FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Political Polarity in US Twitter

Seminar BigData

at the
Friedrisch Schiller University of Jena
Faculty of Mathematics and Computer Science
Graduate Degree Computer Science

submitted to
Prof. Dr. Bücker,
Dr. rer. nat. Bosse and
Herrn Schoder

submitted by
Kenny Gozali,
Chris Gerlach and
Walter Ehrenberger

Jena, January 27, 2023

Abstract

In der vorliegenden Arbeit behandeln wir eine Sentimentalitätsanalyse von US amerikanischen Politikern aus dem *House of Representatives*. Dazu haben wir Daten von Twitter der letzten 12 Jahren zu den genannten Repräsentanten *gescrapt* und mithilfe des Big Data Frameworks Spark verarbeitet. Ziel der Sentimentalitätsanalyse war es, Unterschiede der beiden Parteien (Republikaner und Demokraten) zu bestimmten politischen und auch allgemeinen Themen herauszufiltern. Jedoch haben sich in den gegebenen Daten weniger Diskrepanzen zwischen den beiden Parteien erkennen lassen, als zu Beginn erwartet, wie im Laufe dieser Arbeit deutlich wird.

Contents

List of Figures

1 Introduction

Als mächtigste Weltmacht beeinflussen die Vereinigten Staaten nahezu jeden Teil des Globus. Mitunter deshalb und aufgrund der enormen Präsenz in den Medien sowie des Einflusses auf diese fallen Diskrepanzen in der Bevölkerung schneller auf als in anderen Ländern. Aufgrund dieser Stellung wirkt sich die dortige Sentimentalität somit auch auf das Leben in anderen Ländern aus. Der Kapitolsanschlag sowie die Black Lives Matter Proteste der letzten Jahre sind ein Zeichen für die zunehmende Polarität und Unzufriedenheit in der Bevölkerung, wie sich auch in folgender Grafik erkennen lässt [**?**].

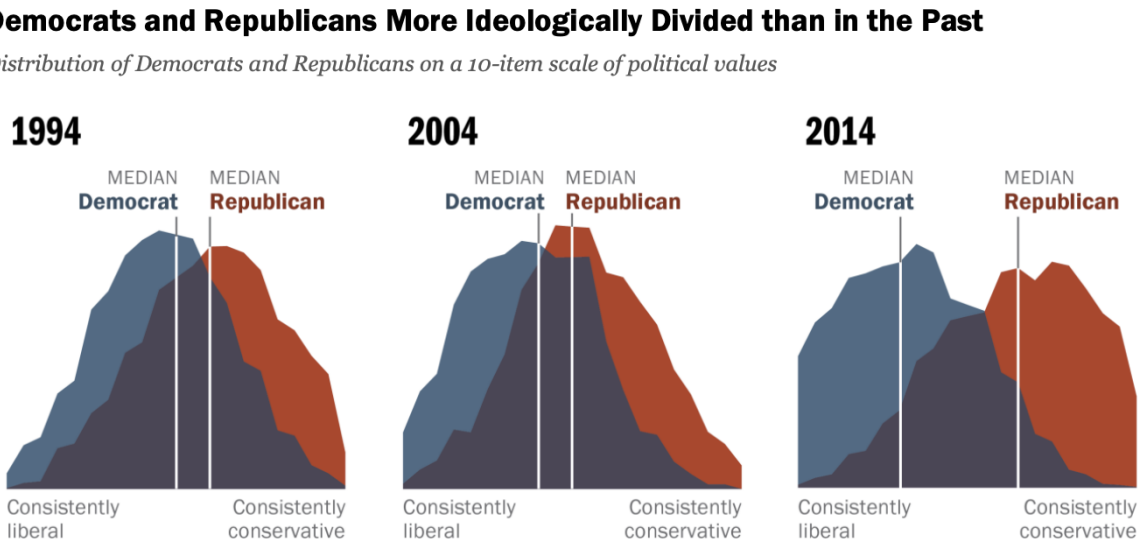

Figure 1.1: Entwicklung der Polarität politisch engagierter Amerikaner. Basierend auf 10 politischen Metriken werden Demokraten (blau) und Republikaner (rot) hier verglichen. Wie zu erkennen bewegen sich die zu Beginn teils noch überlappenden Ideologien in den letzten 10 Jahren auseinander [**?**].

