

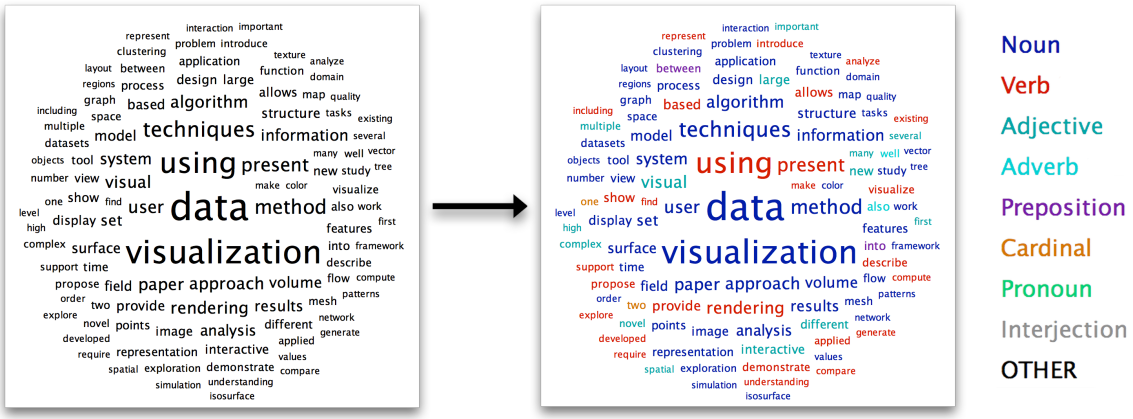


Abbildung 4.22.: Farbliche Hervorhebung der Wortarten mit nebenstehender Farblegende

Multiwörter anzeigen

Über den Menüpunkt „Show multi-words“ (siehe Abbildung 4.18) können Multiwörter ein- beziehungsweise ausgeschalten werden. Multiwörter, wie beispielsweise „Mr. Sherlock Holmes“ oder „New York City“, können anhand ihrer Wortart erkannt werden. Jedem Teil eines Multiworts wird durch das CoreNLP-Framework die Wortart „NNP“ oder „NNPS“ zugeordnet [pos, MMS93, Seite 317]. Diese beiden Abkürzungen stehen für „proper noun, plural“ beziehungsweise „proper noun, singular“, zu deutsch „Eigennamen in Singular- beziehungsweise Pluralform“. Da Wörter stets einzeln vorliegen, müssen Multiwörter aus den einzelnen Teilwörtern zusammengesetzt werden. Die Zusammensetzungsvorschrift lautet wie folgt: Jedes Wort, dessen Wortart eine der oben erwähnten ist, gilt als potentielles Multiwort. Ein potentielles Multiwort wird erst dann zum Multiwort, wenn das darauf folgende Token ebenfalls ein potentielles Multiwort ist, da ein Multiwort aus mindestens zwei Teilwörtern bestehen muss. Sobald dem Multiwort ein beliebiges Token anderer Wortart folgt, ist das Multiwort komplett. Auf diese Weise lassen sich beliebig lange Multiwörter der Mindestlänge zwei erkennen. Dieser Ansatz beruht auf der Annahme, dass Eigennamen stets durch Satzzeichen oder mindestens ein anderes Wort voneinander getrennt werden.

Die Visualisierung eines Multiwortlabels muss sich von der anderer Labels unterscheiden, damit Multiwörter auch als solche erkannt werden. Falls zwei oder mehr Wörter gleicher Schriftgröße nebeneinander dargestellt werden, so kann der Anwender nicht unterscheiden, ob diese Wörter absichtlich oder zufällig nebeneinanderstehen. Um dies zu vermeiden, kamen mehrere Ansätze für die Hervorhebung der Multiwörter in Frage, die in Abbildung 4.23 anhand des Multiwortes „New York“ dargestellt sind. Darstellung (a) verwendet eine vordefinierte Transparenz. Darunter leidet allerdings die Lesbarkeit des Labels und abhängig von dem gewählten Wert der Transparenz hebt sich das Label zu wenig von den anderen Labels ab. Werden darüber hinaus die Wortarten farblich hervorgehoben, so kann es durch die Transparenz leicht zu Verwechslungen kommen. Darstellung (b) verbindet die Teile des Multiwortes mit einem Unterstrich. Das Multiwort bleibt lesbar und hebt sich von den anderen Wörtern ab. Darstellung (c) verwendet die CamelCase-Formatierung. Dadurch hebt sich das Multiwort ausreichend von den übrigen Wörtern ab und die Lesbarkeit bleibt erhalten. Darüber hinaus werden die Multiwörter kompakter dargestellt, wodurch weniger Leerraum entsteht, was die Word-Cloud ansehnlicher macht. Aufgrund der genannten Vorteile wurde die CamelCase-Darstellung für Multiwörter gewählt.

Durch eine Eigenheit des CoreNLP-Frameworks werden Zeilenumbrüche nicht als Token behandelt und tauchen demzufolge in der Analyse nicht mehr auf. So kann es bei einem Text

(a) (b) (c)

Abbildung 4.23.: Mögliche Darstellungen der Multiwörter

vorkommen, dass beispielsweise eine Überschrift, die üblicherweise keine Satzzeichen enthält, mit einem Multiwort endet und der darauf folgende Satz mit einem Multiwort beginnt. Da sowohl Überschrift als auch der darauf folgende Satz als ein einziger Satz behandelt werden, werden die beiden Multiwörter fälschlicherweise als ein einziges Multiwort erkannt, da sie durch kein Satzzeichen oder anderes Wort getrennt sind. Diese und analoge Situationen bilden jedoch die Ausnahme und werden daher nicht weiter verfolgt.

Wie in der Vorstellung des Informationsbereiches erwähnt, werden Multiwörter gemeinsam mit den Wortformen aufgelistet. Da Multiwörter jedoch auch als eigenständige Tags existieren, ist deren Anzahl in der Auflistung mit einem negativen Vorzeichen behaftet, da sie nicht zu dem aktuellen Tag gezählt werden, sondern zu dem Tag des Multiwortes. Als veranschaulichendes, vereinfachtes Beispiel soll das Wort „Sir“ verwendet werden, welches in dem Textkorpus [Doyo1] sowohl alleinstehend (350 mal) als auch in den Multiwörtern „Sir Charles“ (91 mal) und „Sir Henry“ (151 mal) auftritt und in Abbildung 4.24 dargestellt ist. Sind Multiwörter deaktiviert, so kommt das Tag „Sir“ 350 mal vor. Sind Multiwörter jedoch aktiviert, so werden die Vorkommen von „Sir“ in den Multiwörtern zu dem jeweiligen Multiwort gezählt und müssen folglich von dem Tag „Sir“ subtrahiert werden um nicht doppelt gezählt zu werden. Somit kommt das Tag „Sir“ an sich lediglich 108 mal vor.

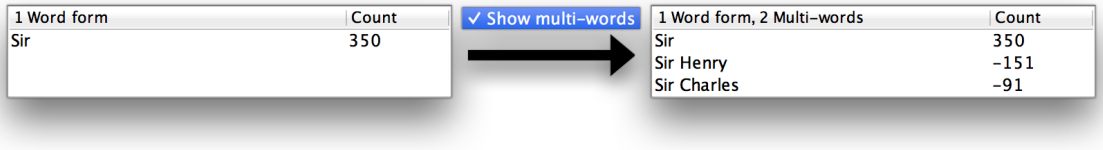


Abbildung 4.24.: Multiwörter im Informationsbereich (für das Infolabel „Sir“)

Stoppwörter anzeigen

Wie in Abschnitt 4.4.9 beschrieben, sind die Stoppwörter interaktiv konzipiert. Infolge dessen ist es dem Anwender möglich, mit einem Klick auf den Menüpunkt „Show stopwords“ (siehe Abbildung 4.18) alle Stoppwörter ein- oder auszublenden. Diese Option ist von der Darstellung des Stoppwortbereiches unabhängig und kann jederzeit wie ein Filter verwendet werden. Wurden die Stoppwörter eingeblendet, wie in Abbildung 4.25 dargestellt, so wird ihre Wirkung auf die Word-Cloud deutlich sichtbar: Einerseits werden die häufigsten Wörter wie „data“ und „visualization“ in den Hintergrund gedrängt, andererseits sinkt die Bandbreite der dargestellten Labelgrößen erheblich. Dies rührt von der Tatsache her, dass Stoppwörter eine sehr viel größere Häufigkeit aufweisen als alle anderen Wörter. So kommt das häufigste Wort (ohne Stoppwörter) „data“ beispielsweise 2327 mal vor, wohingegen das häufigste Stoppwort „the“ 12035 mal vorkommt.

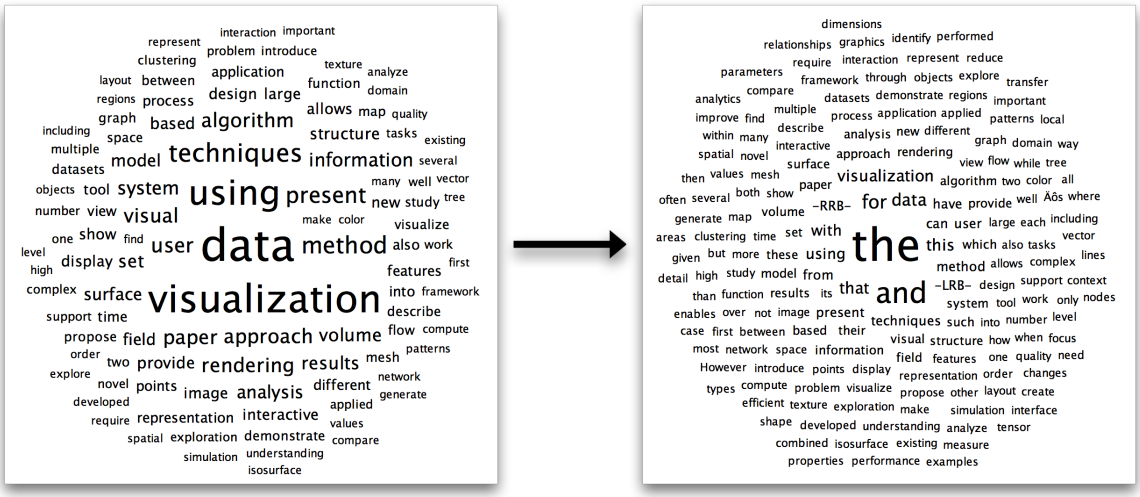


Abbildung 4.25.: Stoppwörter aktivieren

Lemmatisierung verwenden

Die Lemmatisierung stellt ein wichtiges Werkzeug der Verarbeitung von Textkorpora dar (siehe Abschnitt 3.8). Da die Lemmatisierung für gewisse Aufgabenstellungen hinderlich sein kann, wenn beispielsweise Wörter als Tag zusammengefasst werden, die einzeln betrachtet werden sollen, wird es dem Benutzer freigestellt, diese zu verwenden oder darauf zu verzichten. Über den letzten Menüpunkt des Menüreiters Ansicht (siehe Abbildung 4.18) kann die Lemmatisierung ein- und ausgeschalten werden, was in Abbildung 4.26

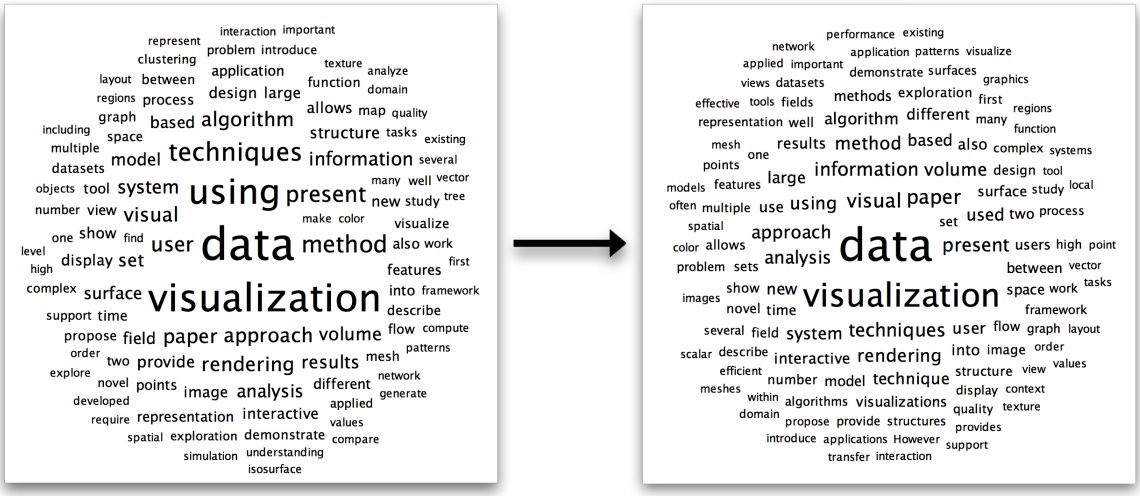


Abbildung 4.26.: Deaktivierung der Lemmatisierung

veranschaulicht ist. Es ist deutlich zu erkennen, dass eines der größten Labels „using“ ohne die Lemmatisierung in seine Bestandteile („using“, „used“, „use“, „uses“) zerfällt und in der Word-Cloud untergeht, während das Wort „data“, welches keine weiteren Wortformen besitzt, seine Größe behält.

4.4.10. Der Menüreiter „Navigation“

Wie in Abbildung 4.27 zu sehen ist, beherbergt der Menüreiter „Navigation“ drei diesbezügliche Menüpunkte, die im Folgenden erläutert werden.

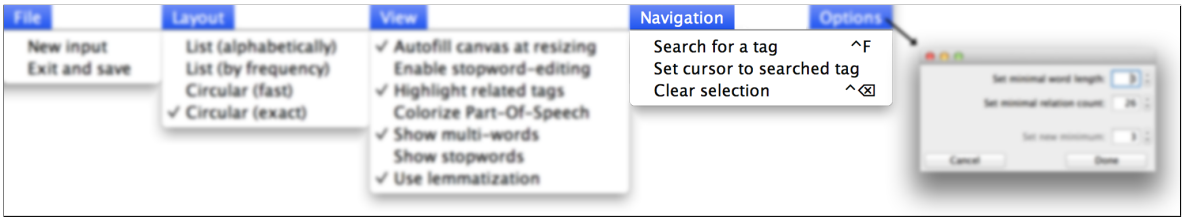


Abbildung 4.27.: Der Menüreiter „Navigation“

Suche

Der erste Menüpunkt „Search for a tag“ bringt den Anwender direkt zum Suchfeld (siehe Abschnitt 4.4.4). Ein weiterer Vorteil dieses Menüpunktes zeigt sich in der Möglichkeit, einem Menüpunkt eine Tastenkombination zuzuweisen. Für die Suche wurde die Tastenkombination „Control+F“ (unter OS X: ^F) gewählt, da diese in vielen Programmen üblich und dadurch für viele Anwender intuitiv ist.

Mauszeigerverschiebung

Mithilfe des zweiten Menüpunktes („Set cursor to searched tag“) hat der Anwender die Möglichkeit, von folgender Funktion Gebrauch zu machen: Wird in dem Suchfeld ein Text eingegeben, zu dem ein passendes Label in der Word-Cloud angezeigt wird, so springt der Mauszeiger automatisch auf dieses Label. Diese Funktion erleichtert dem Anwender die Suche des hervorgehobenen Labels in der Word-Cloud. Da die Verschiebung des Mauszeigers für viele Benutzer verwirrend wirkt, ist diese Funktion standardmäßig deaktiviert.

Auswahl zurücksetzen

Analog zum ersten Menüpunkt ist auch der letzte Menüpunkt der Navigation („Clear selection“) konzipiert. Die Funktionsweise dieses Menüpunktes ist identisch mit der der gleichnamigen Schaltfläche, also dem Zurücksetzen der Auswahl (siehe Abschnitt 4.4.5). Hierfür kommt die Tastenkombination „Control+Backspace“ (unter OS X: ^Delete) zum Einsatz.

4.4.11. Der Menüreiter „Options“

Das Optionsmenü („Options“), welches in Abbildung 4.28 dargestellt ist, stellt eine Ausnahme in der Menüart dar, da es sich nicht um ein Aufklapp-Menü handelt. Der Grund für das eigene Menüfenster sind einerseits dessen Bedienelemente, andererseits die Möglichkeit, getätigte Optionsänderungen über die Schaltfläche „Cancel“ zu widerrufen oder mithilfe der Schaltfläche „Done“ zu bestätigen. Dadurch werden die Optionsänderungen nicht umgehend umgesetzt, was die Bedienbarkeit erheblich verbessert. Welche Einstellungen der Anwender in dem Optionsmenü beeinflussen kann, werden im Folgenden erläutert.

