



LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

CENTRUM FÜR INFORMATIONS- UND SPRACHVERARBEITUNG
STUDIENGANG COMPUTERLINGUISTIK



Bachelorarbeit

im Studiengang Computerlinguistik

an der Ludwig- Maximilians- Universität München

Fakultät für Sprach- und Literaturwissenschaften

Musik und Ludwig Wittgenstein: Semantische Suche in seinem Nachlass.

vorgelegt von
Ines Röhler

Betreuer: Dr. Maximilian Hadersbeck
Prüfer: Dr. Maximilian Hadersbeck
Bearbeitungszeitraum: 07. April - 12. Juni 2017

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, den 12. Juni 2017

.....
Ines Röhrer

Abstract

Die vorliegende Arbeit beschäftigt sich mit der Implementierung einer Erweiterung für die semantische Suche der WiTTFind Web App. Die FinderApp „WiTTFind“ ist eine eigens für einen Korpus bestehend aus den Werken Ludwig Wittgensteins konzipierte Suchmaschine. Diese ist ein wichtiges Tool um den Nachlass Ludwig Wittgensteins zu untersuchen und erforschen. Das Tool enthält, neben einer regelbasierten Suche, bereits eine Option für semantisches Suchen in der Kategorie Farbe“, wobei Begriffe dieser Kategorie nach Gruppen geteilt abgerufen werden können. Die semantische Suche wird nun, basierend auf einem Lexikon musikalischer Ausdrücke aus dem Wittgensteinkorpus, um eine Kategorie für musikalische Begriffe erweitert. Mit dieser Kategorie können sowohl die musikalischen Ausdrücke selbst, in Gruppen aufgeteilt, als auch ihr Kontext ausgegeben werden.

This bachelor's thesis deals with implementations extending the semantic search function of the web application WiTTFind. As a search engine specifically created for a corpus consisting only of Ludwig Wittgenstein's works WiTTFind is an important tool to analyze L. Wittgenstein's inheritance. The search function contains both a rulebased and a semantic search option. A section for terms of 'colour', in which the terms are separated by category, is already incorporated. This semantic search option now is being extended by a section for music related terms from the Wittgensteincorpus. With this new addition the musical terminology which is divided into classes as well as their context can be displayed.

Inhaltsverzeichnis

| | |
|--|-----------|
| Abstract | I |
| 1 Einleitung | 3 |
| 2 Relevante Arbeiten | 5 |
| 2.1 Semantische Annotation von Adjektiven im Big Typescript von Ludwig Wittgenstein | 5 |
| 2.2 Bemerkungen über Musik im Nachlass und den publizierten Schriften Ludwig Wittgensteins | 6 |
| 3 Frontend zur semantischen Suche bei WiTTFind | 9 |
| 3.1 Lokale Webservereinrichtung mithilfe eines Tutorials | 9 |
| 3.2 Eigenschaften des Frontends | 10 |
| 4 Semantische Suche für musikalische Kategorien | 13 |
| 4.1 Struktur der Korpora | 13 |
| 4.2 Arbeiten auf Grundlage eines Musiklexikons | 13 |
| 4.3 Semantische Kategorien | 14 |
| 4.4 Wordclouddarstellung der musikalischen Begriffe | 15 |
| 4.5 Frequenzberechnungen der relevanten Begriffe | 16 |
| 4.6 Finden der Vollformen | 17 |
| 4.7 Probleme | 19 |
| 4.7.1 Leerzeichenfehler | 19 |
| 4.7.2 Multiple Satzvorkommen | 19 |
| 5 Kontext musikalischer Begriffe | 21 |
| 5.1 Darstellung und Relevanz von Kontext | 21 |
| 5.2 Berechnung des Kontextes | 21 |
| 5.2.1 Stopwörter und Ausschlussliste | 21 |
| 5.2.2 Verwendete Methoden | 22 |
| 5.2.3 Auswahl des Kontextumfangs | 26 |
| 5.3 Kontextfrequenztabellen | 26 |
| 6 Verwendung von Ontologien für musikalische Relationen und Eigenschaften | 29 |
| 6.1 Musikalische Relationen und Eigenschaften | 29 |
| 6.2 Umwandlung einer Tabelle zur Ontologie | 31 |
| 6.2.1 Musikontologie 'MO' | 33 |
| 7 Evaluierung - Benutzerfreundlichkeit der Webseite | 35 |
| 7.1 Erstellung des Fragebogens | 35 |
| 7.2 Auswertung der Ergebnisse | 39 |
| 8 Ausblick | 41 |
| 8.1 Konkordanzdarstellung für Kontext | 41 |
| 8.2 Ontologie Darstellung | 41 |
| 9 Zusammenfassung/Fazit | 43 |

| | |
|----------------------------------|-----------|
| Literaturverzeichnis | 45 |
| Abbildungsverzeichnis | 47 |
| Tabellenverzeichnis | 49 |
| Inhalt der beigelegten CD | 51 |

1 Einleitung

"It is impossible for me to say in my book one word about all that music has meant in my life. How then can I hope to be understood?"

- Ludwig Wittgenstein¹

Dieses Zitat aus einem Gespräch mit Drury macht deutlich, wie wichtig Musik in Ludwig Wittgensteins Leben war. Diese bedeutsame Rolle von Musik in seinem Leben wird schon früh durch die Musikalität in der Familie Wittgenstein geprägt. Wie essentiell eine musikalische Betrachtung seines Werkes ist, um es in seiner Gänze verstehen zu können, zeigt das nächste Zitat aus dem Buch „Ludwig Wittgenstein als Musikphilosoph“, Eggers (2014):

„Wittgenstein selbst war wohl der Ansicht, daß niemand seine Philosophie verstehe, der sie nicht auf dem Hintergrund jenes in ihr Nicht-Gesagten versteht; einem Hintergrund, der für uns literarisch nur faßbar ist in den Gesprächen, den Bemerkungen zur Kunst und zur Religion, ja in seinen moralischen und ästhetischen Reflexen und daher in jenen seiner Äußerungen, die seiner Biographie ungleich näher sind als die eigentlich philosophischen Texte.“

Um die Bedeutung von Musik in Wittgensteins Werk besser verstehen zu können, ist ein erster Schritt die Betrachtung von musikalischen Erwähnungen in seinem Nachlass.

Die wissenschaftliche Analyse digitaler Korpora, wie Zeitungsartikel, medizinische Sammlungen, Tweets oder Wikipediakorpora, mithilfe von Programmen oder sogar eigens dafür entwickelten Algorithmen, ist schon seit einigen Jahren sehr verbreitet. Doch auch sich nicht ständig selbst erneuernde abgeschlossene Korpora mit literarischem Wert bieten sich für solche digitalen Herangehensweisen an.

Suchmaschinen wie die Finder Apps „GoetheFind“ und „WiTTFind“ sind Schnittstellen zwischen dem originalen, handschriftlichen Werk - repräsentiert in Form von Faksimile - und ihren digitalisierten Editionen, in welchen nach Ausdrücken gesucht werden kann.

Die Analyse von Musikbegriffen ist von semantischer Natur und passt daher sehr gut in die schon bestehende Option zur semantischen Suche in WiTTFind. In dieser Sektion existiert bereits eine Suchoption für Farbenbegriffe. Hier werden durch das Angebot von Wordclouds und Frequenzlisten dem Nutzer vielfältige Möglichkeiten geboten, in die Thematik einzusteigen. Zur Erweiterung dieser Sektion um ein Modul für musikalische Ausdrücke werden kategorisierte Musikbegriffe verwendet.

Verarbeitet werden diese Daten mit Pythonskripten, für die Webdarstellung wird HTML sowie Javascript verwendet. Für die Ontologie musikalischer Begriffe wird RDF, eine der Standardsprachen für Semantic Web Implementierungen, genutzt.

In den folgenden Kapiteln werden diese Entwicklungen und Untersuchungen im Detail vorgestellt.

¹Aus Rhees (1984)

2 Relevante Arbeiten

Der folgende Teil gibt einen Überblick über andere Arbeiten, welche für dieses Thema relevant sind, oder sogar wichtige Vorarbeit geleistet haben. Auch die Arbeit, in der das Lexikon der Musikbegriffe entstanden ist, das als Basis für alle in dieser Arbeit gezeigten Entwicklungen dient, wird besprochen.

Die Basis, für das in dieser Arbeit verwendete Musiklexikon, stammt aus einer Hausarbeit, welche ebenfalls besprochen wird.

2.1 Semantische Annotation von Adjektiven im Big Typescript von Ludwig Wittgenstein

Die im folgende beschriebene Bachelorarbeit von Krey (2013) leistet viel grundlegende Vorarbeit, was semantische Wortkategorien für die Werke Ludwig Wittgensteins angeht. Vieles, was in dieser Arbeit vorgestellt wird, baut auf ihren Untersuchungen und Entwicklungen auf, oder nutzt ähnliche Methoden.

Ziel der Arbeit war es, speziell abgestimmte, semantische Adjektivklassen für das Big Typeskript (TS-213) zu erstellen und in das WiTTFindsuchsystem zu integrieren. Die Verwendung eines klassischen Klassifikationsmodells stellte sich als unbrauchbar heraus, daher wurde ein eigens autorenspezifisch auf Ludwig Wittgenstein abgestimmtes Klassifikationsmodell entwickelt.

Die lexikalischen Datenbanken WordNet und GermaNet, welche nicht nur die Semantik von Worten sondern auch deren Relationen untereinander enthalten, wurden verwendet. GermaNet ist die in Tübingen entwickelte, deutsche Version von WordNet, der ursprünglichen, englischen Datenbank. In GermaNet sind erstmals semantische Klassen für Adjektive vorhanden, welche auch als Vergleich für die neu eingeführten Adjektivklassen genutzt wurden.

Für die Lexikonarbeit wurden die Lexika witt_WAB_DELA und CISLEX genutzt. Eine aktuelle Version des witt_WAB_DELA Lexikons wird auch für die in Abschnitt 4.6 beschriebene Lexikonarbeit verwendet.

In dieser Bachelorarbeit fiel auf, dass einige Adjektive nicht eindeutig differenzierbar und klassifizierbar sind, und so in mehreren Kategorien vertreten waren. Dass Begriffe schwierig in Klassen einteilbar sind, hat bei der Arbeit mit musikalischen Ausdrücken ebenfalls eine Rolle gespielt. Dies wird beispielsweise in den Abschnitten 4.3 und 6.1 beschrieben.

Ein Beispiel aus der Arbeit von Krey (2013) für ein derartiges Adjektiv ist Folgendes:

Das Wort „gut“ :

1. Bedeutung: Eigenschaft einer Person
2. Bedeutung: Evaluation von Etwas

Daher ist „gut“ sowohl Mitglied der Klasse 'Eigenschaft' wie auch der Klasse 'Evaluation'.

Die Namen der eingeführten Kategorien wurden möglichst kurz und aussagekräftig gewählt, damit die Verwendung ihrer Tags erleichtert wird. In der folgenden Aufzählung stehen die jeweiligen Tags der Kategorien in eckigen Klammern.

Die eingeführten Adjektivklassen lauten:

- Farben [COL]
- Numeralia [NUM]
- Relation [REL]
- Eigennamen [EN]
- Temporalia [TEMP]
- Evaluation [EVAL]
- Zustände [ZU]
- Komparativa [KOMP]
- Stilistika [STIL]
- Eigenschaft [EIG]
- Ereignisse [ER]

Die Einarbeitung von Ludwig Wittgensteins Philosophie, sowie seinen Gedanken zu verschiedenen Themen, wie beispielsweise der Farbenlehre, zeichnen diese Arbeit besonders aus. So wurde aufgrund der großen Bedeutung, die Ludwig Wittgenstein der Farbenlehre in seinem Werk gibt, eine eigene Kategorie für farbenbeschreibende Adjektive geschaffen.

Die gesteckten Ziele dieser Arbeit wurden erreicht, sowie gute Vorarbeit für zukünftige Weiterentwicklungen geschaffen.

2.2 Bemerkungen über Musik im Nachlass und den publizierten Schriften Ludwig Wittgensteins

Die hier beschriebene Arbeit von Dziurla (2016) ist eine Hausarbeit mit dem Titel „Bemerkungen über Musik im Nachlass und den publizierten Schriften Ludwig Wittgensteins“. In ihr werden die Grundsteine des verwendeten Musiklexikons über L. Wittgensteins Nachlass gelegt.

Zuerst wird in der Arbeit eine Einführung zur Bedeutung von Musik in Ludwig Wittgensteins Leben gegeben. Denn obwohl er selbst erst im Erwachsenenalter die Klarinette als sein erstes Instrument erlernte, spielte Musik in seiner Familie immer eine große Rolle. Der Vater Ludwig Wittgensteins spielte selbst Geige und Horn, und war zeitweise in Amerika in einer Jazzkapelle tätig. Der Bruder Paul Wittgenstein war ein großer Pianist, der auch, nachdem er im Ersten Weltkrieg seinen rechten Arm verloren hatte, noch erfolgreich Musikstücke für die linke Hand schrieb sowie interpretierte.

Ein weiterer ausschlaggebender Punkt wird vorgestellt: Die häufige Anwesenheit von verschiedenen namhaften Interpreten im Palais Wittgenstein, welches sich als Treffpunkt etabliert hatte. Daher traf Ludwig Wittgenstein Persönlichkeiten wie Josef Labor, Johannes Brahms, Clara Schumann und Gustav Mahler. Sie alle erwähnt er auch in seinen Werken.

Nach dieser Einführung werden verschiedene Bemerkungen und Urteile L. Wittgensteins über verschiedene Komponisten herausgestellt. Während er beispielsweise für Beethoven oder Brahms Bewunderung ausdrückte, sprach er Gustav Mahler in seinen Werken das Talent zum Komponisten ab, obwohl er ihn als Dirigenten durchaus schätzte.

Anschließend werden einige musikalische Begriffe in L. Wittgensteins Philosophie eingeordnet oder mit ihr gedeutet. So wird der Ausdruck des 'Aspekt-Hörens' aus dem Wittgensteinschen 'Aspekt-Sehen' und einiger Textstellen mit musikalischer Terminologie ab-

geleitet.

Abschließend wird das Musiklexikon, die in Kategorien eingeteilten Musikbegriffe, vorgestellt. Diese bilden auch die Basis der bei allen Untersuchungen in dieser Arbeit verwendeten musikalischen Ausdrücke.