Aufgabe dieser Arbeit ist es nicht, sich mit den komplexen und vielschichtigen Hintergründen für diese Entwicklung auseinanderzusetzen. Vielmehr wird hiermit versucht, eben diese Polarität in den oberen Reihen der amerikanischen Politik genauer

zu analysieren.

Unsere Zielsetzung bestand darin, mit den Tweets der letzten 12 Jahre von 420 Politikern des Repräsentantenhauses eine Sentimentalitätsanalyse durchzuführen. Dabei handelt es sich um ein Mittel der natürlichen Sprachverarbeitung, bei dem die Ansicht beziehungsweise Gefühlslage eines Textes quantifiziert wird.

2 Getting data (good title missing)

2.1 Background

In diesem Kapitel sollen die verwenden Bibliotheken und der Grund für ihre Verwendung genauer beleuchtet werden. Damit wir Daten generieren konnten bzw. können, haben wir die Bibliothek GetOldTweets3 verwendet. Mit einer öffentlich zugängliche Userliste für Politiker aus Amerika wurden dann mit diesem Packages die Daten erhoben. Zur Weiterverarbeitung der Daten haben wir NLTK und TextBlob genutzt. Beides sind Tools für die Verarbeitung von Sprache. Um eine Analyse über die Verarbeiteten Ausgaben durch ein Map-Reduce laufen zu lassen, haben wir als letztes Package Sparks verwendet, um eine zeit effiziente Verarbeitung zu gewährleisten.

2.1.1 GetOldTweets3-Pakage

GetOldTweets3 ist ein kostenlose Python 3 Packages mit welchen Twitterdaten ohne API-Schlüssel abgerufen werden können. Mit GetOldTweets3 können Sie Tweets mit einer Vielzahl von Suchparametern wie Start-/Enddatum, Benutzername(n), Textabfrage und Referenzortbereich durchsuchen. Außerdem können Sie angeben, welche Tweet-Attribute Sie einbeziehen möchten. Einige Attribute sind: Nutzername, Tweettext, Datum, Retweets und Hashtags.[**?**] Die offizielle API von Twitter hat eine lästige Zeiteinschränkung, weshalb man keine Tweets älter als eine Woche abrufen kann. Es gibt einige Tools die Zugang zu älteren Tweets anbieten, diese sind jedoch meistens kostenpflichtig. Das Forscher- team hat nach einem andere Tool gesucht die diese Aufgaben übernehmen, wodurch die Wahl auf das Package GetOldTweets3 gefallen ist.[**?**] Die Analyse des Codes von GetOldTweets3 und die Funktionsweise des Searchthrough Browsers von Twitter zeigt wie das Packages auch an alte Tweets kommt. Grundsätzlich, wenn sie auf Twitter

seiten eingeben oder User suchen, startet ein Scroll-Loader. D.h. wenn sie dann weiter nach unten scrollen, bekommen sie immer mehr Tweets zu den Suchbegriffen. Diese ganzen Tweets bekommen sie durch Abfragen an einen JSON-Provider. GetOldTweets3 imitiert den Searchthrough Browser von Twitter um den Scroll-Loader zu starten und zieht sich dann anhand, der Abfragen an einen JSON-Provider die JSON-Datei und gibt diese decodiert zurück um dann alle Twitts anhand der oben gegebenen Parameter herauszufiltern. Dies kann man in Quelle [?], dem Github-Repositorium gut nachvollziehen. Somit ist es möglich sowohl aktuelle als auch sehr alte Tweets zu scrapen.[?]

```
#!/bin/bash
# cat user_list.csv

while IFS="," read -r rec_column1 rec_column2 rec_column3 rec_column4 rec_column5
do
  echo "Writing to data/$rec_column3"
  python GetOldTweets3/cli.py --username $rec_column3 > data/$rec_column3

done < <(tail -n +2 user_list.csv)
```

Figure 2.1: Code Beispiel für das Scrapen der Tweets
Ist eine Python Bibliothek mit der Twitter Daten durch den Scroll-Loader des Searchthrough Browsers von Twitter als JSON-Datei abgerufen werden können.