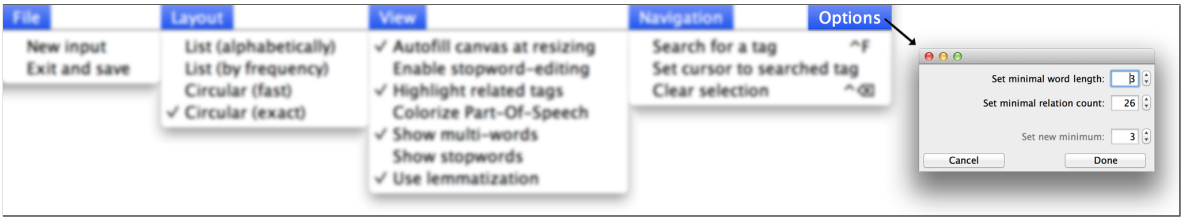


Abbildung 4.28.: Optionen: Ein expliziter Mausklick auf „Options“ öffnet das Optionsmenü

Minimale Wortlänge

Über die erste Einstellung („Set minimal word length“) kann der Anwender die minimale Wortlänge festlegen. Alle Wörter, die weniger Buchstaben besitzen als die eingestellte minimale Wortlänge vorgibt, werden nicht in der Word-Cloud dargestellt. Diese untere Grenze fungiert folglich als Filterfunktion und kann jederzeit beliebig verändert werden. Standardmäßig ist die minimale Wortlänge auf den Wert 3 voreingestellt.

Mindestanzahl gemeinsamen Auftretens

Mithilfe der zweiten Einstellung („Set minimal relation count“) kann die Mindestanzahl des gemeinsamen Auftretens angepasst werden. Diese untere Grenze für die Kookkurrenzen hat zweierlei Auswirkung. Zum einen wird dadurch die Hervorhebung der Kookkurrenzen beeinflusst, was in Abbildung 4.29 veranschaulicht wird. Da die Anzahl der gemeinsamen Sätze stark von dem Umfang des Textes abhängig ist, kann es für den Anwender sehr hilfreich

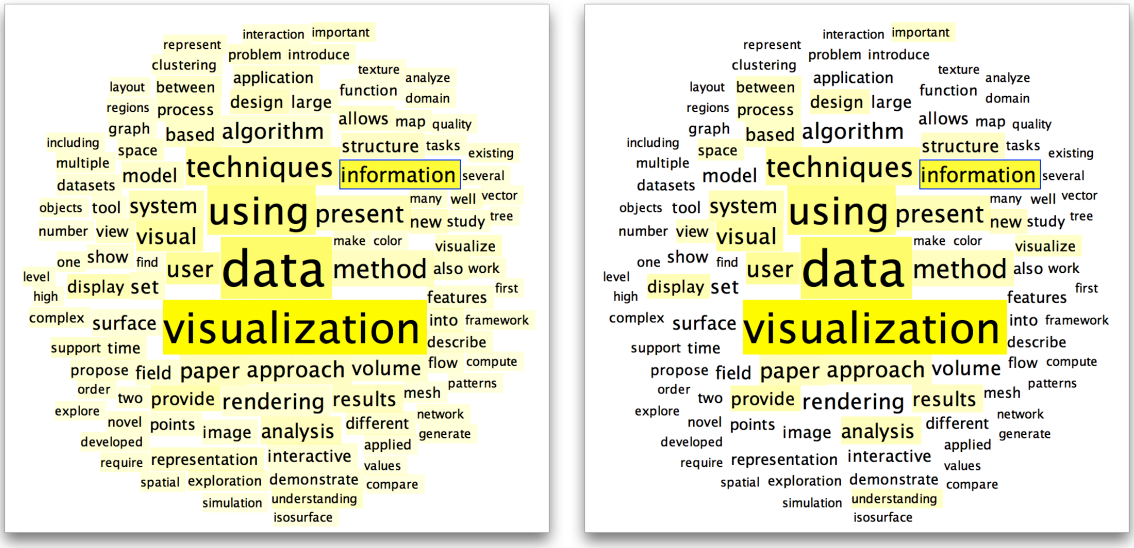


Abbildung 4.29.: Mindestanzahl gemeinsamen Auftretens (links: 1, rechts: 30)

sein, die untere Grenze der Kookkurrenzen dynamisch anpassen zu können. Zum anderen ist die durch die Auswahl eines Wortes entstehende Word-Cloud von der Mindestanzahl des gemeinsamen Auftretens betroffen, was in Abbildung 4.30 anhand des nach Häufigkeit sortierten Layouts dargestellt ist. Dieses Resultat entspricht der Erhöhung der minimalen Häufigkeit, nachdem eine Auswahl getroffen wurde.

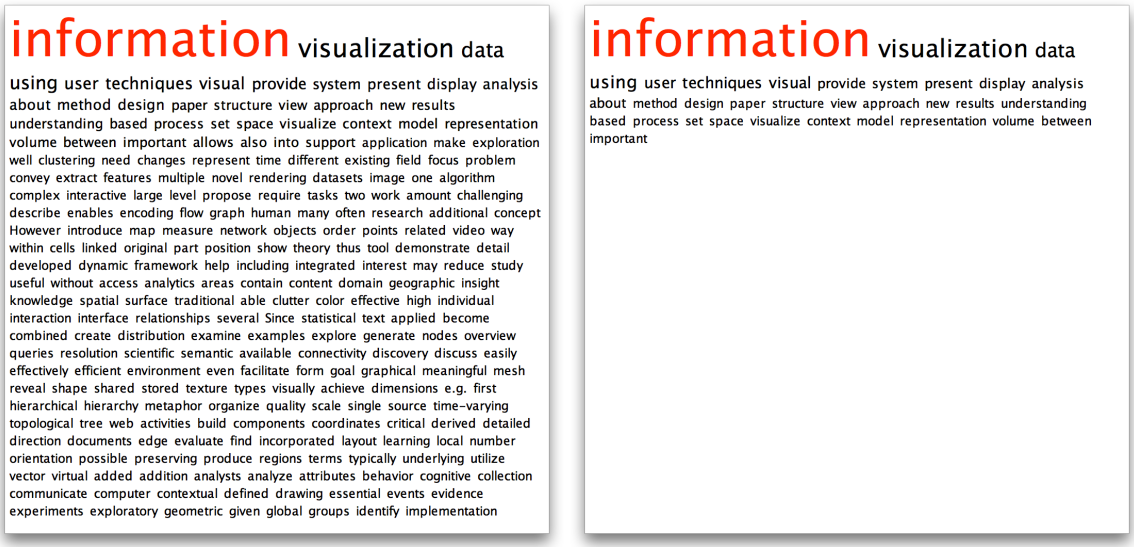


Abbildung 4.30.: Mindestanzahl gemeinsamen Auftretens mit Auswahl (links: 1, rechts: 30)

Minimum

Die letzte Einstellung des Optionsmenüs hebt sich von den anderen beiden deutlich ab, was in Abbildung 4.31 zu sehen ist. Einerseits ist der Abstand zu den anderen Einstellungen größer, andererseits ist die Schriftfarbe heller. Darüber hinaus erscheint eine Kurzinfo, wenn der Mauszeiger über der Einstellung steht. Diese Hervorhebungen haben folgenden Hintergrund: Das Minimum kann lediglich erhöht werden. Um ein kleineres Minimum zu erhalten, muss der Text erneut verarbeitet werden. Aus dem Grund sollte diese Einstellung nicht unabsichtlich getätigt werden. Bei dem Minimum handelt es sich um eine untere Grenze, welche die Wortformen (wobei Multiwörter auch als Wortform gelistet werden), Wortarten und Kategorien eines Tags betrifft. Kommt eine dieser Tageigenschaften nicht mindestens so oft vor, wie von dem Minimum vorgeschrieben, so wird sie entfernt. Aufgrund der Datenstruktur (siehe Abschnitt 4.3.1) ist es nicht möglich, diese Eigenschaft zu markieren und anschließend zu verbergen, sie kann lediglich komplett gelöscht werden. Da die gelöschten Eigenschaften nicht wiederhergestellt werden können, kann das Minimum nur erhöht werden, weshalb standardmäßig der Wert 1 voreingestellt ist. Hintergrund für das Minimum stellen Ungenauigkeiten und Fehler seitens des CoreNLP-Frameworks und der Textkorpora dar, die durch das Minimum entfernt werden können. Generell wird dem Anwender empfohlen, umfangreiche Textdateien nach ihrer Verarbeitung als binäre Textdatei mit dem standardmäßigen Minimum von 1 zu speichern. Eine Möglichkeit, die erneute Verarbeitung der Textkorpora zu umgehen, besteht darin, die gespeicherte binäre Textdatei zu duplizieren und auf dem Duplikat größere Minima einzustellen. Eine andere Möglichkeit besteht darin, die Textdatei, nachdem größere Minima eingestellt wurden, ohne eine Speicherung zu beenden.

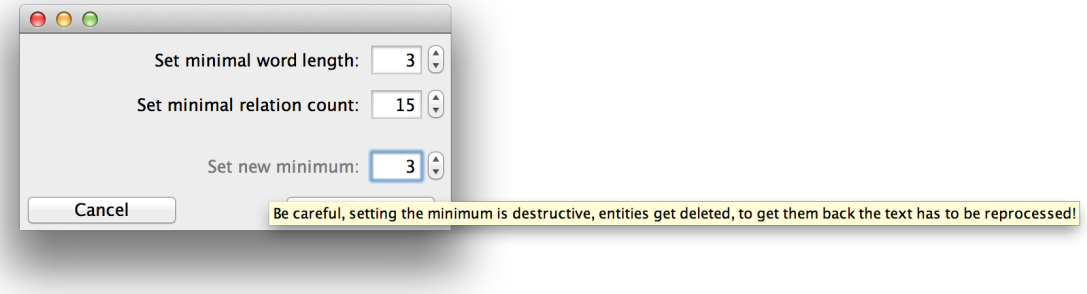


Abbildung 4.31.: Minimum mit Kurzinfo

4.5. Probleme bei der Implementierung

Im folgenden Abschnitt werden die während der Implementierung aufgetretenen Probleme erläutert und diskutiert. Darüber hinaus werden die verwendeten Ressourcen für die Verarbeitung verschiedener Textkorpora dargestellt.

4.5.1. Visualisierungstools

Die Suche nach integrierbaren Visualisierungstools, die eine Word-Cloud erzeugen, blieb ergebnislos. Obwohl solche Ansätze zahlreich im Internet zur Verfügung gestellt werden, ließ sich keiner der Ansätze in das bestehende Programm integrieren. Dies war einerseits der fehlenden Java-Unterstützung seitens der Tools geschuldet, da viele der Tools, wie beispielsweise WordCram [wora] und Wordookie [worb], auf die Programmiersprache „Processing“ [pro] setzen. Andererseits machten Inkompatibilitäten innerhalb von Java den Einsatz von auf Java basierenden Tools, wie beispielsweise Cloudio [clo], zunichte. Grund dafür sind die verwendeten Grafikbibliotheken. In dem bestehenden Programm kamen die Grafikbibliotheken „Swing“ und „AWT“ zum Einsatz, während Cloudio „SWT“ verwendet. Zwischen AWT und SWT bestehen allerdings Inkompatibilitäten, weshalb (ohne größeren Aufwand) lediglich eine der beiden sinnvoll verwendet werden kann.

Dies hatte zur Folge, dass die Layouterzeugung ebenfalls implementiert werden musste, woraufhin eine Implementierung des geplanten „clustered layout“ (siehe Abschnitt 3.1) den Rahmen dieser Arbeit gesprengt hätte.

4.5.2. Lemmatisierung

Bei den durch das CoreNLP-Framework erzeugten Lemmata zeigten sich Anomalien. Aufgabe der Lemmatisierung ist die Überführung einer flektierten Wortform in die Grundform des Wortes (siehe Abschnitt 3.8). Jedoch stellte sich heraus, dass das CoreNLP-Framework diesbezüglich keine zuverlässigen Ergebnisse liefert. Zur Veranschaulichung wird das Verb „justified“ betrachtet, welches die Grundform „justify“ als Lemma besitzt. Das CoreNLP-Framework liefert für „justified“ meist „justify“, manchmal aber auch „justified“ als Lemma zurück. Dies hat fatale Konsequenzen für die generierten Tags, da nun sowohl „justified“ als auch „justify“ als eigenständiges Tag existieren, jedoch nur eines der Tags in der Word-Cloud zu sehen ist und dessen Informationen folglich nicht korrekt sein können, da sich die Vorkommen von „justified“ und „justify“ auf die beiden Tags aufteilen. Darüber hinaus kann über die Suchfunktion nur eines der Tags gefunden werden. Infolge dessen war es nötig, eine Korrekturfunktion der Lemmata zu entwickeln und in diesem Beispiel die Tags „justified“ und „justify“ zu einem einzigen Tag zu vereinen. Diese Funktion besitzt eine quadratische Laufzeit, da für jede Wortform zu prüfen ist, ob sie in weiteren Tags vorkommt. Ist dies der Fall, werden die Tags vereint. Da in den Sätzen die Lemmata der Wortformen gespeichert werden, müssen diese ebenfalls korrigiert werden.

4.5.3. Skalierung

Durch den Einsatz großer Textkorpora zeigte sich, dass die als Label gespeicherten Tags zu schwergewichtig waren, um eine gute Skalierung zu gewährleisten. Da in diesem Fall die Anzahl der dargestellten Labels nur einen Bruchteil der Anzahl an bestehenden Tags darstellt, bot es sich an, die Tags erst zum Zeitpunkt des Zeichnens in Labels zu kapseln. Auf diese Weise kann ein erheblicher Overhead vermieden werden, was eine Verarbeitung größerer Textkorpora ermöglicht.

Darüber hinaus nahm das Einlesen großer Textdateien im Vergleich zu kleineren Dateien unverhältnismäßig viel Zeit in Anspruch. Besonders die Konkatenation vieler Strings brachte das Programm sowohl hinsichtlich des Arbeitsspeichers als auch hinsichtlich der Laufzeit an seine Grenzen. Durch den Einsatz von String-Buffern konnte diesem Problem jedoch Abhilfe geschaffen werden.

4.5.4. Optimierung

Neben der Persistierung verarbeiteter Textkorpora stellte sich heraus, dass weitere Persistierungen ebenfalls sinnvoll sind. Dies betrifft sowohl die Stoppwörter als auch jegliche Konfigurationen und Einstellungen des Programms. An dieser Stelle waren besonders die unterschiedlichen Bearbeitungsmöglichkeiten zu beachten, um die bestmögliche Performanz zu gewährleisten. Darüber hinaus galt es, das Programm entsprechend robust zu gestalten, sodass eine etwaige Abwesenheit persistierter Dateien mithilfe von Standardwerten kompensiert werden kann.

Eine weitere Optimierung stellte die Unterteilung des Verarbeitungsprozesses von Textkorpora dar. Durch die Auslagerung der Kategorieerkennung (NER) (siehe Abschnitt 3.6) kann die Generierung der Word-Cloud wesentlich schneller erfolgen. Mithilfe von Threads kann die Erkennung der Kategorien im Hintergrund erfolgen, während die Word-Cloud bereits dargestellt werden kann. Wichtig war an dieser Stelle, dass die Ergebnisse des NER-Threads nahtlos in die Word-Cloud einfließen, da ein Objekt in Java nicht von zwei Threads zugleich modifiziert werden darf.

4.5.5. Komponente

Der ursprünglich als Komponente konzipierte Ansatz dieser Arbeit stellte sich durch die Verwendung des CoreNLP-Frameworks schnell als problematisch heraus. Der nicht unerhebliche Ressourcenverbrauch dieses Frameworks ist in Abschnitt 4.5.6 zu sehen. Abhängig von dem gewählten Textkorpus und den Annotationen, muss der virtuellen Maschine des Frameworks genügend Speicher zur Verfügung gestellt werden. Wäre die Komponente in ein Programm eingebunden, so müsste in diesem Fall der Speicher der virtuellen Maschine des Programms erhöht werden, was lediglich bei Programmstart festgelegt werden kann. Auf diesen Fakten aufbauend, wurde das Produkt dieser Arbeit als eigenständiges Programm neu konzipiert.

4.5.6. Benötigte Ressourcen für die Verarbeitung von Textkorpora

In Tabelle 4.1 sind die benötigten Ressourcen[7] für die Verarbeitung unterschiedlicher Textkorpora durch das in dieser Arbeit dargestellte Programm zu sehen. Abbildung 4.32 stellt den Speicherverbrauch sowie die benötigte Berechnungszeit grafisch dar. Besonders der Unterschied zwischen dem dritten und vierten Textkorpus ist interessant. Obwohl das vierte Textkorpus nahezu die doppelte Wortmenge und Größe besitzt, wird dafür weniger Zeit benötigt. Dass die Zeit nicht linear mit der Anzahl der Wörter steigt, ist besonders anhand des letzten Textkorpus ersichtlich.