3 Frontend zur semantischen Suche bei WiTTFind

Dieses Kapitel gibt einen Überblick darüber, welche Grundlagen für die Weiterentwicklung des WiTTFind Web Frontends vorhanden sind und mit welchen Methoden daran gearbeitet wurde.

3.1 Lokale Webservereinrichtung mithilfe eines Tutorials

Um an der WiTTFind Webseite arbeiten zu können, bietet es sich an, eine Kopie der Webseite auf einem lokalen Server zu haben. So ist es möglich, direkt im Quellcode der Webseite Änderungen vorzunehmen oder auch Inhalte hinzuzufügen, und sofort die Ergebnisse im Web-Frontend auf dem eigenen Rechner zu sehen.

Um dies umzusetzen, wurde ein lokaler Apache-Webserver aufgesetzt, sowie die Schritte eines Tutorials für eine solche lokale Darstellung der WiTTFind Seite befolgt. Das Tutorial ist eigens für eine derartige Einrichtung der WiTTFind Webseite erstellt worden.

Das „Apache HTTP Server Project“¹ entwickelt einen Open-Source HTTP Server für verschiedene Betriebssysteme, unter anderem Unix und Windows. Dieser Server orientiert sich an geltenden HTTP Standards und versucht so effizient, sicher und erweiterbar wie möglich zu sein. Da dieser zudem leicht aufzusetzen ist, fiel die Entscheidung auf den Apache Webserver.

Den Punkten des Tutorials folgend, wurde zunächst das WiTTFind Web Repository geklont, um alle Dateien, welche die Suchalgorithmen und Webdarstellung benötigen, auf dem eigenen Rechner zu haben. Hierbei ist es wichtig, dass alle Dateien im Rootverzeichnis des Webservers liegen, sonst ist die Webdarstellung nicht einwandfrei und es treten Fehler auf. Eine mögliche Lösung ist, vom geklonten Verzeichnis durch einen symbolischen Link, benannt mit „html“, in den Web Server Root zu verlinken.

Anschließend ist es erforderlich, das Tool „Bower“² auf das Verzeichnis anzuwenden und selbiges somit nutz- und darstellbar zu machen. Bower ist ein Packet Manager für Webseiten, der dafür sorgt, dass benötigte Softwarepakete in der aktuellen Version vorhanden sind, wobei dennoch die Ladezeiten und der Ressourcenverbrauch gering gehalten werden sollen.

Schließlich muss noch ein Perlscript auf vorhandene, für die Wordclouddarstellung erforderliche Frequenzlisten angewandt werden. Diese Frequenzlisten liegen im '.txt' Format vor und das Skript wandelt diese in '.json' Dateien um, welche später die im Frontend sichtbaren Frequenzlisten bilden. Hierbei ist es wichtig, dass diese '.json' Dateien, genau wie alle anderen Frequenzlisten, die dargestellt werden sollen, in dem „data“ benannten Ordner liegen. Nur json-Dateien aus diesem Verzeichnis werden als Wordcloud und Frequenzliste in Web Frontend angezeigt.

Dann ist die Webseite auf dem lokalen Webserver darstellbar und wird im Browser aufgerufen über 'http://localhost'. Nach dieser Einrichtung konnte WiTTFind lokal bear-

¹Siehe <https://https.apache.org/>

²Siehe <https://bower.io/>

beitet und verändert werden. Gerade bei einer Frontend Entwicklung ist es sehr wichtig, direkt zu sehen, welche Auswirkungen neue Änderungen haben. Da man nicht online an der laufenden Seite testen kann, sollte ein erster Schritt der Vorbereitung einer solchen Entwicklungstätigkeit eine Webservereinrichtung sein.

3.2 Eigenschaften des Frontends

Um die Ergebnisse der Frequenz- und Kontextanalysen aus den Abschnitten 4 und 5 direkt im Webfrontend sehen und anpassen zu können, wurde dieses zunächst auf die neuen Inhalte vorbereitet. Mit den im Nachstehenden beschriebenen Schritten wurde das Webfrontend um die Kategorie 'Musik' erweitert:

Die schon vorhandene Suchmaske für Farbenbegriffe diente als Orientierung für die neu hinzukommende Musiksuche. Beide Suchsektionen sind in Abbildung 3.1 zu sehen. Für das Modul der semantischen Suche musikalischer Ausdrücke bot sich als Position der Platz direkt unter der Farbensuche an, da dies die beiden Module mit einem festgelegten semantischen Rahmen sind.

Die Kategorien bekamen je eine eigene Box, auch nach dem Beispiel der Farbensuche. Die Links in jeder Box führen zu in 4.4 beschriebenen Wordclouddarstellungen. Bei dem Klicken auf den je ersten dieser drei Links wird eine Wordcloud aus der Frequenzliste von Begriffen der jeweiligen Kategorie angezeigt. Die anderen beiden Links haben die Funktion Wordclouds der Kontexte jeder Kategorie anzuzeigen. Hier gibt es jeweils die Option, den Kontext mit oder ohne Stopwörter darstellen zu lassen.

In der HTML Datei, welche für die Optik des Frontends verantwortlich ist, wurden die Abschnitte für die Musiksektion genau gleich aufgebaut wie das Modul für Farben. Lediglich zwei zusätzliche Links pro Kategorie für die Kontextsuche mussten addiert werden.

Für die weiteren Funktionen, wie die Wordclouddarstellung oder die Eingabe in der Suchleiste, ist eine Javaskriptdatei zuständig. Um etwas Disambiguierungsarbeit vorweg zu nehmen, wird in der Suchleiste angegeben, von welcher Wortart ein gesuchter Ausdruck sein soll. So kann sichergestellt werden, dass bei einer Suche nach einem Wort aus einer Klasse, in welcher nur Nomen enthalten sein sollen, keine Verben angezeigt werden. Ein Beispiel wäre das Wort „Tasten“, womit Klaviertasten oder Ähnliches gemeint sind, es sollten bei einer Suche danach keine Ergebnisse, welche eine Konstruktion mit dem Verb „tasten“ enthalten, angezeigt werden. Also wurden in der Javaskriptdatei für jede Kategorie festgelegt, ob und welche Wortarten als Einschränkung in der Suchleiste hinzugefügt werden. Für die Kategorie „Sonstige musikalische Begriffe“ ergibt eine Wortarteneinschränkung keinen Sinn da hier Ausdrücke von jeder Wortart enthalten sind, weshalb für diese Kategorie nichts festgelegt wurde. Ähnliches gilt für Wörter der Klasse „Bezug zu Komposition“. In den Kategorien „Instrumente“, „Gattungen“ und „Intervalle“, sollten bisher nur Nomen enthalten sein. Für die Wörter aus der Klasse „Komponisten“ gilt, dass jedes ein Eigenname ist. Diese sind zum einen nicht zu finden, wenn man für sie festlegt, dass nur Textstellen angezeigt werden, in welchen das gesuchte Wort als Nomen auftritt. Zum anderen lässt sich hier auch sehr wenig disambiguieren, da eine mögliche Verwechslungsgefahr sowieso nur mit anderen Nomen gegeben ist. Ein Beispiel ist „(Johann Sebastian) **Bach**“, welcher mit dem „**Bach**“ als Flusslauf zu verwechseln ist.

Mit diesem Frontenddesign wurde eine gute Basis für die Einbindung der in den nächsten Kapiteln beschriebenen Implementationen und Analysen geschaffen. Die veränderten HTML und Javascript Dateien befinden sich auf der beigelegten CD.

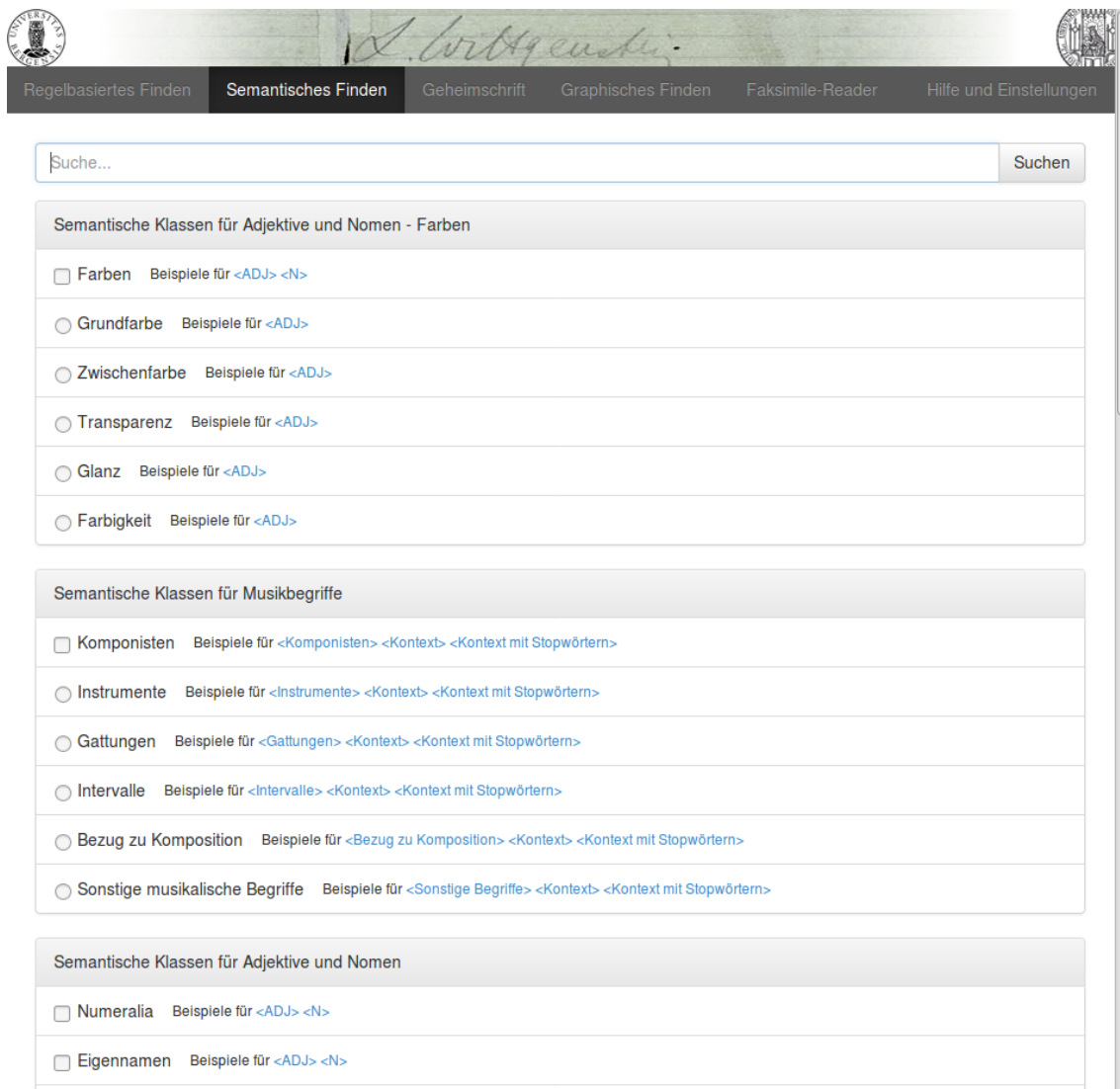


Abbildung 3.1: Screenshot der WiTTFind Webseite auf dem lokalen Webserver

4 Semantische Suche für musikalische Kategorien

Im folgenden Abschnitt „Semantische Suche für musikalische Kategorien“ wird dargestellt, welche Schritte nötig waren, um die Arbeit an der semantischen Suche zu beginnen. Im Anschluss wird genauer auf die Programme zur Frequenzberechnung und Darstellung der musikalischen Begriffe, sowie auf speziell bei diesen Schritten aufgetretene Probleme eingegangen. Der gesamte Programmcode, der für die semantische Suche entwickelt wurde, befindet sich auf der beigelegten CD.

4.1 Struktur der Korpora

In den in dieser Arbeit vorgestellten Untersuchungen wurden mehrere Korpora verwendet und bearbeitet. Diese Korpora haben nicht alle den gleichen Inhalt, sondern teilweise nur inhaltliche Überschneidungen. Auch sind sie unterschiedlich strukturiert: einmal ist pro Zeile ein Satz aufgeführt, ein anderes mal befindet sich in jeder Zeile eine ganze Bemerkung.

Die unterschiedlichen Formatierungen führen zu der Notwendigkeit, die Texte vor der Weiterverarbeitung in eine einheitliche, vom verwendeten Programm verwertbare Form zu bringen.

Grundsätzlich ist der komplette Nachlass in einzelne Bemerkungen unterteilt, die als in sich geschlossene Einheiten angesehen werden. Es kann auftreten, dass ein Thema sich durch mehrere Bemerkungen zieht, es kommt aber auch oft vor, dass ein in einer Bemerkung behandeltes Thema sonst nirgendwo mehr vorkommt. Daher wird beispielsweise der in Kapitel 5 beschriebene Kontext mit dem Anfang sowie dem Ende einer Bemerkung begrenzt.

Diese einzelnen Bemerkungen sind entweder Teile von Manuskripten oder Typoskripten. Jede Bemerkung ist einzigartig gekennzeichnet, durch 'Ts' für Typoskripte oder 'Ms' für Manuskripte begleitet von Nummern zur Identifikation. Diese Kennzeichnungen werden in allen vorgestellten Beispielen als Referenz angegeben.

Hier ein Beispiel für eine besonders kurze, aber vollständige Bemerkung mit ihrer Bezeichnung:

Aus Ms-157a_59v:
„Mir gehts eigentlich jetzt gut; es scheint mir gelungen zu sein Gott zu täuschen.“

4.2 Arbeiten auf Grundlage eines Musiklexikons

Die Grundlage der neuen Kategorie in der semantischen Suchoption von WiTTFind Web ist ein Lexikon musikalischer Ausdrücke. Zu Beginn der Arbeit wurde ein Teil des Lexikons, von schon in Gruppen zusammengefassten Begriffen, als Basis für zukünftige Frequenzberechnungen und Wordclouddarstellungen ausgewählt.

Später wurden noch einzelne Begriffe der so entstandenen Listen verändert oder gestri-

chen, und decken so leider nicht jedes in einem musikalischen Sinn verwendete Wort im gesamten Wittgensteinkorpus ab.

Unter den Begriffen, welche nicht in die Auswahl gekommen sind, oder wieder gestrichen wurden, befanden sich zum Großteil schwer bis gar nicht disambiguierbare Begriffe.

Ein sehr gutes Beispiel für einen solchen Ausdruck ist 'c' womit als Musikbegriff der Ton c gemeint ist welcher unter anderem an der Stelle Ms-115,255[3] verwendet wird:

Ms-115,255[3]:

"Wenn Einer gefragt würde, warum er das den 'gleichen' Ton nennt, so würde er vielleicht antworten: "Es ist wieder ein c". Aber das ist nicht, was ich hören möchte, denn ich frage: "Warum nennt man diesen Ton wieder 'c'?" "

Allerdings gibt es nur einige wenige Stellen, in welchen mit 'c' wirklich der Ton gemeint ist, aber unzählige Stellen mit 'c' als Variable. Hier ein Beispiel für eine solche Stelle:

Ms-122_116v:

"Nun, man kann sagen: der Induktionsbeweis überzeugt uns davon, daß wir zu sagen haben $a + (b + c) = (a + b) + c$ & kommt das im besondern Fall nicht heraus, so haben wir einen Fehler anzunehmen."

Die Frequenz von 'c' ist extrem hoch, weil es so oft in einem mathematischen Zusammenhang verwendet wird. Da es nicht möglich war, diese Stellen zu disambiguieren, wurde der Ausdruck 'c' nicht mit bearbeitet, weil seine sehr hohe Häufigkeit, mit nur einem Bruchteil musikalischer Vorkommen, das Ergebnis verfälscht hätte.

4.3 Semantische Kategorien

Eine Einteilung der relevanten, musikalischen Begriffe in verschiedene Kategorien ist schon im Musiklexikon vorhanden. Diese Einteilung wurde vollständig übernommen. Die Tabelle 4.1 gibt einen Überblick darüber, welche Kategorien verwendet wurden und wie viele Begriffe in einer Kategorie enthalten sind.

Die Kategorie „Komponisten“ enthält die Namen 15 verschiedener Komponisten und somit nur Eigennamen. Die größte Kategorie ist „Sonstige Begriffe“. Die Wörter aus dieser Gruppe können zukünftig noch kategorisiert und auf andere Kategorien aufgeteilt werden. Vermutlich wird eine vollständige Aufteilung der Begriffe in dieser Kategorie nicht möglich sein, da es nicht lohnenswert ist, für alleinstehende Begriffe eine eigene Kategorie zu schaffen. Es gibt sehr viele alleinstehende Begriffe, denn obwohl in vielen Fällen zahlreiche musikalische Ausdrücke zu einer Klasse existieren, finden viele davon in Wittgensteins Nachlass keine Erwähnung. Ein gutes Beispiel hierfür ist die Kategorie „Intervalle“. Obwohl es eine große Anzahl von Intervallen gibt, erwähnt Wittgenstein in den bisher gesichteten Dokumenten nur drei davon.

| Kategorie | Anzahl der Begriffe |
|----------------------|---------------------|
| Komponisten | 15 |
| Gattungen | 6 |
| Instrumente | 15 |
| Intervalle | 3 |
| Bezug zu Komposition | 6 |
| Sonstige Begriffe | 68 |

Tabelle 4.1: Verteilung der musikalischen Begriffe auf ihre Kategorien

Die folgende Tabelle 4.2 enthält Beispiele für Ausdrücke, die in den einzelnen Kategorien enthalten sind:

| Kategorie | Beispielausdruck |
|----------------------|-------------------------------------|
| Komponisten | Mendelssohn, Schubert, Bruckner |
| Instrumente | Pianola, Klavier, Orgel |
| Gattungen | Fuge, Walzer, Suite |
| Intervalle | Oktav Quint, fifth |
| Bezug zu Komposition | Meistersinger, Nothung, Blaue Donau |
| Sonstige Begriffe | singen, Klangfarbe, Tonfolge |

Tabelle 4.2: Kategorien mit Beispielausdruck

Diese Liste der Kategorien sollte zu einem späteren Zeitpunkt, vor allem wenn neue Musikbegriffe hinzukommen, noch erweitert werden. Neue Kategorien würden auch eine feinere Klassifizierung der bisherigen Kategorien, vor allem von 'Sonstige Begriffe' überhaupt erst möglich machen.

4.4 Wordclouddarstellung der musikalischen Begriffe

Wie auch bei den bisherigen Suchoptionen werden Begriffe in Wordclouds dargestellt. Das sind Grafiken, in welchen die Wörter wie durcheinandergemischt angeordnet wiedergegeben werden. Je höher die Frequenz eines Wortes ist, desto größer wird es abgebildet. So entsteht eine Darstellung, in welcher intuitiv auf den ersten Blick erkennbar ist, welche Wörter am häufigsten in einer Kategorie vorkommen.

In Abbildung 4.1 lässt sich sehr schön erkennen, dass die Wörter als „Wolke“ abgebildet werden. In dieser Wordcloud ist „Schubert“ ein Wort mit hoher Frequenz, durch die verhältnismäßig große Darstellung springt es sofort ins Auge. Je kleiner ein Wort, desto niedriger die Frequenz, so hat hier „Bruckner“ eine niedrigere Frequenz als „Schubert“, aber höher als „Bach“. Eine Funktion der Wordcloud ist außerdem, dass durch Klicken auf ein abgebildetes Wort dieses in die Suchleiste geschrieben wird, um so eine direkte Suche zu ermöglichen.

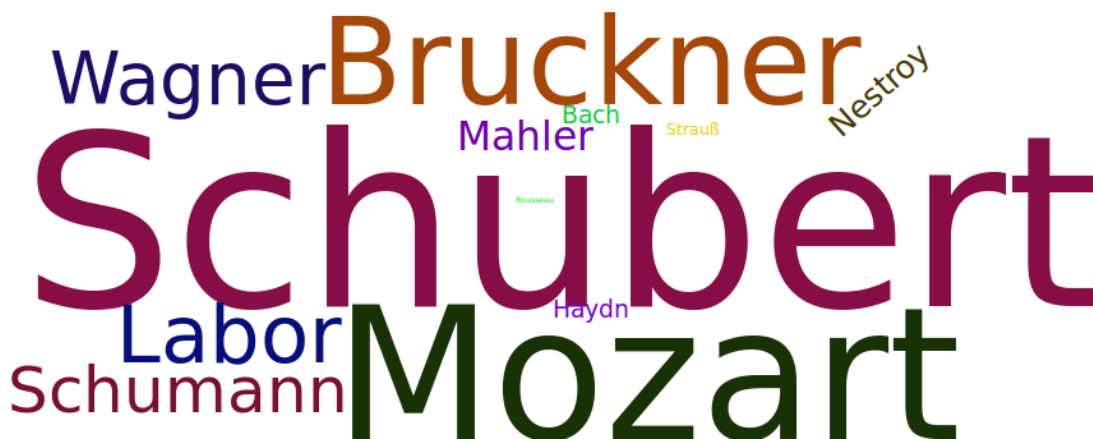


Abbildung 4.1: Beispiel für eine Wordclouddarstellung

Diese Wordclouds werden mit einem Javascript Programm erstellt, welches dazu eine Frequenzliste der gewünschten Wörter benötigt. Auch die bisher existierenden semantischen Klassen nutzen diese Funktion, daher konnte dieser schon bestehende Programmteil für die zusätzlichen Wordclouddarstellungen wiederbenutzt werden.

Auch der in Abschnitt 5 beschriebene Kontext wird als Wordcloud dargestellt. Ein Beispiel für eine Kontextwordcloud ist 4.2. Vor allem bei diesen Kontextwordclouds tritt teilweise durch die Menge der Wörter, die abgebildet werden sollen, der Fehler auf, dass nicht alle Wörter in der Wordcloud dargestellt werden. Auch in der nachstehenden Kontextwordcloud hätte es sehr viel mehr abzubildende Wörter gegeben. Dieser Fehler tritt vor allem dann auf, wenn die Frequenzen der Wörter sehr hohe Differenzen aufweisen. Oft wäre dann ein abzubildendes Wort so viel größer oder kleiner, um sein Verhältnis zu den anderen Wörtern darzustellen, dass es nicht abgebildet werden kann.

Dieser Fehler kann in Zukunft sicher behoben werden, indem eine Funktion eingebaut wird, welche die Häufigkeiten für die Wordclouddarstellung so anpasst, dass die Wortgrößen darstellbar werden.



Abbildung 4.2: Beispiel für eine Kontextwordcloud

4.5 Frequenzberechnungen der relevanten Begriffe

Die Frequenzen der Musikbegriffe, welche die auf der Webseite genutzten Frequenzlisten bilden, werden mit einem Python-Skript berechnet. Hierzu gibt es zu jeder Kategorie jeweils eine Textdatei, welche die Musikbegriffe enthält. Diese sind so abgespeichert, dass jede Zeile genau einen Begriff beinhaltet, damit leicht eingelesen werden kann. Im Python-Skript wird ein Dictionary initialisiert um die Begriffe einzulesen. Das Programm liest die Wörter aus den Dateien so in das Dictionary ein, dass das Wort als 'Key' abgespeichert wird, und sein 'Value' aus seiner Frequenz und seinem Herkunftsdateinamen besteht. Das Hinterlegen des Dateinamens zu jedem Wort ist deshalb nötig, damit es möglich ist, später die Worte wieder den einzelnen Kategorien zuzuordnen. Diese Methode wurde ausgewählt, weil es mit ihr nur einmal nötig ist, den Korpus zu durchlaufen.

Nach dem Abspeichern der Begriffe im Dictionary wird der Text durchlaufen und die Frequenz jedes Musikbegriffes hochgezählt. Dabei wird die in 4.6 vorgestellte Methode der Vollformenextraktion angewandt. Gleichzeitig wird von den gefundenen relevanten Ausdrücken der Kontext, wie in Abschnitt 5.2 gezeigt, mithilfe von Listenoperationen extrahiert. Diese Vorgehensweise erfordert nur einen Textdurchlauf, in welchem alle Operationen durchgeführt werden, und ist somit sehr effizient bei einer geringen Ausführungsdauer.

Abbildung 4.3 stellt schematisch dar, wie die einzelnen Wörter in dem Frequenzdictionary abgespeichert sind. Man sieht links die Wörter, welche die 'Keys' sind, und recht ihre 'Values'. Die Values bestehen aus ihrem ursprünglichen Dateinamen und ihrer gezählten Frequenz. Durch den Dateinamen kann später aus allen Wörtern, die denselben Dateinamen als Key haben, direkt eine Textdatei mit den extrahierten Frequenzen erstellt werden.

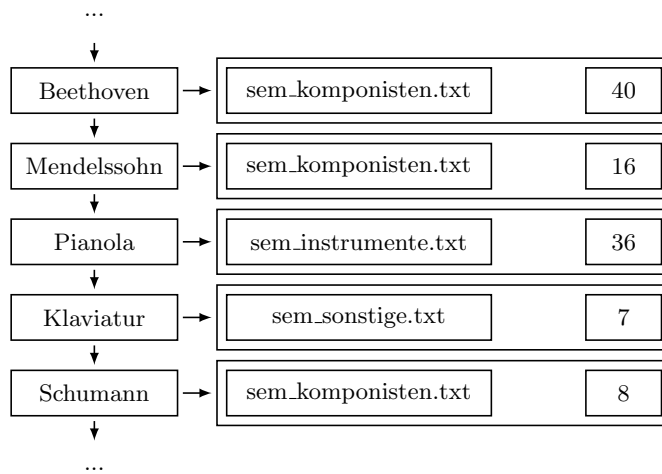


Abbildung 4.3: Beispiel für das Frequenzdictionary

4.6 Finden der Vollformen

Die Wortlisten, welche die relevanten Wörter enthalten, bestehen zum Großteil nur aus Lemmata. Das bedeutet, dass im Text vorkommende Vollformen nicht automatisch gefunden werden. In den meisten Fällen könnte man das durch eine Suche umgehen, bei welcher auch relevante Wörter gefunden werden, wenn sie nur im Suchergebnis enthalten sind. Das zeigt dieses Beispiel:

Aus Ms-183_106:

*Die Themen schon sind schwarz-weiß, wie die **Brucknerschen** schon färbig; auch wenn er Bruckner sie tatsächlich aus irgendeinem Grund auf nur einem System niedergeschrieben hätte, so daß wir von einer **Brucknerschen** Instrumentierung nichts wüßten. [...]*

*Ich glaube nur daß gerade hier die Schwäche der **Brahmschen** Instrumentation liegt, indem sie nämlich vielfach doch nicht ausgesprochen schwarz-weiß ist.*

„Bruckner“ wäre im Vorkommen „Brucknerschen“ enthalten, ebenso wie „Brahms“ in „Brahmschen“. Wäre die Suche so gestaltet, dass auch jedes Wort gefunden wird, welches einen relevanten Begriff enthält, wären diese Fälle sofort detektierbar. Eine solche Vorgehensweise funktioniert allerdings nur bei Begriffen, welche nicht oder nur sehr selten als normaler Teil anderer Worte auftreten.

Die folgenden Beispiele illustrieren das auftretende Problem:

Beispiel 1:

Aus Ts-303.7:

*„Denken wir uns folgenden Fall: In einer Gesellschaft von Menschen sei der eine A besonders geschickt darin, **Tongefäße** zu machen; die andern können das nicht, können aber welche zeichnen.“*

Beispiel 2:

Aus Ts-230b.4:

*„Wenn Einer die **Notenschrift** lernt, so wird ihm eine Art Grammatik beigebracht. Es heißt da: Diese **Note** entspricht dieser Taste am Klavier, das Kreuz erhöht einen Ton, [...], so würden wir ihm sagen, daß die Entfernung des **Notenkopfes** von den Linien nichts ausdrücke, u.s.f.“*

In Fällen wie dem von Beispiel 1 werden auch Wörter gefunden, die überhaupt keinen musikalischen Bezug haben. Hier würde jedes Vorkommen von „Tongefäße“ die Frequenz von „Ton“ unnötig erhöhen und somit verfälschen, was unbedingt zu vermeiden ist.

