PROJEKTBERICHT FÜR *Information Retrieval*

Richard Aude
B.Sc. Digital Humanities
Universität Leipzig
Matrikelnummer: 3733913

August 21, 2019

ABSTRACT

Dies ist die Dokumentation zum Entstehungsprozess einer Suchmaschine, die im Rahmen des Moduls Information Retrieval im Sommersemester 2019 entstanden ist. Die Suche beschäftigt sich mit den Reden der Partei *Alternative für Deutschland (AfD)* im deutschen Bundestag während der 19. Legislaturperiode.

1 Motivation

Aussagen von Politiker*innen der *AfD* werden oft recht gründlich analysiert, da sie aus politikwissenschaftlicher bis hin zu sprachwissenschaftlicher Perspektive interessant sind [1][2]. Diese Suchmaschine kann als Hilfsmittel für die Recherche verstanden werden, außerdem zum Informieren von Positionen der *AfD* zu verschiedenen politischen Themen, da die *AfD* selbst oft in sich widersprüchliche Meinungen vertritt [3].

2 Daten

Die Daten, auf denen die Suche basiert, bestehen aus selbst gecrawlten XML-Protokollen des Bundestags [4]. Es wurden alle bisherigen 109 Sitzungen der 19. Legislaturperiode nach Reden von Mitgliedern der *AfD* durchsucht, diese wurden dann in die Datei `protokolle5.json` extrahiert, welche sich auch im von mir angegebenen Git Repository befindet. Kommentare zu Zwischenrufen, Gelächter usw. wurden entfernt, lediglich zu Protokoll gegebene Zwischenfragen und die Kommentare des Bundestagspräsidenten und seiner Stellvertreter*innen wurden beibehalten. Durch Ausführen der Datei `crawl.py` wurde das JSON Dokument erstellt.

Ein Eintrag in der JSON Datei sieht zum Beispiel folgendermaßen aus:

"dokument_id": 1,
"name": "Dr. Bernd Baumann",
"datum": "24.10.2017",
"sitzungsnummer": "1",
"rede": "Dr. Bernd Baumann (AfD): Herr Präsident! Meine Damen und Herren! Immer deutlicher zeigte sich im Verlauf dieses Jahres, dass die AfD in den Bundestag einziehen würde (...)"

3 Architektur

In diesem Abschnitt wird auf die für die Realisierung des Projekts benutzte Architektur eingegangen. Es gibt dabei zwei große Bausteine: **Python** und **Elasticsearch**. Mit Python wurden Daten gecrawlt und dann mit der in Python vorhandenen Bibliothek für Elasticsearch indiziert. Für die Benutzeroberfläche wurde auf eine einfache Kombination der Python Bibliothek Flask und Html zurück gegriffen.

3.1 Index

Der Index in seiner aktuellen Form wird in `restructuring.py` definiert. Mit Hilfe der `bulk` Methode werden alle Dokumente aus der JSON Datei mit den angegebenen Feldern in den Index eingefügt, dessen Name angegeben wurde. Die unter `es.indices.create` getroffenen Einstellungen erläutere ich unter der Sektion zur zweiten Evaluation: 4.2.

3.2 GUI

Mit Hilfe der Python Bibliothek Flask wurde eine simple Benutzeroberfläche in Html für den Browser erstellt, die unter `localhost:8000` abrufbar ist, wenn das Programm läuft.
Die Suchergebnisse werden als Snippets dargestellt, nachdem eine Query gesucht worden ist. In Blau sind dabei als Überschrift der Name der redenden Person und das Datum gegeben, darunter finden sich ein bis vier Highlights von Matchings der Query in dem gegebenen Dokument. Wenn man sich die vollständige Rede anschauen möchte, klickt man einfach auf das Snippet, und die ganze Rede wird ausgeklappt.
Zum Schluss jeder Rede findet sich ein Link zu dem gesamten Protokoll der Sitzung auf der Seite des Deutschen Bundestags, in der die Rede vorkam, zum Nachlesen des Kontexts. Deshalb wurde auch die Sitzungsnummer zu jedem Dokument in der JSON Datei eingespeichert. Die Ergebnisse sind paginiert zu jeweils 10 Stück pro Seite.