So kann durch eine paar Zeilen Code, wie man in der Abbildung sieht, eine bash-Datei erstellt werden, durch welche die Daten gesucht und abgespeichert werden. Das Scraping an sich kann durch die Größe der JSON-Datei einige Zeit in Anspruch nehmen. Wir haben gerade mal 2 Millionen Tweets insgesamt bei einer Laufzeit von ca. 35 Stunden.

2.1.2 NLTK-Natural language Toolkit

NLTK ist ein Python Package für die Arbeit mit menschlichen Sprachdaten. Es bietet einfach zu bedienende Schnittstellen zu über 50 Korpora und lexikalischen Ressourcen wie WordNet, zusammen mit einer Reihe von Textverarbeitungsbibliotheken für Klassifizierung, Tokenisierung, Stemming, Tagging, Parsing und semantische Schlussfolgerungen, Wrapper für industrielle NLP-Bibliotheken und ein aktives Diskussionsforum.[?] Aus diesem Grund bietet NLTK sehr viele Möglichkeiten zur Vorverarbeitung und Analyse, benötigt aber auch einen gewissen Zeitrahmen zur Einarbeitung in die Analysen. Aus diesen Grund hat sich das Forscherteam dafür entschieden NLTK nur zur Vorver-

arbeitung zu nutzen und TextBlob für die Semantische Analyse zu nutzen. Warum sich für TextBlob entschieden wurde, wird genauer in 2.1.3 besprochen. So verwenden wir den Wordcorbus von NLTK für englische Stoppworte, da wir diese nicht mit in unseren Analysen haben wollen. Für eine individuelle und sehr ausführliche semantische Analyse bietet NLTK sehr viele Möglichkeiten, durch die Interaktion mit verschiedenen Packages in Python, was aber den oben genannten Zeitrahmen benötigt, zum einarbeiten. Der Vorteil von NLTK gegenüber TextBlob sind genau diese Interaktionen mit anderen Packages. Für größere Projekte, bei denen man die semantische Analyse auch individuell anpassen möchte,sollte man eher die NLTK Bibliothek benutzen. NLTK ermöglicht es durch verschiedene Vorverarbeitungsschritte, welche in der Bibliothek eingebaut sind eine individuelle Pipeline und Analyse zu erstellen.

```
>>> import nltk
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
>>> tagged = nltk.pos_tag(tokens)
>>> tagged[0:6]
[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'),
('Thursday', 'NNP'), ('morning', 'NN')]
```

Figure 2.2: Code Beispiel für das Arbeiten mit NLTK
Tokenizierung und Tagging von Texten mit NLTK

2.1.3 TextBlob

- Nutzt eine Patternanalyse von der pattern library –¿ Genauer Beschreiben was ist eine Patternanlyse?
- Vorteile von Textblob gegenüber NLTK
- Verwendet auch viele Bibliotheken und ein paar wordcorbus nicht ganz soviel wie NLTK
- Kann ohne viel Zeit aufwand verwendet werden
- Will man individuellere Pipelines bauen braucht man auch einen gewissen Zeitrahmen um sich einzulesen

- Bietet eine bessere Übersicht, da es auch noch nicht so groß ist wie NLTK

- Warum haben wir das Tool genutzt.
- Durch den einfachen output vom Import "from textblob import Textblob", welcher die polarität und subjektivität zurück gibt
- bietet eine sehr gute Anwendungsmöglichkeit für das Mapreduce
- Mit mehr zeit Hätte man auch noch ander Semantische Analysen die von TextBlob bereitgestellt werden verwenden können.