Beispiel 2 zeigt einen im Grunde noch gravierenderen Fall. Alle drei hervorgehobenen Wörter sind relevant, so wird von jedem auch die Frequenz erfasst. Allerdings würde allein durch diesen Satz die Frequenz von „Note“ um drei erhöht werden, wobei die Frequenzen von „Notenkopf“ und „Notenschrift“ dennoch je um eins erhöht werden. Somit werden die beiden letztgenannten Wörter doppelt gezählt, weil „Note“ in ihnen enthalten ist. Ähnliche zusammengesetzte Wörter kommen sehr oft vor, ein anderes Beispiel wäre „Klavier“ und „Klavierspiel“.

Aus den oben beschriebenen Gründen ist eine solche Suche nicht anwendbar. Um die verschiedenen Vollformen der relevanten Wörter dennoch zu finden, lohnt es sich vor allem einen Blick auf die Wortendungen zu werfen.

Da zu Beginn der Arbeit noch kein Lexikon mit Vollformen und ihren Lemmata, wie das `witt_WAB_dela` Lexikon, verfügbar war, wurde zunächst mit einer primitiven Liste von möglichen Wortendungen gearbeitet. Mit ihr überprüft das Programm für jedes Wort, falls dieses nicht relevant ist, ob es eine in der Liste vorkommende Endung hat. Ist dies der Fall, wird überprüft ob wiederum der Ausdruck ohne diese Endung ein relevantes Wort ist. Die Trefferquote wurde hierdurch immens erhöht. Diese Vorgehensweise stellte sich im Laufe der Arbeit besser heraus als anfänglich gedacht.

Zu einem späteren Zeitpunkt wurde, an diese Arbeit anknüpfend, mit dem Lexikon `witt_WAB_dela_XVI` gearbeitet. Dieses Lexikon ist nach dem, von Paumier (2003) beschriebenen, DELAF Format strukturiert. In diesem Format wird pro Zeile je eine Vollform oder ein Lemma eines Wortes definiert. Die Zeile beginnt mit dem beschriebenen Wort, gefolgt von einem Komma. Falls eine Vollform beschrieben wird, steht nach dem Komma das Lemma, wird ein Lemma beschrieben bleibt diese Position leer. Danach wird durch Abkürzungen beschrieben, in welcher Form das Wort vorliegt, welchen Kasus und Genus es demnach hat. Hinter einem solchen Eintrag können Kommentare nach einem `'/'` eingefügt sein.

Leider ist das Lexikon ziemlich unvollständig. Viele der musikalischen Begriffe sind nicht enthalten, nur das Lemma ist enthalten, oder nur Vollformen, ohne das Lemma oder ohne Bezug zu ihrem Lemma.

Gar nicht enthalten sind diese Begriffe:

'c-dur', 'Notenkopf', 'Notensprache', 'rhythm', 'Meistersinger', '8.',
'Beethoven', 'Bruckner', 'Labor', 'Mahler', 'Schumann', 'Wagner'

Wie zu sehen ist, ist 'Bruckner' nicht im Lexikon enthalten. Dabei sind für zwei Vollformen von 'Bruckner' sogar Einträge enthalten:

Bruckners,.ADJ+EN
Brucknersche,.ADJ+EN

Diese sind allerdings nicht auf ihr Lemma bezogen. Da trotzdem sehr viele Begriffe im Lexikon enthalten sind, wurde eine Lexikonüberprüfung implementiert. Dazu wird das Lexikon eingelesen und im Python Programm in einem 'Dictionary' gespeichert. Hier werden die Vollformen als 'Keys', sowie die Lemmata als 'Values' verwendet. Nun wird für jedes Wort, falls es kein relevantes Wort ist, überprüft, ob es als Key im Lexikon existiert. Ist dies der Fall, wird sein Lemma, also der Value, zurückgegeben. Die Verwendung des Lexikons erbringt im Vergleich zu keiner Vollformensuche eine deutliche Verbesserung. Gerade bei Verben, wie „klingen“, „hören“ oder „pfeifen“, leistet das Lexikon gute Arbeit indem es alle Vollformen zuordnet.

Leider ist vor allem für die Kategorie der Komponisten das Lexikon nicht hilfreich, es kann keine einzige Form zuordnen. Die mit der Suffixlistenmethode erzielten Verbesserungen in der Frequenz führten dazu, dass diese, eigentlich als Provisorium gedachte Suchmethode, dennoch bis zum Schluß im Programm enthalten blieb. Zukünftig, wenn das Lexikon erweitert und verbessert ist, kann diese Methode sicherlich entfernt werden, da sie dann von Lexikonarbeit ersetzt wird.

Ein erster Schritt in der Verbesserung des Lexikons könnte sein, dass für jede schon vorhandene Vollform auch ihr jeweiliges Lemma enthalten ist, das würde Fälle, wie den im Vorigen gezeigten für 'Bruckner', beseitigen. Anschließend könnten mehr Eigennamen hinzugefügt, beziehungsweise das Lexikon auf einem sehr viel größeren Korpus angefertigt werden.

Da die Lexikonoperationen schon im Programm vorhanden sind, könnte ein solches neues witt_WAB_dela Lexikon sofort eingespeist und verwendet werden.

4.7 Probleme

Dieser Abschnitt erläutert Probleme, die speziell während der Arbeit mit der semantischen Suche aufgetreten sind, ob und wie die Untersuchungen beeinträchtigt wurden und zeigt mögliche Lösungen auf.

4.7.1 Leerzeichenfehler

Ein großes, allgemeines Problem der verfügbaren digitalen Wittgensteinkorpora ist die relativ große Anzahl an Editionsfehlern. Dazu gehören beispielsweise fehlende Leerzeichen zwischen zwei Wörtern, welche das Erstellen von Frequenzlisten stark behindern können. Diese Leerzeichenfehler sind nicht in jeder Korpusversion gleich häufig. In den zu Beginn der Untersuchungen verwendeten Korpora traten diese Fehler so extrem häufig auf, dass sie die Frequenzlisten stark beeinflusst haben, wie im folgenden Beispiel veranschaulicht:

Aus Ts-310,139[3]:

*Consider also this expression: "Tell yourself that it's **awaltz**, and you will play it correctly."*

Hier wird das Wort „waltz“ nicht gefunden, obwohl es relevant wäre, weil „awaltz“ natürlich nicht in der Liste relevanter Begriffe enthalten ist. Das am Ende verwendete Korpus hat nur noch sehr vereinzelte Leerzeichenfehler. Hier ein weiteres Beispiel für ein solches Manko:

Aus Ms-183_10:

*Heute zeigte mir Mrs Moore eine dumme Kritik einer Aufführung der 4ten Symphonie von Bruckner wo der Kritiker über Bruckner schimpft & auch von Brahms & Wagner **respe ktlos** redet.*

Das hier gezeigte, falsch gesetzte Leerzeichen führt zu dem Fehler, dass „respe“ und „ktlos“, wie in Kapitel 5 genauer ausgeführt, als 'Wörter' in den Kontext aufgenommen werden. Da diese Makel allerdings in dem jetzt verwendeten Korpus nur noch äußerst selten auftreten, sind sie vernachlässigbar und beeinträchtigen die Frequenzlisten nicht nachhaltig.

4.7.2 Multiple Satzvorkommen

Ein weiteres Phänomen, welches beim Bearbeiten der einzelnen Wortvorkommen auffiel, ist, dass sehr viele Sätze doppelt oder sogar mehrmals im Korpus vorhanden sind. Dadurch

vervielfacht sich die Frequenz der Wörter, die in solchen multiplen Vorkommen enthalten sind. Dabei stellt sich die Frage, ob diese Frequenzerhöhungen als Ungenauigkeiten zu sehen sind oder nicht.

Um das zu beantworten, müssen die Gründe für diese mehrfachen Satzvorkommen betrachtet werden. Einerseits ist es so, dass Ludwig Wittgenstein auf über 20.000 Seiten seine Gedanken über Jahrzehnte weiterentwickelt hat. Somit kann es durchaus vorkommen, dass ein Satz von ihm selbst wieder aufgenommen und niedergeschrieben wurde, weil er mit ihm etwas aussagen wollte. Daher sind einige dieser Doppelungen sogar von Wittgenstein selbst hervorgerufen. Hier lässt sich sagen, dass durch eine solche Doppelung, durch Wiederverwendung eines Gedankens, auch die Wichtigkeit der genutzten Begriffe steigt, wodurch eine erhöhte Frequenz sogar gut ist, um das betreffende Wort hervorzuheben. Andererseits treten mehrfache Satzvorkommen auch immer dann auf, wenn in der Edition verschiedene Optionen für die Schreibweise eines Satzes vorhanden sind. Das ist immer dann der Fall, wenn Wittgenstein in seinem Werk mit verschiedenen Wörtern gearbeitet hat, beispielsweise hat er ein zweites Wort über ein im Satz vorhandenes Wort geschrieben. Ein Beispiel:

Aus Ms-154,85r[2]:

"Man kann auch nicht sagen ich beweise eine Gleichung wenn ich drei beweise.

*Wie die Sätze einer **Sonate** nicht einen Satz ergeben."*

"Man kann auch nicht sagen ich beweise eine Gleichung wenn ich drei beweise.

*Wie die Sätze einer **Suite** nicht einen Satz ergeben."*

Hier sind 'Sonate' und 'Suite' beides Optionen für die Verwendung im Satz. In einem solchen Fall kommt der Satz in der Edition auch doppelt vor, um beide Varianten anzugeben. Auch hier ist das doppelte Vorkommen auch in der Frequenz gerechtfertigt, da Ludwig Wittgenstein schließlich auch beide Varianten des Satzes gedacht hat. Somit wird durch die höhere Frequenz auch die Wichtigkeit, der in solchen mehrfach enthaltenen Sätzen vorkommenden Wörter, betont.

5 Kontext musikalischer Begriffe

Im Kapitel „Kontext musikalischer Begriffe“ wird genauer eingegangen auf Motivation und Umsetzung einer Kontextdarstellung, und welche Schritte in der Implementierung dazu nötig waren. Anschließend wird durch Beispiele erläutert, inwiefern ein Kontext semantisch bedeutungsvoll sein kann.

5.1 Darstellung und Relevanz von Kontext

Auf der Webseite werden zu den verschiedenen musikalischen Kategorien jeweils die häufigsten, im Kontext der Begriffe der einzelnen Kategorien vorkommenden Wörter angezeigt. Der Kontext eines Wortes kann dessen Bedeutung unterstreichen, genauer definieren oder auch eine spezielle, nur in diesem Kontext gültige Bedeutung zuweisen.

Wörter, die in einer semantischen Relation stehen, treten im Text häufig innerhalb sehr kurzer Distanzen zueinander auf. Kommt also ein Wort sehr oft im unmittelbaren Kontext eines anderen Wortes vor, steigt die Wahrscheinlichkeit einer semantischen Korrelation dieser Wörter. Ein ähnliches Konzept wird für Word Embedding Systeme verwendet. In diesen Systemen werden meist zwischen oft zusammen auftretenden Wörtern Vektoren gezogen, welche semantische Relationen anzeigen.

Da viele der behandelten musikalischen Begriffe nur mit sehr niedriger Frequenz im Text vorkommen - z.B. tritt „Schubert“ im OpenSource Teil des verwendeten Korpus nur einmal auf - ist es nicht sinnvoll für jeden einzelnen Begriff dessen Kontext zu extrahieren, da dieser nicht sehr inhaltsreich wäre. Darum wird eine gemeinsame Kontextfrequenzliste aus den Frequenzlisten der einzelnen Kontexte der Begriffe jeder Kategorie erstellt, da diese sehr viel aussagekräftiger sind.

Die Visualisierung erfolgt, wie bei den Häufigkeiten der Begriffe selbst, durch eine Wordcloud und eine Frequenztafel, welche sich durch ein Klicken auf den Link des gewünschten Kontextes anzeigen lassen.

5.2 Berechnung des Kontextes

Um für jede Kategorie eine Frequenzliste mit den häufigsten im Kontext vorkommenden Wörtern zu bekommen, wird zunächst für jedes Vorkommen eines musikalischen Begriffs der Kontext extrahiert. Dieser Kontext besteht aus jeweils fünf Wörtern vor und nach dem relevanten Wort. Jeder dieser Einzelkontexte wird in eine nach Kategorien getrennte Frequenzliste gespeichert, wovon am Ende die 20 häufigsten Kontextwörter dargestellt werden.

Auf die genaue Umsetzung der Implementierung wird in den folgenden Unterpunkten eingegangen.

5.2.1 Stopwörter und Ausschlussliste

Stopwörter werden definiert als die meistverwendeten Wörter einer Sprache, die für die inhaltliche Analyse eines Textes keine Bedeutung haben. [Zitat!] In den meisten Frequenz-

darstellungen sind Stopwörter daher unerwünscht. Weil es für einen Kontext aber auch durchaus interessant sein kann, in welchem Verhältnis häufig auftretende Stopwörter und Nicht-Stopwörter zueinander stehen, fiel die Entscheidung, beide Varianten für den Endanwender zur Verfügung zu stellen.

Zum Herausfiltern der Stopwörter aus einer Variante des Kontextes wurden mehrere vorgefertigte Stopwortlisten in Betracht gezogen. Letztendlich stellte sich eine Liste ¹ mit sehr vielen verschiedenen Vollformen der einzelnen Stopwörter als die Beste heraus. Sie wird, mit einigen kleinen Anpassungen, in der Endversion verwendet.

Als kleine Anpassung sind einige Wörter, die als sinntragend eingeschätzt wurden, aus der Liste entfernt worden. Zu diesen Wörtern zählten beispielsweise: „groß“, „besonders“, „besser“ und „übel“.

Interessant ist die Frage, ob das Wort „Ich“, welches normalerweise ein sehr klassisches Stopwort ist, im Kontext von L. Wittgensteins Nachlass nicht doch sinntragend ist, mit der Bedeutung, dass L. Wittgenstein von sich selbst spricht. Gerade weil es um Musik geht, und es interessant ist, welchen Bezug L. Wittgenstein selbst zur Musik hatte, wurde entschieden auch „Ich“ aus der Stopwortliste zu entfernen.

5.2.2 Verwendete Methoden

Auf dem Weg zum endgültigen Programm wurden mehrere Methoden angedacht, getestet und auch wieder verworfen. In diesem Abschnitt werden zwei Methoden vorgestellt. Die fertiggestellte Variante mit der Verwendung eines Ringbuffers, welche anschließend doch ersetzt wurde durch die zweite, simplere Methode in welcher eine einfache Liste genutzt wird.

Ringbuffer

Ursprünglich schien die Verwendung eines Ringbuffers sich als beste Methode für die Kontextextraktion anzubieten. Ein Ringbuffer ist eine Datenstruktur, welche funktioniert wie eine Warteschlange mit fester Größe, in der Anfang und Ende verbunden sind. Sobald jeder Platz im Ringbuffer gefüllt ist, überschreiben neu hinzugefügte Elemente die Ältesten. So entsteht eine Ringform, da die verwendeten Pointer sobald sie am Ende angekommen sind, den Ringbuffer wieder vom Anfang durchlaufen.

Um mit einem Ringbuffer arbeiten zu können, war die Implementierung einer eigenen Objektklasse erforderlich. Die Umsetzung dieser Klassenimplementierung kann man im folgenden Code-Ausschnitt sehen:

```
1 class ringbuffer(object):
2     def __init__(self, length=101):
3         self.w_idx = 0
4         self.r_idx = 0
5         self.l = [None] * length
6         self.length = length
7
8         # Funktion um Woerter in den ringbuffer einzufuegen
9         def insert(self, obj):
10            self.l[self.w_idx] = obj
11            if self.w_idx < self.length - 1:
12                self.w_idx += 1
13            else:
```

¹Siehe https://github.com/solariz/german_stopwords/blob/master/german_stopwords_plain.txt

```

14         self.w_idx = 0
15
16     # Funktion um Woerter aus ringbuffer auszulesen
17     def get(self):
18         temp = self.l[self.r_idx]
19         if self.r_idx < self.length - 1:
20             self.r_idx += 1
21         else:
22             self.r_idx = 0
23         return temp
24
25     # Funktion um den kompletten ringbuffer (Kontext) zurueckzugeben
26     def returnbuffer(self):
27         return self.l

```

In diesem Ringbuffer wurden zwei Pointer verwendet, ein Lese- und ein Schreibpointer. Der Schreibpointer fügt immer neue Elemente hinzu, sodass bei einer Größe von insgesamt 51 Elementen immer die 51 letzten Worte im Ringbuffer sind. Der Schreibpointer befindet sich in diesem Fall immer genau 25 Elemente vor dem Lesepointer.

Zeigt der Lesepointer auf ein relevantes Wort, enthält der Ringbuffer durch die Positionen der Pointer genau 25 Elemente je vor und nach dem Wort. Diese Wörter sind der Kontext des relevanten Ausdrucks und werden in einer Frequenzliste abgespeichert. Da jedes Wort einer Kategorie zugehörig ist, und es für diese je eine Frequenzliste gibt entsteht so eine Sammlung der häufigsten Kontextwörter jeder Kategorie.

Abbildung 5.1 ist eine Veranschaulichung eines Ringbuffers. Hier lässt sich gut erkennen, wie Lese- und Schreibpointer auf „gegenüberliegende“ Elemente zeigen. Der grün gefärbte Bereich markiert noch nicht ausgelesene Worte, alle Elemente im weißen Halbkreis wurden schon ausgelesen. Der Schreibpointer zeigt auf das älteste Element, welches er als nächstes mit einem neu eingefügten Wort überschreibt.

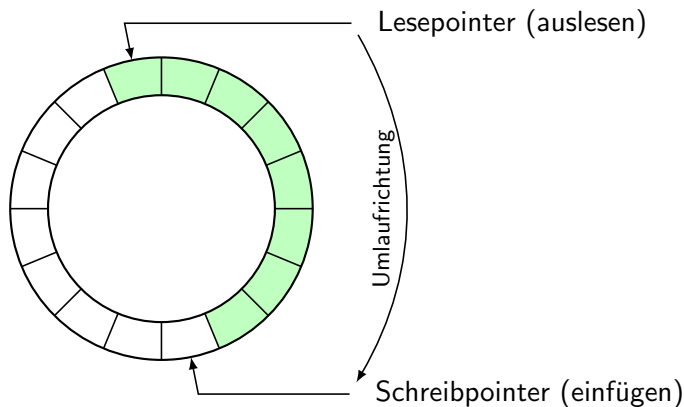


Abbildung 5.1: Veranschaulichung eines Ringbuffers

Der Ringbuffer arbeitet nach dem „FIFO“ Prinzip², also „First in, first out“. Die zuerst eingefügten Elemente, sind auch die ersten, wieder überschriebenen Elemente. Deshalb ist der Kontext immer aktuell, da Wörter mit zu großem Abstand zum relevanten Wort nicht mehr im Ringbuffer enthalten sind.

Auch an verschiedene Kontextversionen ohne Wörter einer Ausschlussliste, bzw. mit und ohne Stopwörter, ist ein Ringbuffer anpassbar. Dies ist am einfachsten lösbar, indem

²Dan and Towsley (1990), S. 145

man für jedes Wort beim Schreiben überprüft, ob es in einer Ausschluss- oder Stopwortliste vorhanden ist, nur wenn das nicht der Fall ist, wird es hinzugefügt.

Ein ausschlaggebender Nachteil der Ringbuffer Methode ist, dass der Kontext für jedes Wort im Text gespeichert wird, unabhängig davon, ob er gebraucht wird oder nicht. Zusätzlich muss der Kontext nur für sehr wenige Wörter im Vergleich zur Korpusgröße extrahiert werden.

Das größte Manko tritt bei der Extraktion verschiedener Varianten des Kontextes, beispielsweise mit und ohne Stopwörter auf. Für jede gewollte Variante ist es nötig einen eigenen Ringbuffer zu erstellen und zu verwenden. Das kommt daher, dass die Entscheidung, ob ein Wort im Kontext enthalten sein soll, bereits vor dem Schreiben in den Ringbuffer getroffen werden muss. So wird bei einer Verwendung von zwei Ringbuffern, einer mit, sowie einer ohne Stopwörtern, der Rechenaufwand signifikant erhöht.

Liste

In der Endversion des Programmes ist eine nur auf Listenoperationen basierende Umsetzung implementiert. Hier überprüft eine Abfrage für jedes Wort in einer Bemerkung, ob es ein relevantes Wort ist. Ist das der Fall, extrahiert eine Funktion, aus der als Liste gespeicherten Bemerkung, den Kontext des Wortes.

Diese Art der Extraktion ist sehr leicht anpassbar an verschiedene Kontextvarianten, da bei dem Listendurchlauf, welcher den Kontext erzeugt, nicht benötigte Wörter einfach übersprungen werden können. Dadurch ist man nicht auf einen Wortabstand festgelegt, sondern dieser verändert sich dynamisch, da die Funktion so lange Wörter einliest, bis der benötigte Kontextumfang oder das Ende der Bemerkung erreicht ist.

Die Funktion innerhalb des Programms, welche den Kontext erzeugt, sieht folgendermaßen aus:

```
1 def make_context(splitline , index , excludelist , direction):
2     count = 0
3     distance = 1
4     ctx = []
5
6     while count < 5:
7         elem = splitline[index + distance * direction]
8         if (elem not in excludelist and not elem.isdigit()
9             and len(elem) >= 2 and not elem in stopwordlist):
10            ctx.append(elem)
11            count += 1
12            distance += 1
13            if (distance + index == len(splitline)
14                or index - distance < 0):
15                break
16    return ctx
```

Diese Methode extrahiert anhand der übergebenen Bemerkung, dem Index des aktuellen, relevanten Wortes, einer Ausschlussliste und einer Richtung fünf valide Wörter zur linken oder rechten Seite des relevanten Wortes. Die Richtung bestimmt, ob der linke oder rechte Kontext ausgegeben wird. Über die übergebene Ausschlussliste kann u.a. bestimmt werden, ob Stopwörter im Kontext enthalten sind oder nicht.

Beide so erstellte Kontextseiten fügt einer Methode „save_context“ zu einem zusammen, und speist den vollständigen Inhalt auch direkt in die Kontextfrequenzliste ein:

```
1 def save_context(splitline , index , excludelist , context):
```

```

2   ctx = make_context(splitline , index , excludelist , 1)
3   ctx += make_context(splitline , index , excludelist , -1)
4
5   for elem in ctx:
6       if elem in context[freq[sliced_currword][0]]:
7           context[freq[sliced_currword][0]][elem] += 1
8       else:
9           context[freq[sliced_currword][0]][elem] = 1

```

Grafik 5.2 ist eine Veranschaulichung der Kontextextraktion ohne Stopwörter mit den soeben vorgestellten Listenoperationen. Das Wort 'Orchester' wird aus dem eingelesenen Satz als relevantes Wort erkannt. Daraufhin werden die Wörter rechts und links von 'Orchester' durchlaufen und zur Kontextliste hinzugefügt, bis dieser jeweils fünf Wörter zu jeder Seite enthält. Dabei ist zu bemerken, dass sich das relevante Wort, hier 'Orchester', eigentlich nicht selbst im Kontext befindet, und hier lediglich als symbolischer Platzhalter in der Liste dargestellt wird. Das verwendete Satzbeispiel ist aus Ms-183_77.

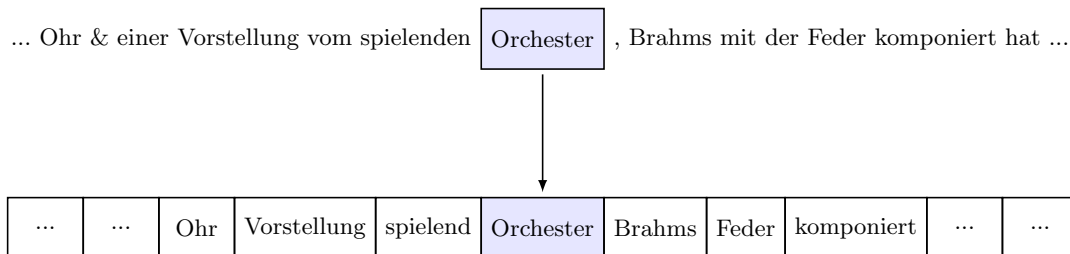


Abbildung 5.2: Listenoperationen

Die beigelegte CD enthält den geschriebenen Programmcode dieser Methode.

Vergleich beider Methoden

Der Vergleich beider Implementierungen mit dem Python-internen Profiler „cProfile“ bestätigt, dass die Verwendung einer Liste effizienter ist. In der Tabelle 5.1 sind die Werte der cProfile Durchläufe abgebildet. Hierbei sind „Primitive Calls“ nur direkte Aufrufe, wobei die Kategorie „Function Calls“ auch rekursive Aufrufe berücksichtigt.

Zusätzlich zu den äquivalenten Implementierungen, die Liste und der Ringbuffer mit je beiden Varianten, mit und ohne Stopwörter, enthält die Tabelle die Werte für eine Version des Ringbuffers, welche nur die Variante des Kontextes mit Stopwörtern extrahiert.

Die Werte zeigen deutlich, dass die Ausführungsdauer mit einer Liste am geringsten ausfällt. Bei Betrachtung beider Ringbufferwerte fällt auf, wie enorm viel höher die Ausführungsdauer in der Ringbuffer Implementierung ist. Das liegt in erster Linie daran, dass ein vollständiger zweiter Ringbuffer erstellt wird, sollten beide Kontextvarianten berechnet werden.

Der große Unterschied zwischen den äquivalenten Listen und Ringbuffer Methoden kommt vor allem daher, dass bei der Liste nur im Falle eines gefundenen relevanten Wortes für die umliegenden Wörter überprüft wird, ob sie Stopwörter sind. Beim Ringbuffer wird diese Überprüfung von vornherein für jedes Wort noch vor dem Einfügen in den Ringbuffer vorgenommen.

Die große Differenz in der Ausführungsdauer und die Schwierigkeit, verschiedene Varianten mithilfe eines Ringbuffers zu verarbeiten, sind die Hauptargumente für die Verwendung der Listenimplementierung.

| Methoden | Function Calls | Primitive Calls | Dauer (s) |
|---------------------------|----------------|-----------------|-----------|
| Liste | 20659862 | 20659859 | 11.244 |
| Ringbuffer | 56545669 | 56545666 | 44.713 |
| Ringbuffer (+ Stopwörter) | 32356932 | 32356929 | 15.204 |

Tabelle 5.1: Vergleich der Implementierungen mit cProfile

5.2.3 Auswahl des Kontextumfangs

Bei der Extraktion des Kontextes stellt sich automatisch die Frage, wie groß ein relevanter Kontext ist, also wie viele Elemente zur linken und rechten Seite des Wortes betrachtet werden sollten.

Ein zu groß gewählter Kontext birgt die Gefahr, dass vorkommende Wörter schon zu einem ganz anderen Thema gehören und daher nicht mehr semantisch mit dem Ausgangswort korreliert sind. Wählt man den Umfang jedoch zu klein, lässt man gegebenenfalls wichtige, bedeutungstragende Wörter außen vor.

Auch muss die Art des verarbeiteten Textes, sowie dessen Strukturierung, welche auch in Punkt 4.1 beschrieben wird, beachtet werden. Der bearbeitete Teil des Wittgensteinkorpus ist in einzelne Bemerkungen aufgeteilt. Viele dieser Abschnitte überschreiten eine Länge von 200 Wörtern nicht, einige sind nur 100 Wörter lang. Also ist die Verwendung eines Kontextes, welcher einen Umfang von 50 Wörtern zu jeder Seite, insgesamt 101 Elementen aufweist, viel zu groß. Der bedeutungstragende Teil des Textes, bezogen auf ein einzelnes Wortvorkommen, muss kleiner gefasst werden.

Das Maß des Tripels allerdings, welches sehr geläufig ist und oft als bedeutungstragendes Wortmaß verwendet wird, ist für diesen Zweck zu klein. Der literarischen Schreibweise Wittgensteins ist es geschuldet, dass das erste inhaltsreiche „Nicht-Stopwort“ vor oder nach einem relevanten Wort, oft bei weitem nicht das einzige, nicht einmal zwingend das semantisch am meisten korrelierte Wort ist.

Ein Beispiel:

Aus Ts-228_8:

*Was heißt es: Ich 'höre' in anderm Sinne: das **Klavier**, seinen Klang, das Musikstück, den Klavierspieler, seine Geläufigkeit?*

Extrahiert man nur die beiden sinntragenden Wörter links und rechts von „Klavier“, „Sinne“ und „Klang“, fehlen offensichtlich sehr viele weitere, sogar musikalische bedeutungstragende Wörter im Kontext.

Aus diesem Grund wurde der Kontextumfang auf je fünf Elemente vor und nach einem relevanten Ausdruck festgelegt. So wird sichergestellt, dass der Kontext nicht zu klein ist, sondern möglichst viele semantisch korrelierte Wörter enthält, aber auch nicht so groß ist, dass enthaltene Wörter schon einen anderen Themenbezug haben.

Ein Kontext muss allerdings nicht zwingend zehn Elemente umfassen, falls ein Wort weniger als fünf zu extrahierende Wörter von einer Bemerkungsgrenze entfernt ist, da der Kontext nicht über mehrere Bemerkungen greifen kann. Das ist so festgelegt, da die meisten Bemerkungen in sich geschlossene Abschnitte sind.

5.3 Kontextfrequenztabellen

Abschließend für dieses Kapitel sollen nun noch einige der erstellten Kontextfrequenztabellen vorgestellt werden. In Tabelle 5.2 sind exemplarisch aus drei verschiedenen Kategorien Auszüge aus den Kontextfrequenzen abgebildet.

Die hohe Frequenz von 'Musik' in der Komponistenkategorie spricht dafür, dass im Zusammenhang mit Komponisten besonders oft von ihrer Musik die Rede ist, teilweise aber auch von ihrem 'Charakter' denn dieser Begriff kommt auch zehnmal vor. Bemerkenswert ist auch, dass Komponisten anscheinend besonders häufig zusammen genannt werden, da 'Beethoven', 'Brahms' und 'Bruckner' alle in diesem Kontext mit zu den häufigsten Begriffen gehören. Auch der 'Klang' sowie ihre 'Werke' finden oft Erwähnung.

Obwohl die Klasse 'Intervalle' sehr klein ist, sind die häufigen Kontextworte dieser Kategorie besonders aussagekräftig. Alle drei in der Tabelle abgebildeten Ausdrücke - 'Ton', 'Töne' und 'Grundton' - sind sehr eng mit Intervallen verbunden.

Da in der Kategorie 'Sonstige Begriffe' mit Abstand am meisten Wörter enthält, sind hier auch die Kontextfrequenzen entsprechend höher. Es ist interessant, dass Ausdrücke wie 'Satz', 'Sprache' und 'Wort' sehr häufig im Zusammenhang mit vielen musikalischen Ausdrücken vorkommen. Dies spricht dafür, dass Sprache und Musik sehr oft von L. Wittgenstein verglichen werden. Aber auch die Begriffe 'Farbe' und 'Bild' sind sehr häufig im Kontext von Musikausdrücken zu finden.

| Kategorie | Wort | Frequenz |
|--------------------------|-------------|-----------------|
| Komponisten | Musik | 28 |
| | Beethoven | 18 |
| | Brahms | 14 |
| | Namen | 12 |
| | Charakter | 10 |
| | Bruckner | 10 |
| | Klang | 10 |
| | Werke | 9 |
| Intervalle | Ton | 13 |
| | Töne | 7 |
| | Grundton | 5 |
| Sonstige Begriffe | Satz | 1127 |
| | Form | 1023 |
| | Spiel | 601 |
| | Sprache | 595 |
| | Wort | 528 |
| | Farbe | 469 |
| | Bild | 458 |

Tabelle 5.2: Interessante Auszüge aus den Kontextfrequenzen

Hier kann nur ein kleiner Teil der Frequenzen gezeigt werden, aber alle weiteren Frequenzlisten befinden sich auf der beigelegten CD.

6 Verwendung von Ontologien für musikalische Relationen und Eigenschaften

Um die Beziehungen musikalischer Begriffe zu einander darzustellen, eignen sich Ontologien sehr gut. Mithilfe einer Ontologie können solche Relationen der Ausdrücke, Eigenschaften einzelner Begriffe und Gruppenzugehörigkeiten modelliert werden.

6.1 Musikalische Relationen und Eigenschaften

Die einzelnen behandelten Musikbegriffe haben untereinander vielfältige Beziehungen. Beispielsweise die Begriffe 'Meistersinger' und 'Wagner'. Hier ist 'Meistersinger' einerseits ein eigenständiges Musikstück und hat andererseits auch eine starke Beziehung zu Wagner, von welchem diese Oper geschrieben wurde. 'Wagner' währenddessen, hat die Eigenschaft, ein Komponist zu sein. Viele dieser Beziehungen und Eigenschaften können mit den schon bestehenden Musikkategorien ausgedrückt werden. Zum Beispiel könnte ein Teil einer Tabelle mit Beziehungen für die Meistersinger-Wagner Relation folgendermaßen aussehen:

| | ist Musikstück | ist Komponist | Bezug zu |
|---------------|----------------|---------------|----------|
| Meistersinger | 1 | 0 | Wagner |
| Wagner | 0 | 1 | - |

Tabelle 6.1: Meistersinger-Wagner Relation

Je mehr Musikbegriffe erfasst und verarbeitet werden, und vor allem wenn mehr verschiedene Relationen aufgezeigt werden sollen, desto höher wird der Bedarf nach mehr Kategorien. Auch die Einführung von Unterkategorien wird ein sinnvoller Schritt sein, die Musikbegriffe besser zu gliedern und zu strukturieren. Eine genauere Gliederung könnte beispielsweise eine Kategorie 'Gesang' ermöglichen. Wie in Tabelle 6.2 zu sehen, wären unter anderem 'Meistersinger' sowie 'Chor' in dieser Kategorie. Die Tabelle zeigt, dass sehr viele Ausdrücke Teil von mehr als einer Klasse sind.

'Trompetenstöße' sowie 'Klavierspiel' sind beides Ausdrücke, welche den Klang bzw. die Wirkung eines Instrumentes bezeichnen. Dadurch werden sie Teil der Kategorie Instrument, da ein Nutzer bei einer Suche nach Instrumenten auch Ergebnisse für Instrumentenklang angezeigt bekommen möchte. Beide Begriffe sind aber dennoch ebenfalls Teil der 'Sonstige' Kategorie, da eine Einteilung als 'Instrument' für eine vollständige Klassifizierung des Begriffes nicht ausreicht. Beide haben offensichtlich einen sehr starken Bezug zu ihrem jeweiligen Musikinstrument und werden diesem in der 'Bezug zu' Kategorie zugeordnet.

Wie im vorigen Absatz schon angedeutet, ist es sehr wichtig, bei der Einteilung der Begriffe darauf zu achten, welche Suchergebnisse ein Nutzer bei der Auswahl einer bestimmten Kategorie angezeigt bekommen möchte. Beziehungsweise auch andersherum, für jeden einzelnen Begriff muss klar sein, bei der Auswahl welcher Kategorie Textstellen, mit einem Vorkommen dieses Wortes erwartet werden. Ein Fall für eine solche nutzerorientierte Einteilung ist das, ebenfalls in Tabelle 6.2 enthaltene, Wort 'Geiger'. Obwohl 'Geiger' eigentlich kein Instrument ist, wird der Ausdruck in diese Klasse eingeteilt, da ein Nutzer vermutlich auch Textstellen in denen 'Geiger' vorkommt angezeigt haben möchte.

| | ist Musikstück | ist Gesang | ist Musikinstrument | ist Sonstige | Bezug zu |
|----------------|----------------|------------|---------------------|--------------|----------|
| Meistersinger | 1 | 1 | 0 | 0 | Wagner |
| Trompetenstöße | 0 | 0 | 1 | 1 | Trompete |
| Trompete | 0 | 0 | 1 | 0 | - |
| Klavierspiel | 0 | 0 | 1 | 1 | Klavier |
| Klavier | 0 | 0 | 1 | 0 | - |
| Chor | 0 | 1 | 1 | 1 | - |
| Geiger | 0 | 0 | 1 | 1 | - |

Tabelle 6.2: Relationen Tabelle

Gleichzeitig ist 'Geiger' auch einer jener Begriffe, welche durch eine neue Kategorie genau genug beschrieben werden könnten, um nicht mehr als 'Sonstiges' eingeteilt zu sein. In diesem Fall könnte das durch eine neue Kategorie 'Musiker' geschehen.

Bisher werden sehr viele Begriffe als 'Sonstige' eingeteilt, da sie sehr schwer, oder nur durch neue Kategorien differenziert werden könnten. Das ist oft nicht sehr einfach, da immer die Struktur der bisherigen Kategorien (und eventuellen Unterkategorien) bedacht werden muss. Auch kann so eine Einteilung oft nur durch Musikwissenschaftler und mit Rücksicht auf Ludwig Wittgensteins eigene Philosophie geschehen.

Ein Problemfall ist beispielsweise 'Chor'. dieses Wort kann sehr eindeutig der Kategorie 'Gesang' zugeordnet werden, allerdings ist das nicht die einzige Bedeutung dieses Wortes. Was ist ein Chor noch? Kann man ihn problemlos als Musikinstrument bezeichnen und dieser Kategorie zuordnen? Aus Sicht einer Komposition wäre ein Chor sehr wohl ein Musikinstrument, da er wie ein solches eingesetzt werden kann. Gleichzeitig könnte man sagen, ein Musikinstrument zeichnet sich dadurch aus, dass es von einem Menschen gespielt werden muss, um Musik oder zumindest eine akustische Reaktion hervorzurufen. 'Chor' bleibt bisher sowohl in der Kategorie 'Sonstige', als auch in der Kategorie 'Musikinstrument', muss aber sicherlich noch einmal genau betrachtet werden.

Ähnliche Klassifizierungsprobleme bereitet beispielsweise das Wort 'Grammophon'. Zunächst auch ein Mitglied der Kategorie 'Sonstige', stellt sich die Frage, ob es als Musik hervorbringendes 'Gerät' auch in der Klasse 'Musikinstrument' einteilen könnte. Einerseits spricht seine Funktion des Musik hervorbringens durchaus dafür, andererseits definiert sich ein Musikinstrument im Allgemeinen dadurch, dass es von einem Menschen gespielt werden muss, um Töne zu erzeugen. In einem solchen Problemfall lohnt es sich oft, im Text nachzulesen und nach Wittgensteins eigenen Gedanken zu einem Begriff zu suchen. Tut man dies für das Wort 'Grammophon' stößt man auf folgende Textstelle:

Aus Ms-104.92:

"So könnte man in gewissem Sinne sagen daß alle Sätze Verallgemeinerungen aus den Elementarsätzen sind. Es gibt keine Ordnung der Dinge a priori. Die Grammophonplatte, der musikalische Gedanke, die Notenschrift, die Schallwellen, stehen alle in jener abbildenden internen Beziehung zueinander, die zwischen Sprache und Welt besteht.

Ihnen allen ist der logische Bau gemeinsam."

Hier wird die Grammophonplatte mit dem musikalischen Gedanken verglichen. Beide stünden in der selben abbildenden Beziehung zueinander, in der auch 'Sprache' und 'Welt' zueinander stehen. Ein musikalischer Gedanke benötigt, um gehört oder gespielt zu werden, also zum Ausdruck zu kommen, ein Instrument. Wenn man nun den musikalischen Gedanken und die Grammophonplatte nach Wittgenstein vergleichen kann, so könnte man das Grammophon als Musikinstrument einteilen, da die Grammophonplatte ein Grammophon benötigt, um gespielt werden zu können. Diese Einteilung ist mit Sicherheit dennoch nicht unstrittig, allerdings wird dies auch für sehr viele andere Ausdrücke ein langer Pro-

zess sein, sie eindeutig in Klassen einzuteilen. Solange keine zweifelsfreie Einteilung sinnvoll erscheint, ist es immer besser einen Ausdruck in der 'Sonstige' Kategorie zu belassen, da eine falsche Einteilung wesentlich fataler und schwerer bemerkbar wäre, als eine noch nicht vorgenommene Klassifizierung.

Neben neuen Kategorien wie 'Gesang' welche in einer Ontologie sehr sinnvoll wären, sollten auch Unter- und gegebenenfalls Überkategorien geschaffen werden. Überkategorien sind dann nötig, wenn neue Kategorien sehr viel gemeinsam haben und in der Menge aller Klassen einer eigenen Gruppe angehören. Zum Beispiel könnte eine neue Kategorie 'Musiker', in welcher Ausdrücke wie 'Geiger' und 'Klavierspieler' zusammengefasst werden, entstehen. Musiker und Komponisten unterscheiden sich überwiegend durch die gleichen Punkte von den anderen Kategorien. Beides sind Menschen, die etwas tun, und dadurch Musik erschaffen oder möglich machen. Daher könnten sie unter einer Überkategorie 'Musikschaffender' zusammengefasst werden. Wie eine Kategoriestructur mit dieser Überkategorie aussehen könnte, zeigt Abbildung 6.1.

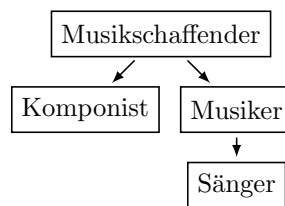


Abbildung 6.1: Überkategorie 'Musikschaffender'

Es ist auch eine Überlegung wert, ob 'Intervalle' eine zu den übrigen Kategorien gleichgestellte Klasse bleiben sollte, oder ob sie nicht besser eine eigene Überkategorie bekommt. Mit bisher nur drei Begriffen hat sie zwar durchaus ihre Daseinsberechtigung, allerdings auch von den Häufigkeiten ihrer Begriffe nicht mit den anderen Kategorien vergleichbar. Hier könnte es daher sinnvoll sein, eine Überkategorie, wie beispielsweise 'Notenterminologie' einzuführen. Eine solche Struktur wird in Abbildung 6.2 dargestellt. Dieser Überbegriff könnte auch Ausdrücke wie 'Tonart', 'Dur' und 'Moll' zusammenfassen. Diese und ähnliche Begriffe befinden sich derzeit noch in der Kategorie 'Sonstige musikalische Begriffe' und könnten durch die neue Kategorie, oder eine andere Unterkategorie neben 'Intervalle' ihre Einteilung finden.

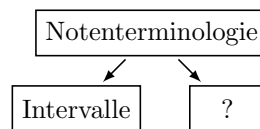


Abbildung 6.2: Überkategorie für 'Intervalle'

6.2 Umwandlung einer Tabelle zur Ontologie

Mit einem Zugang zu diesen musikalischen Beziehungen könnte man die Funktionen von WiTTFind extrem bereichern. Der Anwender hätte direkte Einsicht und Zugriff auf die vorhandenen Relationen und semantischen Schichten der musikalischen Ausdrücke. Im Folgenden wird der Beginn der Arbeit an einem solchen, sehr mächtigen semantischen Tool beschrieben.