Textkorpus[a]	Größe [KB]	Wörter	Sätze	Tags	Tags (mit Lem.)	RAM [MB]	RAM [MB] (NER)	Zeit [Sek.]	Zeit [Sek.] (NER)
Einzelnes Wort	0	1	1	1	1	497	928	0	0
Ebook	327	55828	3852	5712	4480	614	1080	10	42
Sport-News	749	127399	5664	14027	12356	705	1190	94	197
VisWeek-Abstr.	1454	211123	9743	11981	9206	725	1280	53	170
Bible [gut]	4432	748179	29792	17473	14786	1110	1740	183	633
Reuters-News[b]	17307	2814432	120878	116114	106895	2900	4260	9372	17450

Tabelle 4.1.: Benötigte Ressourcen für die Verarbeitung von Textkorpora

[a] Quellen der Textkorpora: siehe Abschnitt 5.2
[b] Konkatenation einer Woche ungefilterter Reuters-Nachrichten (Zeitraum: 20. – 26.8.1996)

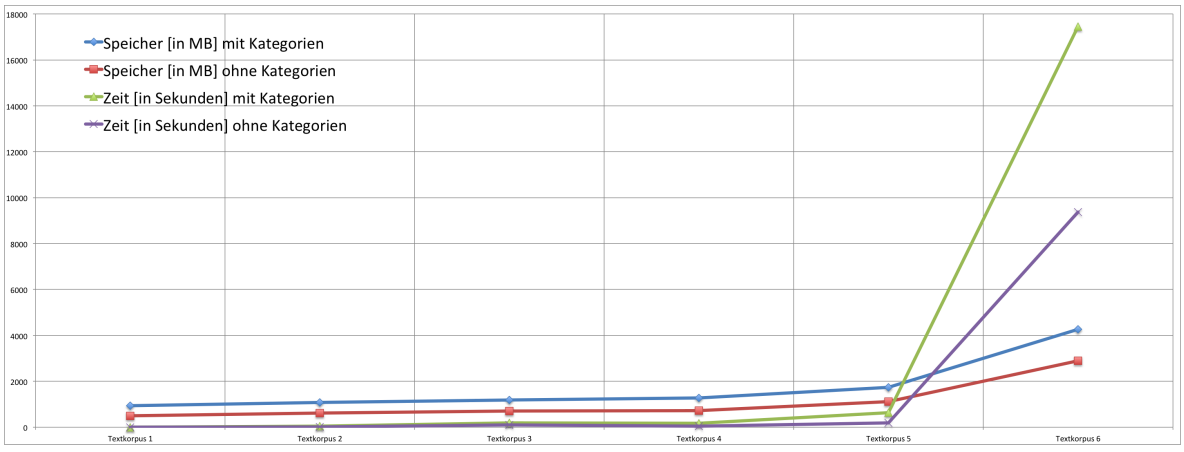


Abbildung 4.32.: Ressourcendiagramm zu Tabelle 4.1

[7] Hardware: MacBook Pro mit 2,6 GHz Quad-Core Intel Core i7 Prozessor, 16 GB 1600 MHz DDR3L RAM

5. Evaluation

Um die Benutzbarkeit des Programms zu überprüfen und außerdem Feedback und Verbesserungsvorschläge einzuholen, war das Mittel der Wahl eine Nutzerstudie. Darüber hinaus sollten diverse Anwendungsgebiete des Programms demonstriert werden um weitere Ideen für einen möglichen Einsatz zu sammeln.

5.1. Vorbereitung

Die Wahl der Studienart fiel aus diversen Gründen auf eine qualitative Studie. Der entscheidende Grund dafür war, dass das Programm mit keinem bisher bestehenden Programm sinnvoll verglichen werden konnte. Darüber hinaus galt das Hauptinteresse der Studie nicht der Fragestellung, wie schnell und präzise ein Teilnehmer alle Aufgaben lösen kann, sondern welche Probleme dabei aufkommen und wie auf die Benutzeroberfläche reagiert wird. Darüber hinaus war das Ziel ein möglichst professionelles und hilfreiches Feedback zu bekommen. Da für solch ein Feedback ein gewisses Hintergrundwissen und ein Mindestmaß an Erfahrungen mit visueller Analyse vorauszusetzen war, wurde der Personenkreis der Teilnehmer eingeschränkt. Laut Nielsen [Nieoo] ist jedoch nicht unbedingt eine große Anzahl an Testpersonen erforderlich, um die meisten Benutzbarkeitsprobleme identifizieren zu können und wertvolle Erkenntnisse zu gewinnen. Deshalb wurde die Anzahl der Teilnehmer auf fünf festgelegt. Da dem Institut für Visualisierung und Interaktive Systeme der Universität Stuttgart viele Mitarbeiter angehören, war es problemlos möglich, fünf Experten zu finden, die über ausreichend viel Erfahrung und Hintergrundwissen bezüglich visueller Analyse verfügen und einwilligten, an der Studie teilzunehmen. Um ein möglichst ausführliches Feedback zu erhalten, kam die „Think-aloud-Methode“ [Holo5] zum Einsatz, bei der die Teilnehmer ihre Gedanken laut formulieren.

5.2. Materialien

Für die Durchführung der Studie wurde ein MacBook Pro 15“ mit Retina Display bei einer Auflösung von 2880x1800 Pixeln eingesetzt. Das Programm wurde im Vollbildmodus gestartet. Um mögliche Farbfehlsichtigkeiten der Teilnehmer auszuschließen, wurde der Ishihara-Test [ish11] verwendet. Jeder Teilnehmer bekam zwei Aufgabenblätter mit insgesamt 25 Fragen unterschiedlichster Art und Komplexität. Für den Fall, dass eine Aufgabe nicht bewältigt werden konnte, lag den Aufgabenblättern ein Blatt mit Hinweisen bei, auf dem zu jeder Aufgabe eine Lösungsstrategie beschrieben war. Darüber hinaus bekam jeder

Teilnehmer noch einen zweiseitigen Fragebogen, der Fragen zur Person, den Aufgaben, der Studie und dem Programm enthielt[1]. Für die Studie kamen folgende Textkorpora zum Einsatz:

- Ein beliebtes E-Book („The hound of the Baskervilles“) [Doyo1] von Projekt Gutenberg [gut]
- Eine Konkatenation nach Sportinformationen gefilterter Artikel der Nachrichtenagentur Reuters (Zeitraum: 20. – 26.8.1996) [reu]
- Eine Konkatenation von Abstracts aus Veröffentlichungen der VisWeek (Zeitraum: 1998-2011) [vis, HKBE12, S. 6]

(weitere Informationen zu den Textkorpora sind in Tabelle 4.1 auf Seite 66 aufgelistet)

5.3. Konfiguration

Tabelle 5.1 zeigt die anfängliche Konfiguration des Programms, welche von den Teilnehmern nach Belieben geändert werden durfte, sofern die Aufgabenstellung keine expliziten Einstellungen vorschrieb.

Reiter	Einstellung	Wert
Layout	Layout	alphabetically
View	Autofill canvas at resizing	true
	Enable stopword-editing	false
	Colorize Part-Of-Speech	false
	Show multi-words	false
	Show stopwords	false
	Use lemmatization	true
Options	Minimal word length	3
	Minimal relation-count	1
	Minimum	2

Tabelle 5.1.: Konfiguration

5.4. Aufgaben

Um die breitgefächerte Anwendbarkeit des Programms zu demonstrieren, wurde für die Aufgaben nicht ein einzelnes, möglicherweise speziell präpariertes, Textkorpus verwendet,

[1]Die vollständigen Fragebögen sind im Anhang A.2 zu finden

sondern die Auswahl fiel auf zwei Textkorpora aus unterschiedlichen Bereichen. Um das Spektrum der Anwendbarkeit noch weiter zu verdeutlichen, wurde für die Einführung in das Programm ein drittes Textkorpus gewählt, welches jedoch nicht Bestandteil der Aufgaben war. Die Aufgabe der Teilnehmer war es, sich in ein hineinzuversetzen, in welchem sie als Analyst mit speziellen Aufgaben konfrontiert sind und mithilfe des Programms nach Lösungen suchen sollen. Durch die Wahl dieses Szenarios sollte eine möglichst realistische Situation für den Einsatz des Programms generiert werden. Die einzelnen Aufgaben waren sowohl auf englisch als auch auf deutsch abgedruckt, um potentielle Missverständnisse durch Formulierungen zu minimieren und Teilnehmern mit geringen Englischkenntnissen Hilfestellung zu leisten. Die Art der Aufgaben hatte durchgehend ein ähnliches Schema, variiert wurde lediglich deren Komplexität und die dafür zur Verfügung stehenden Funktionen. Um ein Beispiel zu nennen, bestand eine Sportnachrichten-Aufgabe etwa darin, den Namen des häufigst genannten olympischen Champions herauszufinden. Eine weitere, auf dem VisWeek-Korpus basierende, Aufgabe bestand darin, herauszufinden, um was es bei der „Nyquist theory“ zu gehen scheint. Bei der Aufgabenerstellung wurde Wert darauf gelegt, möglichst alle Funktionen einzubeziehen, um die Teilnehmer das komplette Programm verwenden zu lassen. Da jedoch für jede Aufgabe diverse Lösungswege möglich sind und die meisten Teilnehmer sich mit dem ersten zielführenden zufrieden geben, mussten entsprechende Vorkehrungen getroffen werden, damit die Teilnehmer sich nicht auf eine kleine Funktionsmenge beschränken konnten. Dazu war es bei einzelnen Aufgaben nötig, bestimmte Funktionen zu verbieten, um dadurch andere in den Vordergrund zu rücken.

5.5. Ablauf

Zunächst wurden die Teilnehmer auf Farbfehlsichtigkeiten getestet. Es folgte eine Einführung in das Programm, bei der alle wichtigen Funktionen erklärt und anhand eines Beispiels [Doy01] demonstriert wurden. Während der Einführung stand es den Teilnehmern frei, Fragen zu dem Programm zu stellen und Funktionen selbst auszuprobieren. Nach Beseitigung aller Unklarheiten begann der Aufgabenteil, wobei das jeweils passende Textkorpus geladen wurde. Die Teilnehmer wurden darum gebeten, Strategien, Probleme und sonstige Gedanken stets laut zu umschreiben und auf möglichst wenig Hintergrund- und Fachwissen zurückzugreifen. Darüber hinaus war es den Teilnehmern erlaubt, Fragen zu dem Programm zu stellen, die nicht die Aufgabe an sich betrafen. Nachdem alle Fragen bearbeitet waren, wurde der Fragebogen ausgeteilt. Sobald dieser ausgefüllt war, folgte ein halbstandardisiertes Interview [int], bei dem die Teilnehmer nochmals die Gelegenheit hatten, Feedback zu geben und das Programm zu beurteilen. Folgende Fragen waren Bestandteil des Interviews:

- Welche Features fanden Sie am besten?
- Welches Layout fanden Sie besonders hilfreich?

5.6. Teilnehmer

Fünf Experten auf dem Gebiet der Visualisierung im Alter von 25 bis 31 Jahren (Median 29) nahmen an der Studie teil. Die Teilnehmer wurden am Institut für Visualisierung und Interaktive Systeme der Universität Stuttgart rekrutiert. Darunter befanden sich vier männliche Teilnehmer und eine weibliche Teilnehmerin. Bei keinem der Teilnehmer konnte eine Farbfehlsichtigkeit festgestellt werden. Die Teilnehmer schätzten ihre Englischkenntnisse auf einer Skala von 1 (gering) bis 10 (sehr gut) auf 8 bis 10 (Median 9). Vier Teilnehmer gaben an, bereits vor der Studie einer Tag-Cloud oder Word-Cloud begegnet zu sein.

5.7. Ergebnisse

Im Folgenden werden zunächst die Ergebnisse präsentiert, die den durch die Teilnehmer ausgefüllten Fragebögen entnommen wurden. Anschließend folgt das durch das halbstandardisierte Interview gewonnene Feedback.

Bezüglich der Aufgabenstellung traten teilweise Unklarheiten im Zusammenhang mit der Wortwahl auf. Besonders oft wurde beispielsweise aus der ersten Frage der Sportnachrichten das Wort „score“ auf unterschiedlichste Weise interpretiert. Die Bearbeitung der Aufgaben bereitete einigen Teilnehmern Probleme. Hierbei handelte es sich stets um Funktionen, welche die Teilnehmer seit der Einführung bereits vergessen hatten und demzufolge nicht an deren Verwendung dachten. Probleme mit der Verwendung der Funktionen tauchten kaum auf. Die einzige Anmerkung hierzu betraf eine zu schwache Hervorhebung eines gesuchten Begriffs. Kein Teilnehmer musste insgesamt mehr als einen Tipp in Anspruch nehmen. Den meisten Teilnehmern kamen die Verhaltensweisen der Funktionen intuitiv vor, in einem Fall wurde jedoch das Hervorheben der Wörter, sobald mit der Maus über einen Filter gefahren wird, als nicht intuitiv bezeichnet, da diese Funktion ohne Erklärung nicht entdeckt worden wäre. Bemängelt wurde auch das Verhalten, dass der Mauszeiger bei erfolgreicher Suche eines Wortes zu dem gefundenen Wort springt. Dieses Verhalten ist jedoch einstellbar.

Alle Teilnehmer waren sich darin einig, dass der Einsatz dieser interaktiven Word-Cloud für bestimmte Aufgaben in der Visualisierung sinnvoll ist. Einen Mehrwert konnten sie sich besonders bei der Suche in Literaturarchiven und dem Herausfinden und Darstellen von Abhängigkeiten vorstellen.

Der Wunsch nach zusätzlichen Funktionen kam bei allen Teilnehmern auf. Es ging dabei beispielsweise um Hilfestellungen wie Tooltipps oder Kontextmenüs, die dem Benutzer verdeutlichen, welche Möglichkeiten ihm zur Verfügung stehen. Eine vollständige Auflistung der gewünschten Funktionen ist in Tabelle 5.2 zu sehen; wobei es sich nicht ausschließlich um neue Funktionen handelt, es wurden ebenso allgemeine Verbesserungsvorschläge genannt. Erläuterungen und dazugehörige Diskussionen sind in Abschnitt 5.8.2 zu finden.

Beschreibung	umgesetzt
Von Filtern/Auswahl abhängige Anzeige von Informationen	nein
Möglichkeit, in den Originaltext zu schauen	ja
Kein Verlust der Auswahl bei Änderungen der View	ja
Löschen des eingegebenen Suchbegriffs nur bei positivem Ergebnis	ja
Suchvervollständigung oder Suchvorschläge	nein
Hervorhebung mehrerer Wörter (ohne Auswahl)	nein
Informationsanzeige für ausgewählte Wörter	ja
Anzeige der Worthäufigkeit in der Word-Cloud (beispielsweise als Pop-out), sobald der Mauszeiger über ein Wort gefahren wird	nein
Hinzufügen von Kontextmenüs	nein
Hinzufügen von Tooltips	ja
Verarbeitung nicht-englischer Texte	nein
Farbliche Hervorhebung der Kategorien	nein
Individuell anpassbare und erweiterbare Kategorien	nein
Split-Screen-Ansicht (um Vergleiche zu erleichtern)	nein
Möglichkeit, einen Filter exklusiv auszuwählen	ja
Zusätzliche Buttons, um nur Wortarten/Kategorien aus-/abwählen zu können	nein
Ein Reset-Button, der die Auswahl aufhebt und Filtereinstellungen zurücksetzt	nein
Verhalten beim Hinzufügen von Suchbegriffen zur Auswahl immer additiv	ja
Umbenennung einiger Label	ja
Selbsterklärende Beschreibung der Zähler	ja
Ständige Anzeige des relativen und absoluten Zählers	ja
Ständige Anzeige der gemeinsamen Sätze	ja
Gruppierung und Positionierung der Zähler	ja

Tabelle 5.2.: Verbesserungsvorschläge

Von den drei zur Verfügung stehenden Layouts wurden für die Aufgaben nur zwei verwendet. Alle Teilnehmer waren sich einig, dass das zirkuläre Layout schön aussieht, jedoch wenig hilfreich ist, da das Ordnungsprinzip sehr schwach ist und in nahezu jeder Aufgabe die Ordnung bestimmt werden musste. Einem Teilnehmer sagte das alphabetische Layout am meisten zu, da dieses auch am ehesten dem gewohnten Bild einer Word-Cloud entspricht. Das favorisierte Layout war jedoch das nach Häufigkeit sortierte, was alle Teilnehmer als hilfreich empfanden.

Dem halbstandardisierten Interview war folgendes Feedback zu entnehmen: Einige Teilnehmer lobten die Auswahl der Aufgaben. Wenn Äußerungen zu den Wortarten und Kategorien gemacht wurden, so waren sich die Teilnehmer darin einig, dass die Kategorien wesentlich interessanter für den täglichen Gebrauch seien als die Wortarten. Der Mehrwert des Programms gegenüber statischer Word-Clouds wurde oft erwähnt. Die meisten Teilnehmer

waren positiv überrascht, wie schnell und einfach sie mithilfe des Programms an Informationen herankamen, bei denen sie sonst keinen Ansatz hätten, diese überhaupt herauszufinden zu könnten. Ein Teilnehmer bezeichnete die Word-Cloud als sehr passende Visualisierung, da diese keine Präzision vortäusche, die, aufgrund der Fehlerwahrscheinlichkeit der NLP-Verfahren und Datenbasis, nicht gegeben sei. Insgesamt betrachtet fiel die Bewertung des Programms durch die Teilnehmer sehr positiv aus. So wurde das Programm als „ziemlich cool“, „gut zu verwenden“, „intuitiv bedienbar“ bezeichnet und alle Teilnehmer hielten das Programm für „interessant“.

Folgende Funktionen gefielen den Teilnehmern am besten oder wurden als sehr nützlich bezeichnet: (absteigend nach Anzahl der Nennungen sortiert)

- Suchfunktion
- Filtermöglichkeit
- Vorschaufunktion von Filtern und Beziehungen
- Anzeige der Häufigkeiten
- Lemmatisierung
- Farbliche Hervorhebung der Wortarten
- Anpassbarkeit der Beziehungshervorhebung
- Multiwörter und deren CamelCase-Darstellung
- Alphabetisch sortiertes Layout

5.8. Diskussion

5.8.1. Herangehensweise

Bei der Studie sind grundsätzlich zwei Lösungsstrategien aufgefallen, die sicherlich mit dem jeweiligen Typ der Teilnehmer zusammenhängen. Die eine Herangehensweise war sehr funktionslastig, die Word-Cloud wurde hierbei nur als Anzeige der Funktionsergebnisse angesehen. Die zweite Strategie beruhte eher auf visuellen Eindrücken, weshalb sich die meiste Interaktion in der Word-Cloud abspielte und die Funktionen eher in den Hintergrund rückten und erst dann getestet wurden, wenn das Visuelle (in Verbindung mit Hintergrundwissen) nicht mehr ausreichte. Optimal wäre eine Kombination der beiden Herangehensweisen, was bei einigen Teilnehmern beobachtet werden konnte, jedoch nur bei einzelnen Aufgaben.

5.8.2. Verbesserungsvorschläge

Angepasste Informationen

Eine Funktionalität, die von allen Teilnehmern gewünscht wurde, betrifft die Informationsanzeige. Hierbei handelt es sich um die Problematik, dass Informationen zu einem Wort nicht dynamisch an ausgewählte Filter angepasst werden. Ebenso wenig sind die Informationen von einer getroffenen Auswahl abhängig. Das hat folgenden Hintergrund: Jedes Wort und jede dazugehörige Wortform wird als Tag (siehe Abschnitt 4.3.1) gespeichert. Sämtliche Informationen eines Wortvorkommens werden in solchen Tags vereint. Auf diese Weise kann sehr viel Speicherplatz und Berechnungszeit eingespart werden. Die Informationen sind so konzipiert, dass für ein ausgewähltes Wort das entsprechende Tag gesucht wird und dessen Informationen dargestellt werden. Mit der bestehenden Datenstruktur ist es also nicht möglich, die Wortart eines ausgewählten Wortes (sofern das Tag mehrere Wortarten enthält) in einem bestimmten Satz festzustellen. Analog verhält es sich mit den Kategorien. Eine weitere Folge dieser Datenstruktur ist genau genommen eine verfälschte Darstellung der Informationen. Ist beispielsweise nur der Verb-Filter aktiviert, werden in der WordCloud zwar alle Verben dargestellt, jedoch ist die Größe der einzelnen Wörter möglicherweise nicht korrekt. Dies rührt von der Tatsache her, dass innerhalb eines Tags nicht unterschieden werden kann, welche der Wortformen als Verben annotiert wurden und welche als etwas anderes. Deshalb wird ein Wort, das 20 mal als Verb und 20 mal als Substantiv erkannt wurde, ebenso groß dargestellt wie eines, das lediglich 40 mal als Verb erkannt wurde. Dieses Problem kann vermeintlich trivial gelöst werden, indem nur die 20 Verbvorkommen in die Größenberechnung einfließen, jedoch ist diese Lösung nicht mehr anwendbar, wenn sowohl Wortart- als auch Kategorie-Filter aktiviert sind. In diesem Fall müsste zu jedem Wortvorkommen die Kombination aus Kategorie und Wortart vorliegen, um entscheiden zu können, ob es aufgrund der Wortart mitgezählt wird oder aufgrund der Kategorie und um ausschließen zu können, dass es doppelt mitgezählt wird. Um die Problematik zu verdeutlichen, wird ein Wort betrachtet, welches insgesamt 30 mal vorkommt. 20 mal wird das Wort als Verb erkannt, 10 mal als Substantiv und 25 mal wird das Wort als Datum kategorisiert. Die Filter Verb und Datum seien aktiv, alle anderen Filter inaktiv. Nun kann keine korrekte Aussage über die Anzahl getroffen werden, da keine Verbindung zwischen Wortart und Kategorie besteht. Die Auswahlfunktion steht vor einem ähnlichen Problem. Zu jedem Satz werden die darin enthalten Tags und die jeweilige Anzahl der Vorkommnisse gespeichert und zu jedem Tag eine Liste der Sätze, in denen das Tag auftritt. Wählt der Benutzer nun ein Wort aus, so wird die Satzliste durchgegangen und alle darin vorkommenden Wörter entsprechend ihrer Häufigkeit in der Word-Cloud repräsentiert. In diesem Fall ist die Größendarstellung korrekt, jedoch gibt es keine Möglichkeit, die Wortarten oder Kategorien dieser Vorkommnisse herauszufinden. Die beiden skizzierten Probleme haben die gemeinsame Ursache in der Einfachheit der Datenstruktur. Demzufolge erfordert eine Lösung eine komplexere Datenstruktur, die zu jedem Wortvorkommen dessen Wortart und Kategorie speichert. Auf diese Weise wäre sowohl eine exaktere Filterung einer Auswahl möglich als auch die exakte Größendarstellung des Wortes in der Word-Cloud. Auch wenn die Ergebnisse von solch einer Datenstrukturanpassung profitieren, bleibt die Frage offen,

ob das Programm durch den erhöhten Speicherbedarf und die zwangsläufig gestiegene Rechenzeit noch benutzbar ist. Da die Anpassung der Datenstruktur den Rahmen dieser Arbeit sprengen würde, kann die Benutzbarkeit folglich nicht überprüft werden.

Textviewer

Die Möglichkeit, in den Originaltext zu schauen, ist ein sehr wichtiger Punkt, der von jedem Teilnehmer angesprochen wurde. Da die Word-Cloud nur einen groben Überblick und Zusammenhänge liefern kann, ist es für exakte Analysen unabdingbar, sich anhand des originalen Kontextes zu vergewissern und die durch die Word-Cloud erhaltenen Ergebnisse zu verifizieren. Im Rahmen der verbleibenden Zeit dieser Arbeit soll ein simpler Textviewer in das Programm integriert werden, welcher die relevanten Sätze, die beispielsweise durch eine Auswahl beschränkt wurden, anzeigen soll. Hierfür wäre jedoch auch eine eigenständige Komponente denkbar, die mit diesem Programm zusammenarbeitet und von diesem aus mit den entsprechenden Parametern aufgerufen werden kann. Darüber hinaus könnte

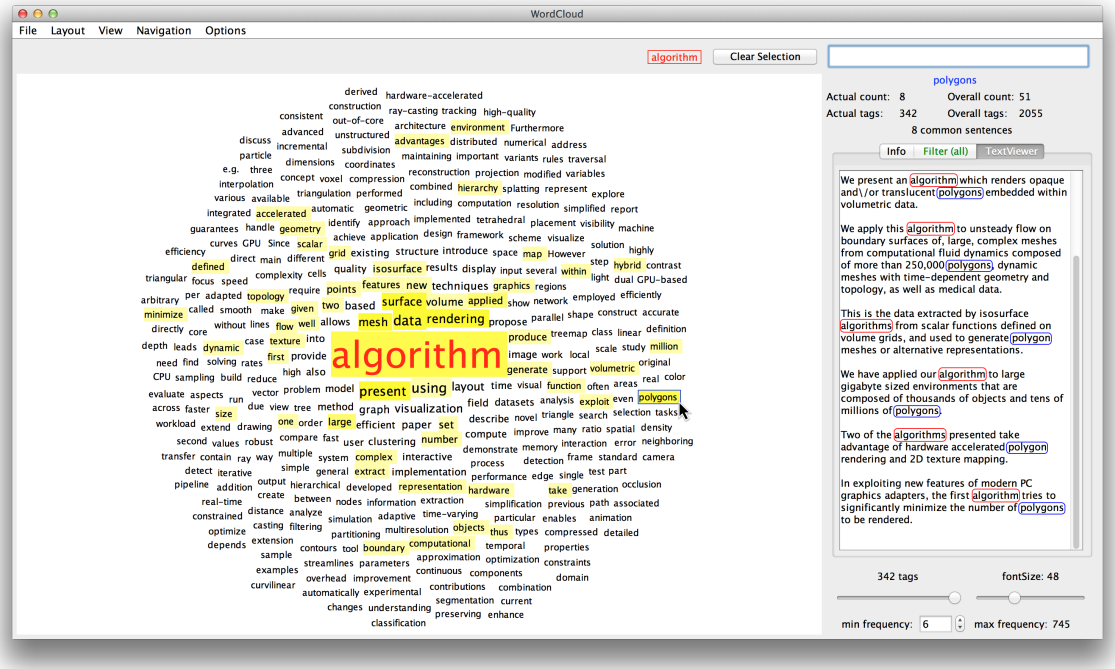


Abbildung 5.1.: Einfacher Textviewer: Die gemeinsamen Sätze des ausgewählten Tags „algorithm“ und des Tags „polygons“ (auf welchem der Mauszeiger steht) werden in dem Textviewer dargestellt. Zur Veranschaulichung werden die Vorkommnisse der ausgewählten Tags farblich hervorgehoben (nicht Bestandteil des Programms).

eine solche „Textview-Komponente“ weiteren Komponenten der visuellen Analyse einen Mehrwert bieten. Ein einfacher Textviewer ist in Abbildung 5.1 zu sehen.

Automatische Wiederauswahl

Viele der Teilnehmer haben sich daran gestört, dass eine getroffene Auswahl von Wörtern verlorengeht, sobald gewisse Einstellungen vorgenommen werden. Diese Einstellungen sind zum einen die Lemmatisierung und zum anderen die Anzeige von Multiwörtern. Der Verlust der ausgewählten Wörter hat folgenden Hintergrund: Grundsätzlich gibt es drei getrennte Mengen von Tags. Die Menge der Tags mit Lemmatisierung, die Menge der Tags ohne Lemmatisierung und die Menge der Multiwörter. Um das Datenmodell möglichst effizient und einfach zu gestalten, liegt der Word-Cloud jeweils eine einzige Tag-Menge zugrunde. Multiwörter bilden hier eine Ausnahme, da sie einer der beiden anderen Mengen hinzugefügt beziehungsweise von dieser entfernt werden können. Dies ist nur möglich, weil Multiwörter keinen Gebrauch von der Lemmatisierung machen. Wird nun zwischen diesen Mengen gewechselt, existieren die ausgewählten Wörter nicht mehr in dieser Form, da ihnen andere Tags zugrunde liegen. Obwohl also ein Wort wie beispielsweise „information“ sowohl in der Menge der Tags mit Lemmatisierung vorkommt als auch in der Menge der Tags ohne Lemmatisierung, sind es zwei unabhängige Tags. Ähnlich verhält es sich bei den Multiwörtern. Wird ein Multiwort ausgewählt und anschließend die Anzeige der Multiwörter deaktiviert, so ist das ausgewählte Multiwort nicht in der Menge der Tags ohne Multiwörter enthalten und kann folglich auch nicht ausgewählt bleiben.

Mit kleinen Einschränkungen wurde dieses Verhalten geändert und eine automatische Wiederauswahl hinzugefügt. In jedem Fall muss die aktuelle Auswahl gelöscht werden, weil die dahinterstehenden Tags nicht länger in der Word-Cloud existieren. Jedoch ist es wie im oben dargestellten Beispiel mit „information“ leicht möglich, das Tag in der neuen Menge von Tags zu suchen und erneut auszuwählen. Es gibt jedoch einige Spezialfälle, bei denen es sich anders verhält. Wird beispielsweise eine beliebige Form eines Verbs (ohne Lemmatisierung) ausgewählt und anschließend die Lemmatisierung aktiviert, so ändert sich die Form des ausgewählten Wortes aufgrund der Tatsache, dass – unabhängig von dem gesuchten Begriff – für alle Wortformen eines Tags dessen häufigste Wortform angezeigt wird. Wird also das unlemmatisierte Wort „controlling“ ausgewählt, so ist nach dem Aktivieren der Lemmatisierung etwa „control“ ausgewählt. Während bei einem Wechsel bezüglich der Lemmatisierung immer ein verwandtes Wort automatisch ausgewählt wird, verhält es sich bei den Multiwörtern anders. Multiwörter lassen sich zwar leicht in ihre Bestandteile zerlegen und einzeln auswählen, jedoch stellt die Gegenrichtung eine ungleich größere Herausforderung dar. Werden n viele Bestandteile von Multiwörtern ausgewählt und anschließend die Multiwörter aktiviert, so gibt es neben den $n!$ vielen Möglichkeiten, die durch unterschiedliche Reihenfolgen entstehen, auch noch diverse Kombinationsmöglichkeiten der einzelnen Bestandteile untereinander. Da dieses Problem nicht durch einen einfachen Algorithmus zu lösen ist und der Aufwand, eine Lösung zu implementierten, den Aufwand, die Multiwörter erneut auszuwählen, bei Weitem übersteigt, wurde dieser Anwendungsfall ignoriert.

Verhalten des Suchfeldes

Einigen Teilnehmern missfiel das Verhalten des Suchfeldes. Wird ein Wort eingegeben und auf Enter gedrückt, so leert sich der Inhalt des Suchfeldes automatisch, um dem Benutzer eine weitere Suche zu ermöglichen. Dieses Verhalten an sich wurde als intuitiv wahrgenommen, sofern das gesuchte Wort gefunden werden konnte und dadurch eine sichtbare Änderung der WordCloud erfolgte. Konnte das gesuchte Wort jedoch nicht gefunden werden, so verschwand die Eingabe ohne Feedback. Zu Recht merkten einige Teilnehmer an, dass sie es für sinnvoll erachten, das Wort nur im Falle einer positiven Übereinstimmung zu entfernen. Im Falle einer negativen Übereinstimmung wurde der Wunsch laut, das nicht gefundene Wort im Suchfeld zu behalten, um eine Korrektur oder Anpassung zu ermöglichen. Da diese Verhaltensänderung einerseits der Intuitivität des Programms dient und andererseits keinen großen programmatischen Aufwand darstellt, wurde sie implementiert.

Autovervollständigung

Ein Teilnehmer vermisste die Funktionalität der Autovervollständigung bei der Suche. Bisher ist ein Anwender dazu gezwungen, stets das gesamte Wort in das Suchfeld einzugeben, um eine positive Übereinstimmung bekommen zu können. Sehr viel komfortabler wäre jedoch die Möglichkeit, nur den Anfang des gesuchten Wortes einzugeben und daraufhin sinnvolle Ergänzungsvorschläge zu erhalten. Abhängig von der Lemmatisierung sollten die Ergänzungsvorschläge gruppiert werden, sofern es sich um mehrere Wortformen eines einzigen Tags handelt, die jeweils das bisher eingegebene Teilwort enthalten. Um dies anhand eines Beispiels zu verdeutlichen, betrachten wir das Wort „control“ mit den Wortformen „control“, „controls“, „controlled“ und „controlling“. Ein Anwender gibt in das Suchfeld den Text „con“ ein. Nun sollte die häufigste Wortform des gesuchten Tags (hier: „control“) hervorgehoben werden, da nur diese in der Word-Cloud zu sehen sein wird, und die anderen Wortformen, die ebenfalls das bisher eingegebene Teilwort enthalten und zu demselben Tag gehören (hier: „controls“, „controlled“ und „controlling“), darunter aufgelistet werden. Weitere Wörter wie „condition“ sollten analog dazu gruppiert werden. Mit fortschreitender Eingabe muss die Liste der Vorschläge entsprechend verkürzt werden. Die häufigste Wortform eines gesuchten Tags, von dem mindestens eine Wortform den Suchtext enthält, sollte jedoch ständig in der Liste präsent sein, unabhängig von möglichen Übereinstimmungen. Wird in diesem Beispiel also „controls“ eingetippt, sollte die Liste der Vorschläge nun das hervorgehobene „**control**“ und darunter „controls“ enthalten. Da dieser Ansatz jedoch lediglich die Komfortabilität steigern würde und einen erheblichen Implementierungsaufwand darstellt, musste aus Zeitgründen darauf verzichtet werden.

Auswahlverhalten der Sucheingabe

Um ein Wort zur Auswahl hinzuzufügen oder es daraus zu entfernen, gibt es diverse Möglichkeiten. Wird ein Wort beispielsweise angeklickt, so wird es entweder zur Auswahl hinzugefügt oder daraus entfernt, abhängig davon, ob dieses Wort bereits Teil der Auswahl ist

oder nicht. Eben dieses Verhalten wurde auch für die Suche verwendet. Wurde ein gesuchtes Wort in der aktuellen Menge der Tags gefunden (und die Enter-Taste betätigt), so simulierte das Programm einen Mausklick auf das Label mit dem entsprechenden Wort. Dies hatte zur Folge, dass eine zweimalige Suche nach demselben Wort (jeweils mit Bestätigung durch die Enter-Taste) das Wort zur Auswahl hinzufügte und wieder daraus entfernte oder vice versa. Vielen Teilnehmer kam dieses Umschaltverhalten in Verbindung mit einem Mausklick intuitiv vor, jedoch nicht in Verbindung mit der Suchfunktion. Infolge dessen wurde das Verhalten der Suche dahingehend geändert, dass Wörter der Auswahl hinzugefügt werden können, jedoch durch ein zweites Hinzufügen nicht aus der Auswahl verschwinden.

Speicherung von Wörtern

In einigen der Aufgaben wurden bestimmte Wörter mehrmals verwendet oder gesucht. Einer der Teilnehmer fragte daraufhin nach einer Möglichkeit, Wörter in irgendeiner Form speichern zu können, um schnelleren Zugriff auf diese zu erhalten. Zwar besteht die Möglichkeit, mehrere Wörter per Auswahl festzuhalten, jedoch verändert sich dadurch zwangsläufig auch die Word-Cloud und macht diese für die eigentliche Aufgabe unbrauchbar. Da es bei dem vorgesehenen Gebrauch der Word-Cloud jedoch keine vorgegebenen Wörter gibt, deren Schnellzugriffsmöglichkeit einen Mehrwert bieten könnte, wurde diese Idee verworfen.

Informationsanzeige für ausgewählte Wörter

Die Anzeige der ausgewählten Wörter oberhalb der Word-Cloud stieß bei den Teilnehmern auf positive Resonanz. Auch die Funktionalität, diese Wörter per Mausklick aus der Auswahl entfernen zu können, wurde häufig verwendet. Darüber hinaus entstand der Wunsch, ähnlich wie bei den Wörtern der Word-Cloud Informationen zum jeweiligen Wort angezeigt zu bekommen, sobald der Mauszeiger auf das Wort gefahren wird. Dieser Wunsch hat folgenden Hintergrund: Sind alle Filter aktiviert, so sind die ausgewählten Wörter mit sehr hoher Wahrscheinlichkeit in der Word-Cloud zu finden. Jedoch gibt es auch Situationen, in denen die ausgewählten Wörter nicht in der Word-Cloud auftauchen sollen. Eine solche Situationen wurde beispielsweise durch Aufgabe 2a (VisWeekAbstracts) gezielt provoziert. In dieser Aufgabe wurde von den Teilnehmern verlangt, das Wort „cells“ auszuwählen und anschließend nach Personen zu filtern. Obwohl eine Person gesucht wurde, ließen sich einige Teilnehmer davon irritieren, dass das ausgewählte Wort „cells“ nicht in der Word-Cloud zu finden war (da keine Person). Um dennoch leicht an Informationen zu dem ausgewählten Wort zu gelangen, ohne jedoch gezwungen zu sein, sich alle momentan uninteressanten Kategorien anzeigen zu lassen, wurde die oben beschriebene Funktionalität hinzugefügt. Sie unterscheidet sich von dem Verhalten der anderen Wörter der Word-Cloud lediglich durch die fehlende Beziehungsvorschau, da alle Wörter der aktuellen Word-Cloud ohnehin in Beziehung zu dem ausgewählten Wort stehen und eine Vorschau aller Wörter der Word-Cloud den Informationsgehalt nicht erhöht.

Anzeige der Worthäufigkeit in der Word-Cloud

Die räumliche Unterteilung des Programms in die grobe Visualisierung auf der linken Seite und exakte Informationen hierzu auf der rechten Seite sagte den meisten Teilnehmern zu. Nahezu alle Aufgaben verlangten von den Teilnehmern einen ständigen Wechsel zwischen den beiden Bereichen. Einem Teilnehmer war es demzufolge ein großes Anliegen, einen Teil der exakten Informationen in die Word-Cloud zu verlagern, damit der Blick nicht ständig von links nach rechts und zurück wechseln muss. Da eine permanente Anzeige der Wortinformationen (wie beispielsweise die Häufigkeit eines Wortes) die Word-Cloud unnötig aufblähen würde und auch die Übersichtlichkeit darunter zu leiden hätte, ist die Idee, nur das aktuell hervorgehobene Wort mit Informationen zu versehen. Die Visualisierung dieser Informationen könnte in Form eines kleinen Pop-ups erfolgen oder auch durch die Veränderung des Wortes selbst. So wäre es denkbar, in der Word-Cloud das Wort „user“ anzuzeigen und sobald die Maus darauf gefahren wird, das Wort beispielsweise in „user (count: 849)“ zu ändern. Da anzunehmen ist, dass dieses Verhalten nicht von allen Benutzern erwünscht ist, sollte es optional sein. Aus zeitlichen Gründen kann diese Ergänzung im Rahmen der vorliegenden Arbeit nicht mehr umgesetzt werden.

Kontextmenüs und Tooltips

Die meisten Teilnehmer bezeichneten das Programm als „intuitiv“ und „gut zu bedienen“. Dennoch wünschten sich einige zusätzliche Hilfestellungen wie Tooltips oder Kontextmenüs. Der Grund für diesen Wunsch war fehlende Routine und eine daraus resultierende Unsicherheit, mit welchen Aktionen welches Ergebnis erzielt werden kann. Diese Hilfestellungen sind also nicht für den täglichen Gebrauch gedacht, sondern um anfänglich leichter mit dem Programm vertraut zu werden. Die beiden Punkte werden gemeinsam diskutiert, da sie im Bezug auf dieses Programm lediglich zwei Lösungswege für ein einziges Problem darstellen. Um neuen Anwendern des Programms den Einstieg zu erleichtern, fiel die Wahl auf Tooltips. Kontextmenüs haben gegenüber den Tooltips einige Vorteile, die in diesem Programm jedoch nicht ausgespielt werden, da es sich nur um Hilfestellungen handeln soll und nicht um zusätzliche Funktionaliäten, die mithilfe von Kontextmenüs angeboten werden könnten. Bei Kontextmenüs muss der Anwender erst auf die Idee kommen, dass es Kontextmenüs geben könnte, wohingegen die Tooltips von selbst in Erscheinung treten und für die reine Informationsvermittlung das passendere Mittel der Wahl zu sein scheinen.

Unterstützung weiterer Sprachen

Manche Nutzer stellten die Frage, ob auch nicht-englische Texte verarbeitet werden können. Das Programm an sich hat diesbezüglich keinerlei Einschränkungen. Jedoch muss die Vorverarbeitung und Annotation des Textes für die entsprechende Sprache angepasst werden. In diesem Programm kommt hierfür das Framework „Stanford CoreNLP“ zum Einsatz und es ist ausschließlich von diesem abhängig, ob die betreffende Sprache unterstützt wird. Da sich die Aufgabenstellung dieser Arbeit auf englische Texte beschränkt, wurde das Programm

ausschließlich dafür konzipiert. Eine Erweiterung ist jedoch grundsätzlich denkbar; für weitere Informationen sei an dieser Stelle auf die Seite des Frameworks [cor] verwiesen.

Farbliche Hervorhebung der Kategorien

Die Meinungen der Teilnehmer bezüglich der farblichen Hervorhebung der Wortarten gingen weit auseinander. Einige fanden sie sehr hilfreich und nützlich, andere sahen darin keinerlei Vorteil. In einem Punkt waren sich jedoch alle einig: Wenn die Wortarten farblich hervorgehoben werden können, ist es nur konsequent die Kategorien ebenfalls hervorzuheben. Der Grund dafür, den Kategorien keine farbliche Hervorhebung zuzuteilen, war vor allem der, dass den meisten Wörtern keine Kategorie zugeordnet werden kann. Selbst wenn ein Wort einer Kategorie zugeordnet wird, kommt es häufig vor, dass dem Wort ebenfalls die Kategorie „OTHER“ (schwarz) zugeordnet wird, weswegen die wenigsten Wörter farbig erscheinen würden. Als Beispiel soll das Wort „future“ dienen. Dieses Wort kommt in den VisWeekAbstracts 33 mal vor und wird davon 6 mal als Datum und 27 mal als „OTHER“ kategorisiert. Folglich würde ihm die Farbe der Kategorie „OTHER“ (also schwarz) zugewiesen werden. Würde für die Farbverteilung die Kategorie „OTHER“ außer Acht gelassen, erhielte auch das Wort „future“ eine Farbe, zusammen mit allen anderen Wörtern, die mindestens eine andere Kategorie als „OTHER“ besitzen. Doch selbst mit diesem Ansatz blieben immer noch viele Wörter farblos, was auf Abbildung 5.2 verdeutlicht werden soll. Zu sehen sind die beiden Textcorpora, die den Aufgabenblättern zugrunde liegen (siehe 5.2), mit zwei unterschiedlichen Hervorhebungsstufen. Ist ein Wort mit sattem Gelb hinterlegt, so bekäme es in jedem Fall eine Farbe. Ist ein Wort hingegen mit blassem Gelb hinterlegt, so würde

Abbildung 5.2.: Hervorhebung kategorisierter Wörter

es nach der bisherigen Farbverteilung schwarz bleiben, mit dem neuen Ansatz bekäme es eine Farbe. In jedem Fall bliebe die Mehrzahl der Wörter farblos. Mit dem neuen Ansatz wäre darüber hinaus das Konzept der Farbgebung nicht länger konsequent umgesetzt. Als weiteren Kritikpunkt kann ein Verwirrungspotential bezüglich der unterschiedlichen Bedeutungen der Farben angeführt werden. Ein Abwägen der potentiellen Vor- und Nachteile hatte den Beschluss zur Folge, dieses Feature nicht zu implementieren.

Individuelle Kategorien

Abhängig von Textkorpus und dessen Fachbereich können einige Kategorien nützlich sein, während es auch Kategorien geben kann, die in diesem Zusammenhang irrelevant sind. Infolge dessen erkundigte sich ein Teilnehmer nach der Möglichkeit, die Kategorien individuell anpassen zu können. Grundsätzlich sind die Kategorien, die in dem Programm angezeigt werden, mit geringem Programmieraufwand änderbar, wenn auch nicht direkt für den Benutzer. Hinter den sichtbaren Kategorien steckt eine zweiwertige Liste, welche die durch das CoreNLP-Framework annotierten Kategorien den im Programm sichtbaren Kategorien zuordnet. Jedoch sind die von dem CoreNLP-Framework verwendeten Kategorien nicht für Änderungen vorgesehen. Da die Verbesserung der in dieser Arbeit verwendeten linguistischen Tools nicht Bestandteil der Arbeit ist, sind die Kategorien des CoreNLP-Frameworks nicht änderbar.

Geteilter Bildschirm

In Zusammenhang mit Aufgabe 1 (VisWeekAbstracts) kam der Wunsch auf, einen geteilten Bildschirm zur Verfügung zu stellen. In dieser Aufgabe ging es um einen Vergleich zwischen der Word-Cloud mit aktivierter und deaktivierter Lemmatisierung. Ein Teilnehmer hatte die Idee, dass ein geteilter Bildschirm hier einen Mehrwert bieten könnte, da der Anwender sonst gezwungen ist, hin- und herzuschalten und sich die Ergebnisse zu merken. Das gleiche Prinzip gilt für weitere Einstellungen wie beispielsweise die Multiwörter. Obwohl dieses Feature einen Mehrwert für vergleichende Aufgaben darstellen könnte, ist die Implementierung äußert fragwürdig. Hat ein Anwender den Wunsch, mehrere Word-Clouds, die sich in jeweils einer Einstellung unterscheiden, zu vergleichen und sie gleichzeitig auf dem Bildschirm sehen zu können, so sei ihm empfohlen, das Programm mehrfach zu öffnen. Um nicht unnötig Zeit zu verlieren, sollte die Word-Cloud samt verarbeitetem Text gespeichert und anschließend entsprechend oft neu gestartet werden. Für jede neue Instanz des Programms muss nun lediglich der binär gespeichert Text geladen und die entsprechende Einstellung angepasst werden. Auf diese Weise kommt der Anwender zu einem geteilten Bildschirm mit maximaler Flexibilität, was die Größe und Anzahl der einzelnen Word-Clouds angeht.

Zusätzliche Filterauswahlmöglichkeiten

Insgesamt stehen dem Benutzer 17 Filter zur Verfügung. Des Öfteren ist es hilfreich, alle Filter zu aktivieren oder auch nur einen einzelnen. Um für den Wechsel dieser beiden Zustände nicht auf 16 Filter klicken zu müssen, stehen zwei Auswahlmöglichkeiten bereit. Eine Auswahlmöglichkeit („show all“) aktiviert, die andere („hide all“) deaktiviert alle Filter auf einmal. Mit Zuhilfenahme dieser beiden Auswahlmöglichkeiten kann zwischen den beiden beschriebenen Zuständen mit einem beziehungsweise zwei Mausklicks gewechselt werden, abhängig von der Richtung. Doch auch wenn nur ein einziger Filter aktiv ist und ein anderer allein aktiviert werden soll, sind zwei Mausklicks vonnöten. Diese beiden Szenarien waren in den Aufgaben sehr häufig gegeben. Einige Teilnehmer stellten deshalb die Frage, ob eine Möglichkeit existiert, einen Filter exklusiv auszuwählen (und alle anderen zu deaktivieren). Tatsächlich besteht diese Möglichkeit, indem mit der rechten Maustaste auf einen Filter geklickt wird. Da die rechte Maustaste jedoch wenig intuitiv für diese Funktion ist, wurde dieses Feature bei der Einführung nicht vorgestellt. Eine andere Möglichkeit, die exklusive Auswahl eines Filters mit nur einem Mausklick zu bewerkstelligen, besteht in der Zuhilfenahme einer bestimmten Taste. Da diese Variante umständlich und nicht wesentlich intuitiver ist, fiel die Wahl auf die bisher unbelegte rechte Maustaste. Trotz des nun vorhandenen Wissens der Teilnehmer um dieses Feature kam es nur unmittelbar nach der Erklärung zum Einsatz, kurze Zeit später wurde wieder die Variante mit zwei Mausklicks gewählt. Dieses Verhalten bestätigte zwar die Vermutung bezüglich der mangelnden Intuitivität der rechten Maustaste für dieses Feature, zeigte jedoch auch, dass es eher eine untergeordnete Rolle spielt und die Nutzer mit dem zusätzlichen Mausklick zurechtkamen.

Bezüglich der Filterauswahl fiel im Verlauf der Studie ein weiteres Szenario auf, für welches ein Teilnehmer einen Optimierungswunsch äußerte. Grundsätzlich sind die Filter in die beiden Gruppen Wortart und Kategorie aufgeteilt, wobei jede Gruppe die gesamte Wortmenge abdeckt. Sind beispielsweise alle einzelnen Filter der Wortarten aktiviert, so zeigt die Word-Cloud alle Wörter an. Jede Veränderung einer Filterauswahl in der Kategoriegruppe wird somit wirkungslos, da der Wortmenge nichts mehr hinzugefügt werden kann, das nicht bereits enthalten ist. Unter Berücksichtigung dieses Wissens wählten einige Teilnehmer stets eine der beiden Gruppen komplett ab. Da für dieses Szenario in jedem Fall mindestens acht Mausklicks nötig sind, kam der Wunsch auf, die oben vorgestellten Auswahlmöglichkeiten („hide all“ und „show all“), welche für alle Filter gelten, zusätzlich für jede Gruppe anzubieten, sodass mit einem Mausklick eine komplette Gruppe aktiviert und deaktiviert werden kann. Jedoch müssten auch nachdem eine komplette Gruppe aktiviert und die andere deaktiviert ist, Bestandteile der aktivierten Gruppe deaktiviert werden, um eine Filterung zu ermöglichen. Die Verwendung dieser Funktionen würde sich also erst dann lohnen, wenn mehr als die Hälfte aller Bestandteile einer Gruppe aktiviert werden sollen, was jedoch in keiner der Aufgaben hilfreich gewesen wäre. Da diese vier zusätzlichen Auswahlmöglichkeiten einerseits für Verwirrung sorgen könnten, da der Überblick unter ihnen leiden würde, und andererseits kein ersichtlicher Mehrwert gewonnen werden kann, wurden diese zusätzlichen Auswahlmöglichkeiten verworfen.

Reset-Button

Da die Aufgaben meist unterschiedlicher Natur waren und entsprechende Einstellungen angepasst werden mussten, schlug ein Teilnehmer vor, einen Reset-Button hinzuzufügen, der nach jeder vollendeten Aufgabe betätigt werden kann. Dieser Button sollte die Funktionalität besitzen, alle bisher ausgewählten Wörter aus der Auswahl zu entfernen, da diese oft vernachlässigt wurde. Für die meisten Aufgaben war anfänglich die gesamte Word-Cloud zu betrachten, deshalb wäre die Aktivierung aller Filter eine weitere hilfreiche Funktionalität des Reset-Buttons. Was Einstellungen wie Lemmatisierung, Multiwörter und Ähnliches angeht, fällt es zunehmend schwerer, eine sinnvolle Vorbelegung zu treffen, da dies sehr von den Aufgaben abhängig ist. Infolge dessen muss der Reset-Button entweder konfigurierbar sein oder sich auf die beiden erstgenannten Funktionalitäten beschränken. Da der Einsatz solch eines Buttons überwiegend bei Aufgaben Sinn macht und weniger im täglichen Gebrauch, würde eine Konfiguration des Buttons einen wenig lohnenswerten Aufwand darstellen. Sowohl für die Zurücksetzung der Auswahl als auch die Aktivierung aller Filter existiert jeweils ein entsprechender Button. Da oft nur eine der beiden Funktionalitäten benötigt wird und die unter einem zusätzlichen Button leidende Übersichtlichkeit den minimalen Mehrwert nicht rechtfertigt, wurde dieser Reset-Button nicht implementiert.

Benutzeroberfläche

Bezüglich der Benutzeroberfläche wurden einige Verbesserungen vorgeschlagen, die allesamt umgesetzt wurden. Es handelte sich dabei unter anderem um Benennungen von Labels. Wurden beispielsweise mehrere Wörter ausgewählt, die keinen Satz gemeinsam hatten, zeigte die Word-Cloud das Label „Too many selections!“ an. Hier wurde zurecht angemerkt, dass der Wortlaut nicht zwingend zutreffend ist, da nicht die Anzahl der ausgewählten Wörter Grund für die Anzeige sein muss, sondern diese ebenso durch die Tatsache, dass mindestens zwei Wörter nicht in den gemeinsamen Sätzen vorkommen, hervorgerufen werden kann. Demzufolge wurde der Wortlaut in „No common sentence found!“ geändert. Außerdem wurde das Wort „co-occurrence“ durch „relation“ ersetzt, da es den meisten Anwendern auf Anhieb mehr sagt. Für jedes Wort gibt es einen absoluten Zähler, der sich auf die Vorkommen im gesamten Text bezieht, und einen relativen Zähler, der die Häufigkeit im Zusammenhang mit der Auswahl und den aktivierten Filtern repräsentiert. Viele Teilnehmer wussten mit der Darstellung der Zähler, beispielsweise $3(8)$, nichts anzufangen und fragten mehrmals nach, deshalb steht nun vor jedem Zähler seine jeweilige Bedeutung. Dass, bei Übereinstimmung von relativem und absolutem Zähler, nur eine Zahl angezeigt wurde, irritierte einige Teilnehmer, weshalb nun ständig beide Zähler angezeigt werden. Ähnlich ging es einigen mit der Anzeige der gemeinsamen Sätze, die nur auftauchte, sobald eine Auswahl getroffen wurde und die Maus auf ein Wort gefahren wurde. Auch diese Anzeige wird nun ständig dargestellt. Viele Teilnehmer wünschten sich eine Gruppierung der Zähleranzeigen, da diese bislang teilweise räumlich getrennt dargestellt wurden. Da nichts gegen diesen Wunsch sprach, wurde er dankend umgesetzt.

6. Diskussion und Ausblick

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst und diskutiert. Darüber hinaus werden Anknüpfungspunkte und Erweiterungsmöglichkeiten dargestellt.

6.1. Zusammenfassung

Im Rahmen dieser Diplomarbeit wurde ein interaktiver Ansatz für die visuelle Analyse von Textdokumenten entwickelt. Die verwendete Visualisierungstechnik baut auf dem Prinzip der Word-Cloud auf und bietet neben unterschiedlichen Layouts interaktive Funktionalitäten, welche die visuelle Analyse von Textdokumenten unterstützen. Die dafür notwendige Verarbeitung von Textkorpora basiert auf dem CoreNLP-Framework der Universität Stanford [cor] und wurde um einige nützliche Funktionalitäten erweitert. Die Konzeption der Funktionalitäten entstammt teilweise der Inspiration durch verwandte Arbeiten, einige Funktionalitäten wurden dagegen erst im Verlauf der Arbeit entwickelt. Durch die Optimierung der Abläufe und Datenstrukturen bietet dieses Programm ein gutes Nutzererlebnis, was die Ergebnisse der qualitativen Nutzerstudie bestätigten. Durch die Evaluation konnten darüber hinaus einige Verbesserungsvorschläge gesammelt werden, welche größtenteils in die Arbeit einflossen. Erwähnenswert aus den Ergebnissen der Studie ist, dass alle Teilnehmer den Einsatz dieses Programms für bestimmte Aufgaben in der Visualisierung für sinnvoll erachten und ihr Interesse daran bekundeten. Als Einsatzbereich konnten sich die Teilnehmer besonders die Suche in Literaturarchiven sowie das Herausfinden und Darstellen von Abhängigkeiten vorstellen. Da der Ansatz weiterhin Verbesserungspotential besitzt, werden nach dessen Diskussion Anknüpfungspunkte vorgestellt.

6.2. Diskussion

Die Konzeption als interaktive Word-Cloud ließ erfreulich wenige Wünsche offen, da viele Vorteile der verwandten Arbeiten und Ansätze vereint werden konnten. Auf diese Weise konnten beispielsweise mehrere Layouts angeboten werden, von denen der Anwender das jeweils passende auswählen kann und nicht dazu gezwungen ist, mit den Nachteilen eines einzelnen Layouts auskommen zu müssen. Die Interaktivität ermöglichte zudem, sämtliche Funktionalitäten und Einstellungen optional zu gestalten und somit eine Überladung der Word-Cloud zu vermeiden.

Auch die Integration herkömmlicher Analysemethoden wie beispielsweise einer Suchfunktion konnte das Konzept aufwerten und machte einen Vergleich zwischen Word-Cloud und Suchfunktion hinfällig, da der Anwender stets die bevorzugte Methode verwenden kann, ohne auf die andere verzichten zu müssen. Die Nutzerstudie bestätigte den Erfolg dieser Symbiose.

Die eigenen Ideen, wie beispielsweise filter- und colorierbare Wortarten anzubieten, erwies sich bei einigen Aufgaben als überaus nützlich und stieß bei den Teilnehmern der Nutzerstudie auf positive Resonanz.

Eine Evaluation des Ansatzes durchzuführen, erwies sich als sehr hilfreich, um nützliches Feedback, Verbesserungsvorschläge und neue Ideen zu erhalten. Darüber hinaus konnten die Experten auf dem Gebiet der Visualisierung bestätigen, dass das Programm intuitiv bedienbar ist, und halten den Einsatz des Programms für bestimmte Aufgaben der visuellen Analyse für sinnvoll.

Dass während der Nutzerstudie keine weiteren Einstellungen des Programms vermisst wurden, lässt auf eine gute Abwägung der ausgelagerten Einstellungen schließen (siehe Abschnitte 4.3.4 und 4.3.5).

Besonders anhand der Ergebnisse der Nutzerstudie zeigte sich der Nutzen und die Effektivität der Vorschaufunktionen von Filtern und Kookkurrenzen. Obwohl die durch die Vorschaufunktionen hervorgehobenen Wörter nicht die vollständige Menge der Funktionsergebnisse abdecken, genügten diese oftmals für das Lösen der Aufgaben, wodurch viel Zeit gespart werden konnte. Besonders im Hinblick auf größere Textkorpora bieten diese Vorschaufunktionen enorme Vorteile, da sie lediglich von der Anzahl der angezeigten Tags (nicht von der Gesamtanzahl) abhängen und demzufolge nicht von einer Skalierung beeinflusst werden.

Das Persistieren von verarbeiteten Textkorpora und getroffenen Einstellungen (siehe Abschnitt 3.11) war besonders für die Durchführung der Nutzerstudie von großem Vorteil, um gleiche Ausgangsbedingungen zu garantieren und mehrmalige Verarbeitung der Textkorpora zu vermeiden.

Obwohl während der Konzeption und Implementierung stets darauf geachtet wurde, die Verarbeitung sehr umfangreicher Textkorpora zu unterstützen, konnte diese Zielsetzung nur bedingt umgesetzt werden. Die Limitierungen des Programms sind in Abschnitt 4.5.6 veranschaulicht und werden im Ausblick aufgegriffen. Betroffen von einer Skalierung des Textumfangs sind neben den benötigten Ressourcen die Rechenzeiten, die benötigt werden, um Layouts und Kookkurrenzen zu berechnen sowie Filterungen durchzuführen.

Der für das CoreNLP-Framework benötigte Arbeitsspeicher erschwert eine Verwendung dieses Ansatzes als Applet oder Komponente erheblich, weswegen das Konzept dieser Arbeit von einer als Komponente konzipierten Visualisierung zu einem eigenständigen Programm angepasst wurde.

6.3. Ausblick

Da der Ansatz in unterschiedlicher Hinsicht Verbesserungspotential besitzt, werden in diesem Abschnitt Anknüpfungspunkte und Weiterentwicklungsmöglichkeiten vorgestellt. Einige dieser Punkte wurden in der Arbeit bereits diskutiert, sollen in diesem Abschnitt jedoch nochmals erwähnt werden, um einen kompletten Überblick zu gewährleisten.

Multilingualität

Das Konzept beschränkt sich bisher auf die Verarbeitung von englischsprachigen Texten. Jedoch wurde während der Implementierung stets darauf geachtet, die Unterstützung weiterer Sprachen zu ermöglichen. Da dieser Ansatz auf dem CoreNLP-Framework [cor] aufbaut, ist auch in diesem eine Unterstützung weiterer Sprachen notwendig. Hierzu existieren bereits Erweiterungen, die beispielsweise eine Erkennung von Kategorien für die deutsche Sprache ermöglichen [FP10], welche lediglich integriert werden müssten.

Erweiterung der Layoutpalette

Da der zeitliche Rahmen dieser Arbeit eine Implementierung des gruppierten Layouts („clustered layout“) verhinderte, hierdurch jedoch ein Mehrwert für den Benutzer geboten werden könnte, sollte dieses Layout implementiert werden. Besonders hinsichtlich thematischer Zusammenhänge konnte dieses Layout überzeugen [LZT09] und könnte den hier dargestellten Ansatz dahingehend erweitern und aufwerten.

Erweiterung der Datenstruktur

Die Einfachheit der Datenstruktur macht eine Berechnung der gewünschten Informationen in Echtzeit sowie eine flüssige Bedienung des Programms möglich (sofern der Textkorpus keine extremen Ausmaße annimmt). Wie in Abschnitt 5.8.2 unter „Angepasste Informationen“ bereits erwähnt, bringt diese Vereinfachung jedoch auch Ungenauigkeiten mit sich, welche für einen groben Überblick über den Textkorpus keine große Bedeutung haben, durch eine komplexere Datenstruktur jedoch vermieden werden können. Diese Datenstruktur hätte neben einem erhöhten Speicherbedarf eine gestiegene Rechenzeit zur Folge, was einen negativen Einfluss auf die Bedienbarkeit mit sich bringen könnte. Da diese komplexe Datenstruktur für die visuelle Analyse jedoch zweifellos einen Mehrwert bieten kann, ist eine Implementierung und Erprobung sinnvoll. Abhängig von der Bedienbarkeit sollte die komplexe Datenstruktur optional konzipiert werden und mit der bisherigen koexistieren, damit sowohl eine flüssige Bedienbarkeit gewährleistet ist als auch für exakte Analyseaufgaben auf die komplexe Datenstruktur zurückgegriffen werden kann.

Erweiterung der Suchfunktion

Wie in Abschnitt 5.8.2 unter „Autovervollständigung“ ausführlich diskutiert wurde, könnte das Suchfeld um Funktionalitäten hinsichtlich einer Autovervollständigung oder einer Anzeige sinnvoller Ergänzungsvorschläge erweitert werden.

Textview-Komponente

Unabhängig von dem potentiell integrierten Textviewer könnte eine Textview-Komponente entwickelt werden, die mit Informationen aus diesem Programm aufgerufen werden kann und eigene Funktionalitäten mit sich bringt. Neben den relevanten Sätzen (siehe Abschnitt 5.8.2 unter „Textviewer“) sollte der Textviewer in der Lage sein, den kompletten Textkorpus anzuzeigen und dabei die relevanten Sätze oder/und Wörter hervorzuheben. In dem Textviewer sollten ebenfalls die üblichen Funktionalitäten wie beispielsweise die Einstellung der Schriftart und -größe integriert sein. Darüber hinaus könnte eine solche Textview-Komponente weiteren Programmen oder Komponenten der visuellen Analyse einen Mehrwert bieten.

Erweiterung der Word-Cloud-Anzeige

Im Rahmen der Evaluation wurde eine Erweiterung der Word-Cloud-Anzeige gewünscht, um die Worthäufigkeiten im Word-Cloud-Bereich ablesen zu können (siehe Abschnitt 5.8.2 unter „Anzeige der Worthäufigkeit in der Word-Cloud“). Auch wenn diese Erweiterung keine neue Funktionalität mit sich bringt, könnte sie dennoch implementiert werden, um den Bedienkomfort zu erhöhen. Die Realisierung dieser Informationserweiterung könnte sich an die Interaktionsmöglichkeit der in Abschnitt 2.2.4 vorgestellten Arbeit (Tagul) anlehnen und neben der Worthäufigkeit weitere Informationen beinhalten. Jedoch sollten diese Erweiterungen der Word-Cloud optional gestaltet werden, da sie bei einigen Analyseaufgaben durchaus hinderlich sein können.

Leistungssteigerungen

Wie in Abschnitt 4.5.6 ersichtlich wird, ist dieser Ansatz hinsichtlich des Umfangs des Textkorpus nicht beliebig skalierbar. Tabelle 4.1 ist beispielsweise zu entnehmen, dass die Verarbeitung eines Textes mit 17307 KB (2814432 Wörter) beinahe fünf Stunden (auf dem getesteten Rechner) in Anspruch nimmt. Verantwortlich dafür sind hauptsächlich die Verarbeitungen durch das CoreNLP-Framework sowie deren Korrekturen (siehe Abschnitt 4.5.2). Einerseits kommen hier Algorithmen mit quadratischer Laufzeit zum Einsatz, andererseits wird für die Verarbeitung nur ein einziger Prozessorkern verwendet. Um größere Textkorpora verarbeiten zu können, liegt die Überlegung nahe, ein schnelleres Framework zu verwenden, das in seinen Funktionen etwas beschränkter ist (beispielsweise Stemming anstelle von

Lemmatisierung). Darüber hinaus könnte die Unterstützung mehrerer Prozessorkerne eine erhebliche Beschleunigung mit sich bringen.

Da die Verarbeitung solch extrem umfangreicher Textkorpora nicht für jeden Anwender eine Rolle spielt, sollte das CoreNLP-Framework optional ausgewählt werden, damit die Entscheidung zwischen Genauigkeit und Schnelligkeit dem Anwender selbst obliegt.

A. Anhang

A.1. Die wichtigsten Datenstrukturen

Listing A.1 Die Klasse „Tag“

```
public String word;
public String frequentWordform;
public int absoluteCount;
public int count;
public int fontSize;
public Rectangle rectangle;
public boolean isStopWord;
public boolean selected;
public boolean multiword;
public HashMap<String, Integer> tagList;
public HashMap<Integer, Integer> posList;
public HashMap<Integer, Integer> neList;
public HashMap<String, Integer> multiwords;
public HashSet<Integer> sentenceList;
```

Listing A.2 Die Klasse „SortedLabels“

```
public String text;
public HashMap<Integer, Sentence> sentences;
private TreeSet<Tag> lemmaTags;
private TreeSet<Tag> lemmaLessTags;
private TreeSet<Tag> workingSet;
public int min;
public int max;
private Integer[] frequencies;
public SpinnerListModel spinnerModel;
public TreeSet<Tag> selectedLabels;
private HashSet<String> visLabels;
public HashMap<String, Integer> coOccurrenceTags;
private Tag coOccTag;
public boolean noNe;
public int minimum;
```

Listing A.3 Die Klasse „Config“ (Interne Konfiguration)

```
public String config_FileName = "internal_config.data";
public String sortedLabels_FileName = "lastTextFile.data";
public String activatorForDroppedLastTextFiles = "Restore file:\n";
public boolean orderByFrequency = false;
public boolean circularLayout = false;
public Color colorDefault = Color.black;
public Color colorHighlight = Color.blue;
public Color colorSelected = Color.red;
public Color colorPreview = Color.blue;
public Color colorCoOcc = Color.yellow;
public String fontName = "Lucida Grande";
public int gapX = 5;
public int gapY = 0; //3
public int fontMax = 48;
public double proportion = 0.75;
public boolean logarithmic = false;
public int CoOccAlphaMin = 32;
public int threshold = 1;
public int minCoOcc = 1;
public int minWordLength = 3;
public int maxMultiwords = 15;
public boolean getMultiwords = true;
public boolean autofill = true;
public boolean hideStopwords = true;
public boolean colorizePOS = false;
public boolean jumpToSearchLabel = false;
public boolean enableStopwordEditing = false;
public HashSet<Integer> posFilter = new HashSet<Integer>();
public HashSet<Integer> neFilter = new HashSet<Integer>();
public boolean[] posValues = new boolean[]{true, true, true, true, true, true,
    true, true, true};
public boolean[] neValues= new boolean[]{true, true, true, true, true, true, true,
    true};
public String[] posCategories = new String[]{"Noun", "Verb", "Adjective", "Adverb",
    "Preposition", "Cardinal", "Pronoun", "OTHER", "Interjection"};
public String[] neCategories = new String[]{"Person", "Location", "Organization",
    "Date", "Time", "Money", "Numeric", "OTHER"};
public boolean useLemma = true;
public boolean highlightCoOccurrences = true;
public boolean multiwords = true;
public boolean exactCalc = false;
public HashMap<String, Integer> posMap = new HashMap<String, Integer>();
public HashMap<String, Integer> neMap = new HashMap<String, Integer>();
private String emergencyFilter = "No filter selected!";
private String emergencySelection = "No common sentence found!";
private String emergencyElse = "No tags to display!";
public boolean startInFullScreen = true;
```

A.2. Evaluation

Auf Abbildung A.1 ist der Ishiharatest für die Feststellung von Farbfehlsichtigkeiten zu sehen. Darauf folgen die Aufgabenblätter mitsamt Hinweisblättern sowie der Fragebogen.

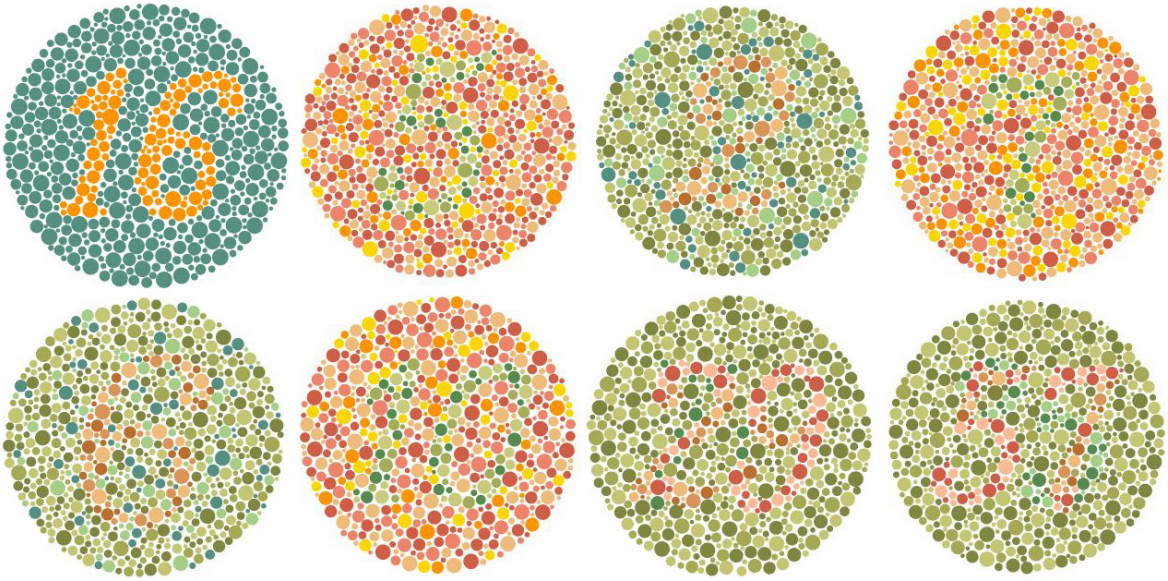


Abbildung A.1.: Ishiharatest [ish11]

Sportnews

***Hint**: Sometimes filtering/selecting may be helpful!*
[***Tipp**: Filter und Selektionen können sehr hilfreich sein und die Suche stark beschleunigen!*]

Task 1

a) What's the most frequently mentioned score? (like 4-4)
[Welches Endergebnis kommt am häufigsten vor? (wie z.B. 4-4)]

-

b) Is the dollar the most frequently mentioned currency (**Money**)?
[Ist der Dollar die am häufigsten vorkommende Währung?]

☐ *Yes* ☐ *No, it's:*

c) Which **day of the week** and which **month** are the most frequently mentioned?
[Welcher Wochentag und welcher Monat wird jeweils am häufigsten erwähnt?]

Day of the week: *Month:*

Task 2

a) What's the most frequently mentioned score concerning **Wimbledon**?
[Welches Endergebnis taucht im Zusammenhang mit Wimbledon am häufigsten auf?]

-

b) What's the name of the **person** who was the **champion** of **Wimbledon**? (Full name)
***Hint**: To get first- and last name combined, it's helpful to activate multi-words!*
(View > Show multi-words)
[Wie heißt der damalige Champion des Wimbledon? (Vor- und Nachname)
***Tipp**: Um Vor- und Nachnamen zusammen zu sehen, können Multiwords verwendet werden! (View > Show multi-words)*]

c) On which **day of the week** are (game-) **results** most often published?
[An welchem Wochentag war mit Spielergebnissen zu rechnen?]

Day of the week:

Task 3

a) Have there been more **win**nings or **lose**s in the most **frequently mentioned month**? (1c)
[Wurden im häufigst-genannten Monat mehr Spiele gewonnen oder verloren? (siehe 1c)

Count of winnings: *Count of loses:* *Answer:*

b) What seems to be the **most important discipline** on the **most frequently mentioned day**? (1c)
[Welche Sportart wird am häufigst-genannten Tag besonders oft erwähnt? (siehe 1c)]

c) Who was the most important **person** for '**Ferrari**'? (Full name)
[Welche Person war für ‚Ferrari' besonders wichtig? (Vor- und Nachname)]

Task 4

a) With which kind of sport can the '**Blacks**' be linked and what seems to be their favorite score?
[Um welche Sportart handelt es sich bei den ‚Blacks' und was scheint deren Lieblingsergebnis zu sein?]

Kind of sport: *Favorite score:*

b) Which **person** seems to play an important role in this team and why?
[Welche Person spielt in diesem Team eine wichtige Rolle und warum?
(Welche Eigenschaft lässt sich über diese Person herausfinden)]

Person: *Reason:*

c) Search for **Olympic champions**. Who are the most frequently mentioned male and female champions? (Full name)
***Hint**: Linford Christie is not female!*
[Gesucht werden zwei olympische Champions, wie heißt der meistgenannte Mann, wie die meistgenannte Frau unter ihnen? (Vor- und Nachname)
***Tipp**: Linford Christie ist nicht weiblich!*]

Male: *Female:*

d) In which **location** were the most **games lost**?
[Wo wurden die meisten Spiele verloren?]

VisWeekAbstracts

Task 1

a) What's the **verb** that's most frequently related with '**algorithm**'?
[Welches Verb steht am häufigsten mit ‚algorithm' in Zusammenhang?]

Verb:

b) What does the result (from above) look like without using lemmatization? Any ideas why?
***Instruction**: Turn **off** the lemmatization (View > Use lemmatization)*
[Wie sieht das Ergebnis aus, wenn man ohne Lemmatisierung arbeitet? Eine Idee woran es liegt?
***Anweisung**: Lemmatisierung **aus**schalten (View > Use lemmatization)*]

Verb: *Possible reason:*

c) How many different word-forms of the two verbs can be found?
***Instruction**: Turn **on** the lemmatization (View > Use lemmatization)*
[Wie viele verschiedene Wortformen der beiden Verben kommen in diesem Text vor?
***Anweisung**: Lemmatisierung **an**schalten (View > Use lemmatization)*]

Verb1a: *wordforms* *Verb1b:* *wordforms*

Task 2

a) Which **person** is linked to the topic '**cells**'?
[Welche Person wird mit dem Thema ‚cells' in Verbindung gebracht?]

Person:

b) What's the '**Nyquist**' '**theory**' probably about?
[Um was scheint es in der ‚Nyquist' ‚theory' zu gehen?]

It has something to do with:

c) Which **noun** is closely related to the Russian mathematician and physicist '**Lyapunov**'?
[Welches Substantiv fällt am häufigsten wenn es um den russischen Mathematiker und Physiker ‚Lyapunov' geht?]

Noun:

d) Concerning **colors**, what's the most frequently related **adjective**?
[Welches Adjektiv kommt am häufigsten im Zusammenhang mit ‚color' vor?]

Adjective:

e) How many sentences do share '**visualization**' and '**information**'?
[In wie vielen Sätzen kommt sowohl ‚visualization' als auch ‚information' vor?]

Count of common sentences:

f) Is the word '**decision**' mostly linked with past or future?
[Wird ‚decision' häufiger mit der Vergangenheit oder Zukunft verbunden?]

☐ *Past* ☐ *Future* ☐ *Equally*

Task 3

The following task has to be solved without clicking any filter and without colorizing Part-Of-Speech!
***Instruction**: Turn **off** colorizing Part-Of-Speech (View > Colorize POS)*
***Hint**: You may use hovering!*
[*Für die folgende Aufgabe sollen keine Filter geklickt und keine farbliche Hervorhebung der Wortarten verwendet werden!*
***Anweisung**: Farbliche Hervorhebung der Wortarten ausschalten (View > Colorize POS)*
***Tipp**: Über die Filter zu fahren ist erlaubt.*]

Concerning '**curves**', which is the most frequently mentioned **adjective** and who is the most frequently related **person**?
[Es geht um ‚curves': welches Adjektiv und welche Person wird am häufigsten damit verbunden?]

Adjective: *Person:*

Task 4

The following task has to be solved without clicking a word or filter and without Searching!
***Hint**: The color-saturation (= co-occurrence frequency) can be changed by raising (up to 14) the minimal co-occurrence-count. (Options > Set minimal co-occurrence count)*
For Determing whether it's a noun, colorizing Part-Of-Speech can help (View > Colorize POS)
[*Für die folgende Aufgabe soll weder auf ein Wort geklickt werden, noch dürfen irgendwelche Filter benutzt werden. Die Suche ist ebenfalls tabu!*
***Tipp**: Um die Unterscheidung der Gelbsättigung leichter zu machen, kann der minimale co-occurrence count (auf bis zu 14) erhöht werden (dafür auf Options klicken). Um Festzustellen ob es sich um Substantive handelt, ist es hilfreich die farbliche Hervorhebung der Wortarten zu aktivieren*]

Concerning '**participants**', which are the two most frequently mentioned **nouns**?
[Es geht um ‚participants': Welche zwei Substantive stehen am Engsten damit in Verbindung?]

Noun 1: *Noun 2:*

Sportnews - Hints

Task 1

a) A score is detected as Cardinal (Part-Of-Speech) or Numeric (Category).
[Ein Endergebnis wird als Cardinal (Part-Of-Speech) oder Numeric (Category) erkannt.]

b) Filtering by 'Money' makes the result more manageable.
[Ein Filtern nach ‚Money' erleichtert die Suche erheblich.]

c) Filtering by 'Date' makes the result more manageable.
[Ein Filtern nach ‚Date' erleichtert die Suche erheblich.]

Task 2

a) Selecting ‚Wimbledon' is necessary. A score is detected as Cardinal / Numeric.
[‚Wimbledon' muss ausgewählt werden. Ein Endergebnis wird als Cardinal oder Numeric erkannt.]

b) The words 'Champion' and 'Wimbledon' have to be selected, now hovering over the filter 'Person' or filtering by 'Person' leads to the goal.
[‚Wimbledon' und ‚Champion' müssen ausgewählt werden und es hilft über ‚Person' zu fahren oder danach zu filtern.]

c) Selecting 'Result' is necessary and hovering over or filtering by 'Date' can help.
[‚Result' muss ausgewählt werden, darüber hinaus hilft es über ‚Date' zu fahren oder danach zu filtern.]

Task 3

a) Selecting 'August' is necessary and looking for verbs (win and lose) speeds up the search.
[‚August' muss ausgewählt werden und über ‚Verb' zu fahren oder danach zu filtern hilft.]

b) Selecting 'Saturday' is necessary and looking for nouns can help a little.
[‚Saturday' muss ausgewählt werden und über ‚Noun' zu fahren oder danach zu filtern hilft.]

c) Selecting 'Ferrari' is necessary and hovering over or filtering by 'Person' can help.
[‚Ferrari' muss ausgewählt werden und über ‚Person' zu fahren oder danach zu filtern hilft.]

Task 4

a) Selecting 'Blacks' is necessary, a score is detected as Cardinal (Part-Of-Speech) or Numeric (Category).
[Die ‚Blacks' müssen ausgewählt werden, Spielergebnisse werden als Cardinal oder Numeric erkannt.]

b) Selecting 'Blacks' is necessary and hovering over or filtering by 'Person' helps a lot. For finding the reason it is very helpful to select the important person.
[Die ‚Blacks' müssen ausgewählt werden und über ‚Person' zu fahren oder danach zu filtern hilft. Um den Grund herauszufinden hilft es sehr, diese Person zu selektieren und alle Filter anzuschalten]

c) Selecting 'Olympic' and 'Champions' is necessary and filtering by 'Person' helps a lot.
[‚Olympic' und ‚Champions' müssen ausgewählt werden, über ‚Person' zu fahren oder danach zu filtern macht das Ergebnis deutlich übersichtlicher.]

d) Selecting 'games' and 'lost' is necessary, hovering over or filtering by 'Location' helps a lot.
[‚games' und ‚lost' müssen ausgewählt werden, über ‚Location' zu fahren oder danach zu filtern macht das Ergebnis deutlich übersichtlicher.]

VisWeekAbstracts - Hints

Task 1

a) Selecting ‘algorithm’ is necessary and hovering over or filtering by ‘Verb’ helps a lot.
[‚algorithm’ muss ausgewählt werden, und über ‚Verb’ zu fahren oder danach zu filtern hilft.]

b) Selecting ‘algorithm’ is necessary and hovering over or filtering by ‘Verb’ helps a lot.
[‚algorithm’ muss ausgewählt werden, und über ‚Verb’ zu fahren oder danach zu filtern hilft.]

c) Hovering over or filtering by ‘Verb’ should be done. The count of different wordforms can be found in the Infopanel.
[Über ‚Verb’ zu fahren oder danach zu filtern ist wichtig. Die genaue Anzahl kann im Infopanel abgelesen werden.]

Task 2

a) Selecting ‘cell’ is necessary and hovering over or filtering by ‘Person’ helps a lot.
[‚cell’ muss ausgewählt werden und über ‚Person’ zu fahren oder danach zu filtern hilft.]

b) Selecting ‘Nyquist’ and ‘theory’ is necessary.
[‚Nyquist’ und ‚theory’ müssen ausgewählt werden.]

c) Selecting ‘Lyapunov’ is necessary and hovering over or filtering by ‘Noun’ helps a lot.
[‚Lyapunov’ muss ausgewählt werden und über ‚Noun’ zu fahren oder danach zu filtern hilft.]

d) Selecting ‘color’ is necessary and hovering over or filtering by ‘Adjective’ helps a lot.
[‚color’ muss ausgewählt werden und über ‚Adjective’ zu fahren oder danach zu filtern hilft.]

e) At least one of the words has to be selected, hovering over the second will show the common sentences (bottom-right of the window) or you can read the (relative) count.
[Mindestens eines der Wörter muss ausgewählt sein. Fährt man nun über das andere, so erscheint unten rechts die Anzahl der gemeinsamen Sätze oder man kann jeweils oben den Zähler ablesen.]

f) Selecting ‘decision’ is necessary and hovering over or filtering by ‘Date’ is helpful. The exact count of occurrences can be found between the search and the panels.
[‚decision’ muss ausgewählt werden und über ‚Date’ zu fahren oder danach zu filtern hilft. Die jeweilige Anzahl wird beim Darüberfahren zwischen der Suche und den Filtern angezeigt.]

Task 3

Selecting ‘curves’ is necessary and hovering over ‘Adjective’ / ‘Person’ helps a lot.
[‚curves’ muss ausgewählt werden und dann über ‚Adjective’ / ‚Person’ gefahren werden. Die Antwort muss komplett gelb sein, da nach einem Adjektiv gefragt wird.]

Task 4

Turn on colorizing Part-Of-Speech (View > Colorize POS) for recognizing nouns.
To find ‘participants’, it’s helpful to use the alphabetical layout.
Hovering over ‘participants’ is necessary and setting the minimal co-occurrence count helps a lot.
[Zunächst ist es von Vorteil die farbliche Hervorhebung von Wortarten zu aktivieren falls nicht schon geschehen. (View > Colorize POS) Um nun das Wort ‚participants’ ohne Suche zu finden, bietet es sich, auf das alphabetisch sortierte Layout zu wechseln. Um nun die damit zusammenhängenden Worte zu farblich markiert zu sehen, muss auf das Wort gefahren werden. Um die Gelbsättigungsstufen eindeutiger unterscheiden zu können, kann der minimale co-occurrence count (Options) auf bis zu 14 erhöht werden.]