2.1.4 Sparks

- Aus der Vorlesung Vorteile von Spark finden und einbauen

2.2 Scraping and Sanitization

2.2.1 Scraping

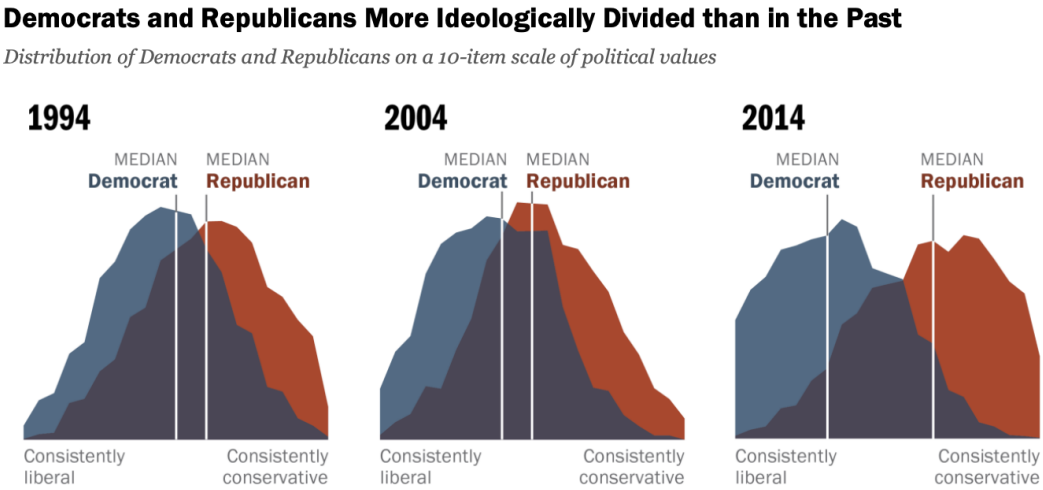

Figure 2.3: Ine Beschreibung f

2.2.2 Sanitization

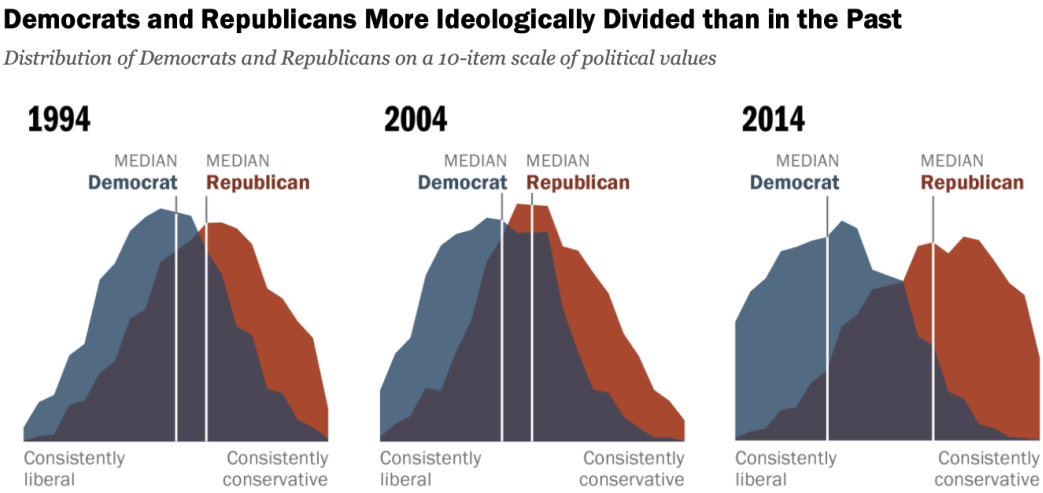

Figure 2.4: Ine Beschreibung f

2.3 MapReduce (good title missing)

Hallo Palmoooooo und Walta

3 Analysing data (good title missing)

3.1 data1 (good title missing)

Hey Kennyyyyy

3.2 data2 (good title missing)

Hey Kennyyyyy

4 Schluss

4.1 Resume

Wie sie sehen sehen sie nichts.

Bibliography

[al.14] AL., Michael D.: Political Polarization in the American Public. (2014)

Internet sources

[Hen18] Henrique, Jefferson: *GetOldTweets-python*. https://github.com/Jefferson-Henrique/GetOldTweetspython/blob/master/got3/manager/TweetManager.py. Version: 2018. – Last visited on 27.01.2023

[Hen19] Henrique, Jefferson: *GetOldTweets3 0.0.11*. https://pypi.org/project/GetOldTweets3/. Version: 2019. – Last visited on 27.01.2023

[Yos20] Yoss, Andrea: *GetOldTweets3*. https://andrea-yoss.medium.com/getoldtweets3-830ebb8b2dab. Version: 2020. – Last visited on 27.01.2023