Um diese, im vorherigen Abschnitt genauer beschriebenen, Relationen, Eigenschaften und Zugehörigkeiten darzustellen, eignen sich am besten Tabellen. Diese könnten beispielsweise aussehen wie die Tabellen 6.1 oder 6.2. Um solche Tabellen als Ontologien zu

modellieren und zugänglich zu machen, müssen sie in ein dafür geeignetes Format übersetzt werden.

Mit einer Digitalisierung der Wortrelationen wird eine, auf der WiTTFind Webseite zur Verfügung gestellte, Ontologie zu einer Semantic Web Anwendung. Ähnliche Tools für Wittgensteins Werke existieren bereits, ein Beispiel ist der 'Wittgenstein Ontology Explorer'¹ der Wittgenstein Archive an der Universität von Bergen. Hier werden Personennamen der einzelnen Bemerkungen als Ontologie extrahiert, und sogar Teilweise mit klassifizierenden Tags versehen. So erhält man bei einer Suche nach 'Brahms' schon in der Suchleiste die Option, dem Namen den Tag '[:music]' nachzustellen. Sucht man dann nach 'brahms, johannes [:music]' erhält man eine Übersicht über Bemerkungen, in welchen Brahms genannt wird, und sieht direkt, welche weiteren Personen in dieser Bemerkung genannt werden. Einige der angezeigten Personennamen haben ebenfalls das Tag '[:music]', so ist sehr leicht erkennbar, zu welchem Bereich die anderen erwähnten Personen gehören.

Um Daten, wie eine Relationentabelle, in eine Online integrierte Ontologie umzuwandeln, wurden die, bei Horrocks (2008) beschriebenen², Standardformate RDF und OWL entwickelt. Mit ihnen kann man Wissen als konzeptuell geordnetes Schema - genannt Ontologie - darstellen. Das 'Resource Description Framework', kurz RDF genannt, ist eine Sprache in der Beziehungen zwischen einzelnen Einheiten der flexibel beschrieben werden können. Dazu werden IRIs, 'Internationalized Resource Identifiers' verwendet, welche eine Generalisierung von URLs (Uniform Resource Locators) sind, um auf andere Quellen zu verweisen. In RDF sind Elemente als Tripel strukturiert. Das bedeutet, es gibt immer drei Komponenten, ein Subjekt, ein Prädikat und ein Objekt. Hierbei sind Subjekt und Objekt über das Prädikat miteinander verbunden.

Ein simples Beispiel für ein Tripel, welches man auch in RDF formulieren könnte wäre:

Wittgenstein - SpielteInstrument - Klarinette

Hierbei ist 'Wittgenstein' das Subjekt, 'SpielteInstrument' das Prädikat und 'Klarinette' das Objekt. Außerdem existieren in RDF Klassen, so dass man Subjekten auch einen 'Type' geben kann. beispielsweise könnte man 'Beethoven' der Klasse 'Komponist' zuordnen:

Beethoven rdf:type Komponist

Allerdings hat RDF auch Grenzen, welche man, je nach Verwendungszweck, mehr oder weniger schnell erreicht. Da vieles mit RDF nicht ausgedrückt werden kann, wie beispielsweise Kardinalität von Dingen, etwas hat eine 'mindestens ein'-Relation, oder ein Element gehört mehr als einer Klasse an, wurde OWL entwickelt. OWL, 'Web Ontology Language', wurde von einer Initiative des World Wide Web Konsortiums (W3C) entwickelt. Die OWL Syntax basiert auf RDF und erweitert dessen Features.

Die Standard Software um mit OWL zu arbeiten und zu entwickeln ist Protégé³ eine Software der Stanford Universität. Durch sehr viele verfügbare und leicht zugängliche Plugins ist Protégé sehr anpassbar an den jeweiligen Verwendungszweck und das Aussehen der einzelnen Ontologien. Um CSV Dateien, wie die Musikrelationen Tabelle, in OWL umzuwandeln gibt es verschiedene Herangehensweisen. Es gibt ein Plugin namens 'Excel Import'⁴, welches Excel oder CSV Dateien direkt in eine OWL umwandeln soll. Dieses Plugin liegt nicht für die aktuelle Protégé Version vor, sondern nur für Protégé 4.

Während der Beschäftigung mit dem Excel Import Tool in einer Version von Protégé 4 wurde schnell klar, dass die bisherigen Musikrelationen aus den Wittgenstein Nachlass

¹Siehe <http://wab.uib.no/sfb/>

²Hier und im Folgenden aus Horrocks (2008)

³Siehe <http://protege.stanford.edu/>

⁴Vgl. https://protegewiki.stanford.edu/wiki/Excel_Import

noch nicht auf einem Stand sind, vollständig in OWL umgesetzt zu werden.

Ein weiteres getestetes Verfahren ist eine Umwandlung der CSV Dateien mithilfe des 'csv2rdf' Tools ⁵. Dieses Tool wurde entwickelt, um möglichst einfach eine RDF-Ausgabe aus einer CSV Datei zu generieren. Dies geschieht über ein zuvor festgelegtes Template, mit welchem die gewünschten Zuordnungen aus der CSV Datei auf Bestimmte Klassen festgelegt werden. Im nächsten Abschnitt wird an einem Beispiel gezeigt, wie mit diesem Tool und einer Musikontologie Begriffe aus dem Korpus zu RDF umgewandelt werden.

6.2.1 Musikontologie 'MO'

Um eine Tabelle überhaupt in eine Ontologie in RDF oder OWL umwandeln zu können, muss man ein Ontologie Modell als Rahmen verwenden, um Relationen ausdrücken zu können. Bei längerer Beschäftigung mit dem Thema wurde klar, dass es ein sehr ambitionierter Plan ist, eine eigene Ontologie für Musik und Wittgenstein zu kreieren. Für die Zukunft ist es mit Sicherheit ein möglicher Schritt, um gut auf die Besonderheiten dieses Korpus eingehen zu können, wenn eine solches spezielles Ontologie Modell entwickelt wird.

Bei der Suche nach schon bestehenden, geeigneten Ontologien stach die Musikontologie 'the Music Ontology' besonders hervor. Mit dieser Ontologie können verschiedenste Musikalische Relationen und Instanzen Modelliert werden. Beispiele für vordefinierte Klassen sind 'MusicArtist', 'Instrument' und 'Performance'.

Mit cvs2rdf, dieser Ontologie und einem sehr einfachen Tabellenbeispiel kann exemplarisch gezeigt werden, wie musikalische Relationen in RDF umgewandelt werden können.

Hierzu wurde diese sehr simple, einspaltige Komponistentabelle verwendet:

| Komponist |
|-----------|
| Wagner |
| Bach |
| Haydn |
| Brahms |

Danach wurde, dem Beispiel zur Verwendung von csv2rdf nachempfunden, ein Template erstellt, um die Werte in der CSV Datei einzulesen:

```

1 @prefix mo: <http://purl.org/ontology/mo/#> .
2 @prefix dc: <http://purl.org/dc/elements/1.1/> .
3 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
4 @prefix tl: <http://purl.org/NET/c4dm/timeline.owl#> .
5 @prefix event: <http://purl.org/NET/c4dm/event.owl#> .
6 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
7 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
8
9 :Komponist-`${Komponist}` a mo:MusicArtist ;
10   rdfs:label "`${Komponist}`" .

```

Die Zeilen eins bis sieben definieren die Namespaces der einzelnen Ausdrücke 'mo:', 'dc:', 'xsd:', 'tl:', 'event:', 'foaf:' und 'rdfs:' mit ihren jeweiligen IRIs. Zeile neun und zehn legen fest, dass jedes Element der Spalte 'Komponist' der CSV Datei eine Instanz von mo:MusicArtist, also der Klasse 'MusicArtist' aus der 'the Music Ontology'-Ontologie sein soll.

Die Ausgabe würde folgendermaßen aussehen:

⁵Siehe <https://github.com/clarkparsia/csv2rdf>

```
1 @prefix [...]
2
3 :Komponist-Wagner a mo:MusicArtist ;
4   rdfs:label "Wagner" .
5
6 :Komponist-Bach a mo:MusicArtist ;
7   rdfs:label "Bach" .
8
9 :Komponist-Haydn a mo:MusicArtist ;
10  rdfs:label "Haydn" .
11
12 :Komponist-Brahms a mo:MusicArtist ;
13  rdfs:label "Brahms" .
```

An diesem sehr einfachen Prototypen lässt sich schon erkennen, dass die Umwandlung einer derartigen CSV Datei in das RDF Format mit einer Musikontologie durchaus möglich ist. Diese Arbeit ist nur der Einstieg in das Thema Musikontologie für den Nachlass von Ludwig Wittgenstein und kann mit Sicherheit in der Zukunft noch weiter ausgebaut und zu einem guten Tool entwickelt werden. Hierfür muss zuerst der Ausbau und die Strukturierung der Relationen Tabelle sehr viel weiter voranschreiten, und eventuell kann zünftig sogar eine eigenes Ontologiemodell für die Ausdrücke kreiert werden.

7 Evaluierung - Benutzerfreundlichkeit der Webseite

Um die Frontend Umsetzung der semantischen Suche der WiTTFind Webseite zu bewerten, wurde ein Fragebogen entwickelt.

Da das weiterentwickelte Frontend sich, wie in Kapitel 3 beschrieben, auf einem lokalen Webserver befindet, ist es nicht möglich eine Onlineumfrage durchzuführen. Die Befragte Person muss mit dem Frontend über das lokale Computersystem interagieren, daher ist es sehr schwer eine Befragung mit vielen Personen zu realisieren.

Aus diesen Gründen fiel die Entscheidung, das Tool nur von einer Person testen und bewerten zu lassen. Allerdings ist die Befragte, als Studentin der Philosophie, welche sich in ihrem Studium ebenfalls mit Wittgenstein beschäftigt, eine mögliche Endanwenderin der Finder-App und somit sehr gut geeignet das Tool auf seine Benutzerfreundlichkeit und Anwendbarkeit zu testen.

In Zukunft könnte durch eine größere Zahl möglicher Endanwender bei einer solchen Befragung ein repräsentatives Ergebnis erzielt werden.

Die folgenden Unterpunkte geben einen genaueren Überblick über die Auswahl der gestellten Fragen sowie die Auswertung der Ergebnisse.

7.1 Erstellung des Fragebogens

Die Basis des Fragebogens ist das „System Usability Scale“ welche in Bangor et al. (2009) sowie Brooke (1996) vorgestellt wird. Dieses „SUS“ genannte Bewertungssystem für Benutzererfahrung besteht aus zehn Aussagen, welche in eine von fünf Antwortmöglichkeiten zwischen 'Starker Widerspruch' bis zu 'Starke Zustimmung' eingeordnet werden. Von diesen zehn Aussagen, sind je fünf positiv und negativ. Im Original kommt je alternierend eine positive und eine negative Aussage hintereinander.

Ursprünglich von Brooke (1996) entwickelt, stellt es bis heute einen Standard für Nutzerbefragungen verschiedenster Art dar. Eigentlich unter der Prämisse 'quick and dirty' entwickelt, hat es sich als erstaunlich nützliches und langlebiges Tool entwickelt. Als Evaluationstool ist es ein stabiles und verlässliches Maß, welches vielleicht noch weitere 25 Jahre verwendet werden wird (Sauro (2011)).

Wie von Brooke (1996) beschrieben, kann Nutzerfreundlichkeit nicht als absolut gesehen werden, da dieser Begriff in verschiedenen Zusammenhängen unterschiedliche Bedeutungsdimensionen hat. Treffender beschreibt man 'Nutzerfreundlichkeit' als Angemessenheit zu einem vorgesehenen Zweck der Anwendung. Um die Kriterien dafür festzulegen, müssen einige Punkte in Betracht gezogen werden, unter anderem:

- Welche Endanwender werden das System nutzen?
- Welche Art Aufgaben sollen mit dem System ausgeführt werden?
- Wie wird die Umgebung sein, in welcher die Nutzer das Tool anwenden?

Beispielsweise sollte eine Social Media Anwendung anders gestaltet sein, und auf andere Benutzerbedürfnisse eingehen als ein Medizinisches Tool, welches nur von Ärzten

verwendet wird.

Ein gutes, benutzerfreundliches System sollte effizient sein, sowohl im Bezug auf die Qualität des erzeugten Ergebnisses als, auch auf die verbrauchten Ressourcen um dieses Ergebnis zu erzielen und den Nutzer zufriedenstellen.

Die zehn Fragen des SUS-Testes sind der Grundstock für den Fragebogen, als Ergänzung enthält er zwei weitere Fragen, welche noch ausführlichere Antworten ermöglichen als die Skalenbewertung. Für den Fragebogen wurden die zehn SUS-Fragen lediglich auf deutsch übersetzt.

Die beiden einleitenden Sätze am Beginn stammen aus der SUS-Version von Bangor et al. (2009). Die kurze Anleitung ist nützlich, da sie Befragten hilft, den Test auszufüllen.

Die fertige Version des Fragebogens sieht folgendermaßen aus:

Fragebogen zum Frontend für Semantische Suche bei WiTTFind:

Bitte kreuze bei den folgenden Aussagen an, wie sehr du ihnen zustimmst. Hierbei steht „1“ für starken Widerspruch und „5“ für starke Zustimmung.

Halte dich nicht zu lange mit den einzelnen Aussagen auf, im Zweifel kreuze „3“ an.

1. Ich denke ich möchte das Tool in Zukunft häufiger nutzen:

1 2 3 4 5

2. Das System ist unnötig komplex:

1 2 3 4 5

3. Das System war einfach zu nutzen:

1 2 3 4 5

4. Ich denke ich brauchte Hilfe von einer fachlichen Person, um das System nutzen zu können:

1 2 3 4 5

5. Ich finde die Features des Produktes sind gut integriert:

1 2 3 4 5

6. Ich denke, es gibt zu viele Unstimmigkeiten in diesem System:

1 2 3 4 5

7. Ich kann mir vorstellen, dass viele Menschen sehr schnell gut mit diesem Tool umgehen können:

1 2 3 4 5

8. Ich kam mir seltsam vor, das Tool zu benutzen:

1 2 3 4 5

9. Ich war selbstsicher im Umgang mit dem Tool:

1 2 3 4 5

10. Ich müsste noch vieles lernen bevor ich gut mit diesem Tool umgehen kann:

1 2 3 4 5

11. Der Aufbau der Webseite ist:

unverständlich | verwirrend | nicht ganz übersichtlich | gut zu verstehen | leicht zu verstehen

12. Was ist das erste, was auf der Webseite auf den ersten Blick auffällt?

7.2 Auswertung der Ergebnisse

Bei der Evaluierung beschäftigte sich die Befragte einige Zeit selbst mit dem System und testete verschiedene Eingaben und Funktionen. Anschließend füllte sie den Fragebogen aus und stellte Fragen zum Tool. Dann folgte eine Besprechung der auf dem Fragebogen gegebenen Antworten.