4 Evaluation

Zum Evaluieren der Suchmaschine wurden 50 Topics erstellt. Die einzelnen Topics können in den Ordnern `relevances` bzw. `relevances2` nachgelesen werden, als Query wurde immer der jeweilige Titel ("title") des Topics verwendet. Für jedes Topic wurden die ersten 5 Ergebnisse binär als relvant oder irrelevant gekennzeichnet. Darauf basierend wurde die `precision@5` berechnet, um schließlich die Mean Average Precision zum Stand der ersten Evaluation zu erlangen.

4.1 Evaluation 1

Bis zum Stand der ersten Evaluation wurden alle Indizierungseinstellungen auf Standardniveau gelassen. Beim Suchen wurde nur das "rede" Feld eines jeden Dokuments mit dem Elasticsearch Query Syntax Befehl `match_phrase` durchsucht. Da in der Rede auch nochmal der Name jeder sprechenden Person steht, wurde es erstmal nicht für nötig befunden, das "name" Feld auch einzubeziehen.

4.2 Evaluation 2

Zum Stand der zweiten Evaluation wurden verschiedene Maßnahmen zur Optimierung der Performance getroffen.

mehrere Felder Das bisherige Feld "rede" wurde nochmal in zwei kleinere Felder unterteilt: Der ersten Absatz jeder Rede bekam ein eigenes Suchfeld, und, wenn vorhanden, wurden alle restlichen Absätze zusammen in ein weiteres Suchfeld gepackt. Das ursprüngliche "rede" Feld mit dem Volltext der Rede wurde im Index gelassen, allerdings wurde es ein klein wenig manipuliert, wie im nächsten Absatz beschrieben wird. Außerdem wurde das "name" Feld mit in die Query integriert.
Jedes Feld wurde mit einer Gewichtung versehen: Das "name" Feld am höchsten, da, wenn jemand nach dem Namen eines AfD-Bundestagsmitglieds suchen sollte, er höchstwahrscheinlich zuerst alle Reden dieser Person sehen möchte. Der erste Absatz bekam nur eine normale Gewichtung, da recht häufig im ersten Absatz für das eigentliche Thema irrelevante Stichwörter auftauchten. Alle anderen Absätze bekamen eine etwas höhere Gewichtung, und die Gewichtung der Gesamtrede (des "rede" Felds) lag zwischen dem des ersten Absatzes und aller anderen Absätze.

Beseitigung von Bindestrichen Bei der ersten Evaluation fiel auf, dass Wörter, die durch Bindestriche zusammengesetzt waren, sehr oft Probleme bereiteten. Zum Beispiel war "linksgrün" ein Topic, welches recht schwierig war, da "linksgrün" sehr häufig als "links-grün" geschrieben wurde - was bisher durch die Suche fiel, da der Bindestrich anscheinend dafür sorgte, dass die Wörter "links" und "grün" als einzelne Token betrachtet wurden. Also wurden im

gesamten "rede" Feld eines jeden Dokuments alle Bindestriche gelöscht, in den anderen Feldern wurde die ursprüngliche Formatierung beibehalten.

Stemming Es wurde Stemming für Deutsch als Filter in den Index eingebaut, außerdem wurden neben den für das Deutsche üblichen Stoppwörtern noch einige englische Stoppwörter hinzugefügt, für den Fall dass englischsprachige Namen internationaler Bewegungen, Abkommen, Organisationen usw. in einer Rede genannt werden.

Synonyme Es wurden manuell einige Synonyme eingefügt. Diese sind keineswegs als politische Statements zu verstehen, das primäre Ziel war dort einfach nur eine gewisse Verwandtheit zwischen Begriffen in bestimmten Themenkomplexen wie z.B. Sexualität herzustellen, in denen es sehr viele Unterbegriffe gibt.