Fragebogen zur Experten-Evaluation bezüglich der interaktiven Tag-Cloud

Vielen Dank, dass Sie an der Experten-Evaluation bezüglich der interaktiven Tag-Cloud teilgenommen haben. Der folgende Fragebogen wird nur für wissenschaftliche Zwecke im Rahmen dieser Evaluation verwendet. Alle Angaben, die Sie machen, werden **streng vertraulich** behandelt. Ihre Daten bleiben **anonym**, ein Rückschluss auf Ihre Person ist somit nicht möglich. Bei diesem Fragebogen gibt es keine richtigen und falschen Antworten.

1. **Angaben zur Person:**

   Alter: _____ **Geschlecht:** ☐ männlich ☐ weiblich

2. **Ist bei Ihnen eine Farbfehlsichtigkeit festgestellt worden (z.B. Rot-Grün-Blindheit)?**

   ☐ Ja ☐ Nein

3. **Wie hoch würden Sie Ihre Englischkenntnisse auf einer Skala von 1-10 einschätzen?**

gering	1	2	3	4	5	6	7	8	9	10	*sehr gut*

4. **Sind Sie vor dieser Evaluation schon einer Tag-Cloud begegnet?**

   ☐ Ja ☐ Nein

5. **Gab es irgendwelche Unklarheiten bezüglich der Aufgabenstellung?**

   ☐ Ja ☐ Nein

   Wenn ja: Was war Ihnen unklar?

   _____

   _____

   _____

6. **Hat Ihnen die Bearbeitung der Aufgaben Probleme bereitet?**

   ☐ Ja ☐ Nein

   Wenn ja: welche?

   _____

   _____

   _____

7. **Hatten Sie Probleme bei der Benutzung von Funktionen(Filter, Suche, Selektion, etc.)?**

   ☐ Ja ☐ Nein

   Wenn ja: welche und womit?

   _____

   _____

   _____

8. **Halten Sie den Einsatz dieser interaktiven Tag-Cloud für bestimmte Aufgaben in der Visualisierung für sinnvoll?**

   ☐ Ja ☐ Nein

   Wenn ja: Könnten Sie sich eine Situation oder Aufgabe vorstellen, bei denen dieser Einsatz einen gewissen Mehrwert bieten könnte?

   _____

   _____

   _____

9. **Hätten Sie sich zusätzlichen Funktionen für die Bearbeitung der Aufgaben gewünscht?**

   ☐ Ja ☐ Nein

   Wenn ja: Welche?

   _____

   _____

   _____

10. **Sind Ihnen Verhaltensweisen von Funktionen oder Darstellungen/Rückmeldungen aufgefallen, die Ihnen nicht intuitiv vorkamen?**

    ☐ Ja ☐ Nein

    Wenn ja: Welche?

    _____

    _____

    _____

11. **Haben Sie sonst noch irgendwelche Anmerkungen zur Evaluation?**

    _____

    _____

    _____

VIELEN DANK FÜR IHRE MITARBEIT !!!

Literaturverzeichnis

[AA11] M. Abulaish, T. Anwar. A web content mining approach for tag cloud generation. In *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, iiWAS '11, S. 52–59. ACM, New York, NY, USA, 2011. (Zitiert auf den Seiten 24 und 26)

[ASM10] H. Aras, S. Siegel, R. Malaka. Semantic Cloud: An Enhanced Browsing Interface for Exploring Resources in Folksonomy Systems. In *Workshop on Visual Interfaces to the Social and Semantic Web*, VISSW '10. CEUR-WS.org, 2010. URL http://ceur-ws.org/Vol-565/paper5.pdf. (Zitiert auf Seite 27)

[Bau07] B. Baumann. Named Entity Recognition. Technische Universität Dortmund, 2007. (Zitiert auf Seite 24)

[BGN08] S. Bateman, C. Gutwin, M. Nacenta. Seeing things in the clouds: the effect of visual features on tag cloud selections. In *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia*, HT '08, S. 193–202. ACM, New York, NY, USA, 2008. (Zitiert auf Seite 14)

[clo] Cloudio. URL https://github.com/sschwieb/Cloudio. (Zitiert auf Seite 64)

[cor] The Stanford Natural Language Processing Group. URL http://nlp.stanford.edu/software/corenlp.shtml. (Zitiert auf den Seiten 25, 79, 83 und 85)

[CVW09] C. Collins, F. B. Viégas, M. Wattenberg. Parallel Tag Clouds to explore and analyze faceted text corpora. In *IEEE Conference on Visual Analytics Science and Technology*, VAST '09, S. 91–98. IEEE, Atlantic City, NJ, USA, 2009. (Zitiert auf den Seiten 18 und 19)

[CWL+10] W. Cui, Y. Wu, S. Liu, F. Wei, M. X. Zhou, H. Qu. Context-Preserving, Dynamic Word Cloud Visualization. *IEEE Computer Graphics and Applications*, 30(6):42–53, 2010. (Zitiert auf Seite 19)

[Dah06] M. Dahm. *Grundlagen der Mensch-Computer-Interaktion*. Pearson Studium, München, 2006. (Zitiert auf Seite 44)

[Dav] J. Davis. Word Cloud Generator. URL http://www.jasondavies.com/wordcloud/. (Zitiert auf Seite 19)

[DGWC10] M. Dork, D. Gruen, C. Williamson, S. Carpendale. A Visual Backchannel for Large-Scale Events. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1129–1138, 2010. (Zitiert auf Seite 18)

[Die98] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998. (Zitiert auf Seite 23)

[Doy01] S. A. C. Doyle. *The Hound of the Baskervilles*. Project Gutenberg, 2001. URL http://www.gutenberg.org/ebooks/2852. (Zitiert auf den Seiten 58, 68 und 69)

[Fei10] J. Feinberg. Wordle. In *Beautiful Visualization: Looking at Data through the Eyes of Experts*, Kapitel 3, S. 37–58. O'Reilly Media, Sebastopol, CA, USA, 2010. (Zitiert auf Seite 23)

[Fei11] J. Feinberg. Wordle – Beautiful Word Clouds, 2011. URL http://www.wordle.net. (Zitiert auf den Seiten 15, 16 und 17)

[FGM+99] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1, 1999. URL http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html. (Zitiert auf Seite 16)

[FGM05] J. R. Finkel, T. Grenager, C. Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, S. 363–370. Association for Computational Linguistics, Stroudsburg, PA, USA, 2005. (Zitiert auf Seite 24)

[Fin07] J. R. Finkel. Named Entity Recognition and the Stanford NER Software. Stanford University, 2007. (Zitiert auf Seite 24)

[fli] Die beliebtesten Tags aller Zeiten. URL http://www.flickr.com/photos/tags/. (Zitiert auf Seite 13)

[FP10] M. Faruqui, S. Padó. Training and Evaluating a German Named Entity Recognizer with Semantic Generalization. In *Proceedings of the Conference on Natural Language Processing*. Saarbrücken, Germany, 2010. (Zitiert auf Seite 85)

[gut] Project Gutenberg - free ebooks. URL http://www.gutenberg.org. (Zitiert auf den Seiten 66 und 68)

[GV10] P. Gambette, J. Véronis. Visualising a Text with a Tree Cloud. In *Classification as a Tool for Research*, Studies in Classification, Data Analysis, and Knowledge Organization, S. 561–569. Springer, Berlin/Heidelberg, Germany, 2010. (Zitiert auf Seite 20)

[HK07] M. J. Halvey, M. T. Keane. An assessment of tag presentation techniques. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, S. 1313–1314. ACM, New York, NY, USA, 2007. (Zitiert auf Seite 19)

[HKBE12] F. Heimerl, S. Koch, H. Bosch, T. Ertl. Visual Classifier Training for Text Document Retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2839–2848, 2012. (Zitiert auf den Seiten 16, 36 und 68)

[Holo5] A. Holzinger. Usability engineering methods for software developers. *Communications of the ACM*, 48(1):71–74, 2005. (Zitiert auf Seite 67)

[Hol11] P. Holme. Peter Holme's word stemmer, 2011. URL http://holme.se/stem/. (Zitiert auf den Seiten 16 und 17)

[int] Das qualitative Interview. URL http://arbeitsblaetter.stangl-taller.at/FORSCHUNGSMETHODEN/Interview.shtml. (Zitiert auf Seite 69)

[ish11] Ishihara Test for Color Blindness, 2011. URL http://ebookbrowse.com/ishihara-test-for-color-blindness-pdf-d133178847. (Zitiert auf den Seiten 67 und 91)

[KHGW07] B. Y.-L. Kuo, T. Hentrich, B. M. . Good, M. D. Wilkinson. Tag clouds for summarizing web search results. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, S. 1203–1204. ACM, New York, NY, USA, 2007. (Zitiert auf Seite 19)

[KL07] O. Kaser, D. Lemire. Tag-Cloud Drawing: Algorithms for Cloud Visualization. In *Proceedings of World Wide Web 2007 Workshop on Tagging and Metadata for Social Information Organization*. Banff, Canada, 2007. (Zitiert auf Seite 19)

[KLKS10] K. Koh, B. Lee, B. Kim, J. Seo. ManiWordle: Providing Flexible Control over Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1190–1197, 2010. (Zitiert auf Seite 19)

[Kol12] P. Kolbe. Frei verfügbare NLP-Tools und -Web-Services für die deutsche Sprache, 2012. URL http://www.ling.uni-potsdam.de/~kolb/nlp-tools.html. (Zitiert auf Seite 25)

[LBSW12] S. Lohmann, M. Burch, H. Schmauder, D. Weiskopf. Visual analysis of microblog content using time-varying co-occurrence highlighting in tag clouds. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, S. 753–756. ACM, New York, NY, USA, 2012. (Zitiert auf den Seiten 17 und 18)

[Leu] H. Leung. Taxedo - Word Cloud with Styles. URL http://www.tagxedo.com. (Zitiert auf Seite 19)

[Lewo5] D. Lewandowski. *Web Information Retrieval : Technologien zur Informationssuche im Internet*. Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis, Frankfurt am Main, 2005. (Zitiert auf Seite 25)

[Lido1] E. D. Liddy. Natural Language Processing. In *Encyclopedia of Library and Information Science*, ELIS '01. Marcel Decker, New York, NY, USA, 2001. (Zitiert auf Seite 23)

[LZT09] S. Lohmann, J. Ziegler, L. Tetzlaff. Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration. In *Human-Computer Interaction – INTERACT 2009*, S. 392–404. Springer, Berlin/Heidelberg, Germany, 2009. (Zitiert auf den Seiten 14, 15, 19, 27 und 85)

[Mat04] A. Mathes. Folksonomies - Cooperative Classification and Communication Through Shared Metadata. Technischer Bericht, Graduate School of Library and Information Science, University of Illinois Urbana-Champaign, 2004. URL http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html. (Zitiert auf Seite 20)

[MMS93] M. P. Marcus, M. A. Marcinkiewicz, B. Santorini. Building a large annotated corpus of English: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993. (Zitiert auf Seite 57)

[MPR00] L. Màrquez, L. Padró, H. Rodríguez. A Machine Learning Approach to POS Tagging. *Machine Learning*, 39(1):59–91, 2000. (Zitiert auf Seite 24)

[MS99] C. D. Manning, H. Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA, 1999. (Zitiert auf den Seiten 24, 25 und 26)

[Nie00] J. Nielsen. Why You Only Need to Test with 5 Users, 2000. URL http://www.useit.com/alertbox/20000319.html. (Zitiert auf Seite 67)

[NS07] D. Nadeau, S. Sekine. A survey of named entity recognition and classification. *Linguisticæ Investigationes*, 30(1):3–26, 2007. (Zitiert auf den Seiten 23 und 24)

[OC10] J. Oosterman, A. Cockburn. An empirical comparison of tag clouds and tables. In *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*, OZCHI '10, S. 288–295. ACM, New York, NY, USA, 2010. (Zitiert auf Seite 19)

[Por97] M. F. Porter. An algorithm for suffix stripping. In *Readings in information retrieval*, S. 313–316. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1997. (Zitiert auf Seite 24)

[Por01] M. F. Porter. Snowball: A language for stemming algorithms, 2001. URL http://snowball.tartarus.org/texts/introduction.html. (Zitiert auf Seite 24)

[pos] Penn Treebank P.O.S Tags. URL http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html. (Zitiert auf Seite 57)

[pro] Processing.org. URL http://processing.org. (Zitiert auf Seite 64)

[PTT+12] F. V. Paulovich, F. M. B. Toledo, G. P. Telles, R. Minghim, L. G. Nonato. Semantic Wordification of Document Collections. *Computer Graphics Forum*, 31(3):1145–1153, 2012. (Zitiert auf Seite 19)

[reu] Reuters. URL http://www.reuters.com. (Zitiert auf Seite 68)

[RGMM07] A. W. Rivadeneira, D. M. Gruen, M. J. Muller, D. R. Millen. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, S. 995–998. ACM, New York, NY, USA, 2007. (Zitiert auf den Seiten 14 und 19)

[Rou11a] M. Rouse. Natural Language Processing (NLP), 2011. URL http://searchcontentmanagement.techtarget.com/definition/natural-language-processing-NLP. (Zitiert auf Seite 23)

[Rou11b] M. Rouse. What is machine learning?, 2011. URL http://whatis.techtarget.com/definition/machine-learning. (Zitiert auf Seite 23)

[SA11] D. Skoutas, M. Alrifai. Tag clouds revisited. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, S. 221–230. ACM, New York, NY, USA, 2011. (Zitiert auf Seite 13)

[SBB+02] I. A. Sag, T. Baldwin, F. Bond, A. A. Copestake, D. Flickinger. Multiword Expressions: A Pain in the Neck for NLP. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, S. 1–15. Springer, London, UK, 2002. (Zitiert auf Seite 24)

[Sch94] H. Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK, 1994. (Zitiert auf Seite 24)

[Sch95] H. Schmid. Improvements In Part-of-Speech Tagging With an Application To German. In *Proceedings of the Association for Computer Linguistics SIGDAT-Workshop*, S. 47–50. The Association for Computer Linguistics, Dublin, Ireland, 1995. (Zitiert auf Seite 24)

[SCH08] J. Sinclair, M. Cardew-Hall. The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34(1):15–29, 2008. (Zitiert auf den Seiten 19, 20 und 27)

[sch12] Schlagwortwolke - Wikipedia, 2012. URL http://de.wikipedia.org/wiki/Schlagwortwolke. Version vom 07.08.2012. (Zitiert auf Seite 13)

[ser12] Serendipity - Wikipedia, the free encyclopedia, 2012. URL http://en.wikipedia.org/wiki/Serendipity. Version vom 27.12.2012. (Zitiert auf Seite 20)

[SHH99] S. Suri, P. M. Hubbard, J. F. Hughes. Analyzing bounding boxes for object intersection. *ACM Transactions on Graphics*, 18(3):257–277, 1999. (Zitiert auf Seite 51)

[SLT09] J. Schrammel, M. Leitner, M. Tscheligi. Semantically structured tag clouds: an empirical evaluation of clustered presentation approaches. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, S. 2037–2040. ACM, New York, NY, USA, 2009. (Zitiert auf Seite 19)

[Ste06] D. Steinbock. TagCrowd, 2006. URL http://www.tagcrowd.com/. (Zitiert auf Seite 18)

[tag] Tagul - Gorgeous tag clouds. URL http://tagul.com. (Zitiert auf den Seiten 13, 14, 19 und 20)

[TKMS03] K. Toutanova, D. Klein, C. D. Manning, Y. Singer. 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*, NAACL '03, S. 173–180. Association for Computational Linguistics, Stroudsburg, PA, USA, 2003. (Zitiert auf Seite 24)

[twi] What is Twitter? Twitter for Business. URL https://business.twitter.com/basics/what-is-twitter/. (Zitiert auf Seite 17)

[vis] IEEE VIS 2013. URL http://visweek.org. (Zitiert auf den Seiten 16, 36 und 68)

[VW08] F. B. Viégas, M. Wattenberg. Timelines: Tag clouds and the case for vernacular visualization. *Interactions - Changing energy use through design*, 15(4):49–52, 2008. (Zitiert auf Seite 14)

[VWF09] F. B. Viegas, M. Wattenberg, J. Feinberg. Participatory Visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1137–1144, 2009. (Zitiert auf Seite 16)

[wora] WordCram.org - open-source word clouds for Processing. URL http://wordcram.org. (Zitiert auf Seite 64)

[worb] Wordookie - The open alternative to Wordle. URL http://code.google.com/p/wordookie/. (Zitiert auf Seite 64)

[WW05] J. Weeds, D. Weir. Co-occurrence Retrieval: A Flexible Framework for Lexical Distributional Similarity. *Computational Linguistics*, 31(4):439–475, 2005. (Zitiert auf Seite 25)

Alle URLs wurden zuletzt am 07.01.2013 geprüft.

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Simon Lange)