Das im vorigen Punkt beschriebene SUS System wird folgendermaßen bewertet: ¹

Die Fragen der Nummern eins, drei, fünf, sieben und neun werden je mit der Skalenbewertung (eins bis fünf) minus eins bewertet. Bei den Fragen zwei, vier, sechs, acht und zehn ist der Wert fünf minus die Skalenbewertung. So ergibt sich für jeden Punkt des SUS Fragebogens ein Wert zwischen eins und vier. Diese Werte werden aufsummiert und mit 2,5 multipliziert, um den Endwert zu bilden.

Dieser Endwert gilt als Maß, für die Benutzerfreundlichkeit eines Systems, je höher er ist, desto besser ist das System. Aus den Skalenbewertungen und SUS-Scores ergibt sich Tabelle 7.1.

| Fragenummer | Skalenbewertung | SUS-Score |
|-------------|-----------------|-----------|
| 1. | 5 | 4 |
| 2. | 1 | 4 |
| 3. | 5 | 4 |
| 4. | 2 | 2 |
| 5. | 4 | 3 |
| 6. | 1 | 4 |
| 7. | 5 | 4 |
| 8. | 1 | 4 |
| 9. | 4 | 3 |
| 10. | 2 | 2 |

Tabelle 7.1: Tabelle mit Skalenbewertungen und den daraus entstehenden SUS-Scores

Aus diesen Bewertungen ergibt sich ein Endwert von 85. Der Wert ist damit überdurchschnittlich hoch, wobei der Durchschnitt bei einem Wert von 68 ² liegt.

Zusätzlich zu dieser Bewertung hatte die Befragte noch einige Anmerkungen zur Webseite, welche im Folgenden besprochen werden. Es sollte besser sichtbar angezeigt werden, auf welchen Link zuletzt geklickt wurde. Besonders in der neuen Musikkategorie mit den verschiedenen Kontexten ist es sonst zu leicht zu verwechseln, und man kann nicht sofort erkennen, zu welchem Kontext die angezeigte Wordcloud gehört. Desweiteren ist es unpraktisch, dass die Auswahl der runden, in Abbildung 3.1 zu sehenden Checkboxen nichtmehr rückgängig gemacht werden kann. Ist eine solche runde Checkbox einmal ausgewählt, muss man die Seite neu laden um diese Auswahl aufzuheben. Die Funktion der Wordcloud, dass man durch Klicken auf ein angezeigtes Wort dieses der Suchleiste hinzufügt, ist nicht intuitiv ersichtlich. Man könnte beispielsweise, bei dem Klicken auf den Link, welcher die Wordcloud anzeigt, ein Informationsfeld anzeigen, welches auf diese Funktion hinweist.

Es ist auch zu überlegen, ob der Begriff 'Stopwort', welcher auf der Webseite im Zusammenhang von Kontext verwendet wird, für die Anwender verständlich ist. Hierbei sollte beachtet werden, von welchem Publikum die Anwendung genutzt wird. Da dieser Begriff eher linguistischer Natur ist, WiTTFind aber als interdisziplinäre Anwendung auch für die Philosophie interessant ist, könnte er zum leichteren Verständnis zukünftig eventuell von einem anderen Wort ersetzt werden.

¹Übernommen aus: Brooke (1996)

²Sauro (2011)

Abschließend lässt sich sagen, dass die Evaluierung der Webseite sehr gut ausgefallen ist. Zukünftig sollten noch mehr Anwender zu WiTTFind befragt werden, um so eine noch bessere Benutzerfreundlichkeit zu gewährleisten.

8 Ausblick

Diese Arbeit ist die Weiterführung und der Beginn neuer Herangehensweisen an den Nachlass Ludwig Wittgensteins. In Zukunft gibt es noch sehr viele Möglichkeiten, die vorgestellten Themen weiter auszubauen und zu verfeinern. In diesem Kapitel soll ein Ausblick über die möglichen anschließenden Schritte gegeben werden, vor allem, was den Ausbau der WiTTFind Finderapp betrifft.

8.1 Konkordanzdarstellung für Kontext

Ein interessantes Feature für WiTTFind könnte es sein, zusätzlich zu den Kontextübersichten der einzelnen Kategorien, den Kontext einzelner Musikbegriffe als Konkordanz darzustellen. Somit könnten die Kontextvorkommen direkt verglichen werden, um dem Anwender ein noch besseres Gefühl für die semantisch korrelierten Kontextwörter zu vermitteln.

Solch eine Konkordanzdarstellung könnte ähnlich aussehen wie Tabelle 8.1:

| | | |
|---|--|--|
| fehlen. Die Melodien der frühen letzten Satz eines der letzten sehenswert aussieht. “Das große Herz | Beethovenschen Beethovenschen Beethovens | Werke haben ein anderes Rassegesicht Quartette den er mehr als ” – niemand könnte sagen “das große |
|---|--|--|

Tabelle 8.1: Beispiel für Konkordanzdarstellung

Je nach Bedarf und Kontextvariante mit mehr oder weniger abgebildetem Kontext, wäre das eine nützliche Funktion für die semantische Darstellungsweise.

8.2 Ontologie Darstellung

Wenn die Ontologie der Musikbegriffe einen Entwicklungsstand erreicht hat, in dem sie gut anwendbar ist, sollte sie auch für den Nutzer auf der Webseite zur Verfügung stehen. Mit einem Zugriff auf die in Kapitel 6 beschriebenen Beziehungen zwischen den Wörtern können sehr viele weitere interessante Zusammenhänge und Besonderheiten von Ludwig Wittgensteins Nachlass gefunden werden.

Im Webfrontend könnte diese Umsetzung durch eine interaktive Tabelle geschehen, in welcher der Nutzer einzelne Klassen auswählt und die entsprechenden Begriffe angezeigt bekommt. Die entsprechenden Begriffe sind dabei immer solche, die in ausgewählten Kategorien enthalten sind, oder einen Bezug zu diesen haben. Wichtig ist, dass diese Funktion übersichtlich bleibt und nicht zu viel auf einmal darstellt. Sost kann der Nutzer sie nicht Überblicken und dadurch nicht effizient und zufriedenstellend benutzen. Bei der Darstellung der Zusammenhänge sollte vielleicht mit abgebildet werden, welche zusätzlichen Verbindungen ein Ausdruck zu anderen Klassen oder Begriffen hat.

Diese Umsetzung ist aber erst dann möglich, wenn durch eine umfassende Klassifizierung der musikalischen Begriffe Tabellen erstellt werden können, die durch Tools in RDF oder OWL Ontologien umgewandelt werden können.

9 Zusammenfassung/Fazit

Im Laufe dieser Arbeit erfolgte die Erweiterung der WiTTFind-Web-Applikation um ein Modul für die semantische Suche von Musikbegriffen. Zuerst waren Anpassungen der Webanwendung nötig, um die neu erstellten Daten für den Anwender zugänglich zu machen. Als Grundlage für die verwendeten musikalischen Ausdrücke diente eine schon existierende, speziell für den Wittgenstein-Korpus angefertigte Liste. Diese Liste ist dato unvollständig und deckt nicht alle verwendeten musikalischen Ausdrücke ab, so dass sie in Zukunft noch erweitert werden kann. Diese Daten wurden anschließend mit diversen Skripten aus dem Wittgenstein-Nachlass extrahiert. Zunächst fiel hier der Fokus ausschließlich auf die Häufigkeiten der vorkommenden Musikbegriffe. Es war jedoch schnell klar, dass die einzelnen Textstellen so interessant sind, dass es lohnenswert ist, den Kontext der Wörter ebenfalls zu betrachten.

Um die Benutzerfreundlichkeit der bisherigen und neuen Features beurteilen zu können, evaluierte sie ein Endanwender mithilfe eines auf dem SuS-Score aufgebauten Fragebogens. Dieser Test ergab eine im Großen und Ganzen sehr positive Bewertung. Die Seite ist sehr effizient und zweckerfüllend nutzbar. Dennoch könnte das Nutzungserlebnis durch kleinere Optimierungen noch angenehmer gestaltet werden.

Da die Relationen der einzelnen Musikbegriffe zueinander ebenfalls betrachtenswert sind, lohnt sich eine Modellierung als Ontologie. Für diese bot sich die Nutzung von verschiedenen Tools sowie etablierten Standardsprachen für Semantic-Web-Anwendungen an. Im Rahmen dieser Untersuchungen fand sich eine schon bestehende Ontologie für musikalische Ausdrücke, die dann auch prototypisch zur Modellierung eines rudimentären Beispiels zur Anwendung kam. In zukünftigen Projekten lohnt es sich, auf diesen Punkt noch einmal ein Augenmerk zu werfen.

Diese Arbeit stellt den Anfang des musikalischen Herangehens an den Nachlass Ludwig Wittgensteins dar. Sie legt einen Grundstein für zukünftige Untersuchungen und Erweiterungen in dieser Hinsicht. Besonders die Implementierung einer Ontologiedarstellung ist hier erstrebenswert.

Literaturverzeichnis

- Antonacopoulos, A. and Schulz, K. U., editors (2014). *Digital Access to Textual Cultural Heritage 2014, DATeCH 2014, Madrid, Spain, May 19-20, 2014*. ACM.
- Apache Free Software Foundation (2017). The apache http server project. Website. Online erhältlich unter: <http://httpd.apache.org/>; abgerufen am 11.6.2017.
- Bangor, A., Kortum, P., and Miller, J. (2009). Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4:114–123.
- Bower Team (2017). Bower - a package manager for the web. Website. Online erhältlich unter: <https://bower.io/>; abgerufen am 11.6.2017.
- Brooke, J. (1996). Sus - a 'quick and dirty' usability scale. Website. Online erhältlich unter: www.usabilitynet.org/trump/documents/Suschapt.doc; abgerufen am 06.06.2017.
- Clark and Parsia LLC (2017). Simple, open source utility to convert csv/tsv files to rdf. Website. Online erhältlich unter: <https://github.com/clarkparsia/csv2rdf>; abgerufen am 11.6.2017.
- Dan, A. and Towsley, D. (1990). An approximate analysis of the lru and fifo buffer replacement schemes. *SIGMETRICS '90 Proceedings of the 1990 ACM SIGMETRICS conference on Measurement and modeling of computer systems*, 18:143–152.
- Dziurla, P. (2016). Bemerkungen über musik im nachlass und den publizierten schriften ludwig wittgensteins. Hausarbeit.
- Eggers, K. (2014). *Ludwig Wittgenstein als Musikphilosoph*. Musik - Philosophie / MusikPhilosophie. Verlag Karl Alber.
- Horrocks, I. (2008). Ontologies and the semantic web. *Communications of the ACM*, 51(12):58–67.
- Krey, A. (2013). Semantische annotation von adjektiven im big typescript von ludwig wittgenstein. CIS Bachelorarbeit.
- Marco G (2017). Extended list of german stopwords. Website. Online erhältlich unter: https://github.com/solariz/german_stopwords/blob/master/german_stopwords_plain.txt; abgerufen am 11.6.2017.
- Paumier, S. (2003). Unitex user manual. Webseite. Online erhältlich unter: <http://www.cis.uni-muenchen.de/people/lg3/ManuelUnitex.pdf>.
- Rhees, R. (1984). *Recollections of Wittgenstein*. Oxford University Press.
- Sauro, J. (2011). Measuring usability with the system usability scale (sus). Website. Online erhältlich unter: <https://measuringu.com/sus/>; abgerufen am 06.06.2017.
- Stanford Center for Biomedical Informatics Research (2017a). A free, open-source ontology editor and framework for building intelligent systems. Website. Online erhältlich unter: <http://protege.stanford.edu/>; abgerufen am 11.6.2017.

- Stanford Center for Biomedical Informatics Research (2017b). Protege wiki - excel import. Website. Online erhältlich unter: https://protegewiki.stanford.edu/wiki/Excel_Import; abgerufen am 11.6.2017.
- Stern, D. (2010). The bergen electronic edition of wittgenstein's nachlass. *European Journal of Philosophy*, 18(3):455–467.
- Wittgenstein Archives at the University of Bergen (2017). Wittgenstein ontology explorer. Website. Online erhältlich unter: <http://wab.uib.no/sfb/>; abgerufen am 11.6.2017.

Abbildungsverzeichnis

| | | |
|-----|--|----|
| 3.1 | Screenshot der WiTTFind Webseite auf dem lokalen Webserver | 11 |
| 4.1 | Beispiel für eine Wordclouddarstellung | 15 |
| 4.2 | Beispiel für eine Kontextwordcloud | 16 |
| 4.3 | Beispiel für das Frequenzdictionary | 17 |
| 5.1 | Veranschaulichung eines Ringbuffers | 23 |
| 5.2 | Listenoperationen | 25 |
| 6.1 | Überkategorie 'Musikschaffender' | 31 |
| 6.2 | Überkategorie für 'Intervalle' | 31 |

Tabellenverzeichnis

| | | |
|-----|--|----|
| 4.1 | Verteilung der musikalischen Begriffe auf ihre Kategorien | 14 |
| 4.2 | Kategorien mit Beispielausdruck | 15 |
| 5.1 | Vergleich der Implementierungen mit cProfile | 26 |
| 5.2 | Interessante Auszüge aus den Kontextfrequenzen | 27 |
| 6.1 | Meistersinger-Wagner Relation | 29 |
| 6.2 | Relationen Tabelle | 30 |
| 7.1 | Tabelle mit Skalenbewertungen und den daraus entstehenden SUS-Scores . . | 39 |
| 8.1 | Beispiel für Konkordanzdarstellung | 41 |

Inhalt der beigelegten CD

- Diese Arbeit
- Quellcode der Pythonskripte
- Alle Quellen aus dem Literaturverzeichnis
- Programmausgaben