Anpassen der Okapi BM25 Formel In der Okapi BM 25 Formel, die der Suche in Elasticsearch zu Grunde liegt, wurden die Parameter **b** und **k1** von mir manuell angepasst. Der Parameter b hat per default einen Wert von 0.75 in Elasticsearch und ist für den Einfluss von Dokumentlänge auf Score zuständig. Ich habe b in meinem Index auf 0 gesetzt, da ich nicht wollte, dass die Länge eines Dokuments irgendeinen Einfluss auf den Score hat.
Der Parameter k1 hat per default einen Wert von 1.2 in Elasticsearch. Ich habe ihn auf 10 gesetzt, da nach meinem Verständnis ein höherer Wert von k1 dafür sorgt, dass mit relativ häufigem Vorkommen des Suchterms in einem Dokument der Score für das Dokument und den jeweiligen Suchterm relativ steigt [5].
Zusammengefasst war mein Ziel, längere Dokumente mit einem häufigeren Vorkommen des Suchterms hoch zu ranken, da eine Beobachtung während der ersten Evaluation war, dass vor Allem eher kurze Dokumente, in denen der Suchterm nur einmal vorkam, am höchsten gerankt wurden.

4.3 Evaluationsvergleich

Im folgenden werden die Mean Average Precisions der beiden Evaluationen verglichen und interpretiert.

Evaluation	Datum	Mean Average Precision
1	August 11, 2019	0.9187222222222223
2	August 20, 2019	0.931861111111111

Zu Evaluation 1 gab es bei manchen Topics gar keine Ergebnisse. Dies hatte sich bei Evaluation 2 verbessert - es gab nur noch ein Topic ohne relevante Ergebnisse ("Rechtsextreme"). Es war extrem schwierig, das zu beurteilen, aber die AfD spricht im Bundestag nicht in einem "besonderen Aspekt" über Rechtsextremismus, sondern zählt ihn nur auf, wenn es um alle möglichen Arten von Extremismus geht. Es geht also in keinem der Top 5 Ergebnisse nur um die Gefahren des Rechtsextremismus allein, wie in der "narrative" des Topics gefordert.

5 Fazit

5.1 Workflow

Durch eine Reihe unglücklicher Umstände ist es dazu gekommen, dass von ursprünglich vier Personen nur ich in diesem Projekt übrig geblieben bin. Ich habe mich bestens bemüht, eine gute Suche zu gestalten und denke, dass die Aufgabe vielleicht für mich allein zu anspruchsvoll war. Auf Grund von Kommunikationsschwierigkeiten zwischen dem letzten anderen verbliebenen Gruppenmitglied und mir ist bis zum Ende der Vorlesungszeit nicht an diesem Projekt gearbeitet worden, da wir zuerst ein anderes Thema (UFO Sichtungen) hatten. Auf Grund der kurzen Texte und relativ armen Auswahl an Topics diesbezüglich haben wir uns für eine Domäne mit längeren Texten und einem breiteren thematischen Spektrum entschieden. Letztendlich fielen mir alle Teilaufgaben des Projekts zu, da das letzte andere Gruppenmitglied aus gesundheitlichen Gründen kurzfristig abgesprungen war.
Natürlich wäre der Workflow besser gewesen, wenn auch schon während der Vorlesungszeit an diesem Projekt gearbeitet worden wäre.

5.2 Ergebnisse

Dennoch finde ich, die Ergebnisse können sich sehen lassen. Die GUI ist simpel, aber voll funktionsfähig, und die Suche an sich hat eine relativ hohe Mean Average Precision. Für mich als einen Digital Humanities Studenten im vierten Semester sind die Resultate selbst durchaus zufriedenstellend, da ich vorher auch noch nichts dergleichen in die Richtung Suche und Benutzeroberflächen entworfen habe.

6 Acknowledgements

Diese Arbeit wurde im Rahmen des Kurses *Information Retrieval* (Sommersemester 2019) unter der Leitung von Jun.-Prof. Dr. Martin Potthast und Shahbaz Syed an der Universität Leipzig realisiert.

References

[1] Jens Bisky: Sprache der AfD,
`https://www.sueddeutsche.de/kultur/sprache-der-afd-sie-verwechseln-sich-mit-deutschland-1.4501687`

[2] Heinrich Detering:
`https://www.ndr.de/nachrichten/niedersachsen/braunschweig_harz_goettingen/Germanist-AfD-Sprache-Jargon-von-Verbrechern,afd2100.html`

[3] Nadine Lindner:
`https://www.deutschlandfunk.de/afd-im-bundestag-produktiv-provokativ-widerspruechlich.1773.de.html?dram:article_id=416542`

[4] Übersicht aller Protokolle:
`https://www.bundestag.de/services/opendata`

[5] The bm25 algorithm and its variables:
`https://www.elastic.co/de/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables`