UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR IT-SICHERHEIT

Provenance of YouTube Comments - Dataset-Creation for BERT-Model Based Authorship Attribution

Die Herkunft von YouTube-Kommentaren - Datensatzerstellung für die Authorship Attribution mit BERT-Models

Bachelorarbeit

im Rahmen des Studiengangs
IT-Sicherheit
der Universität zu Lübeck

vorgelegt von
Tammo Polle

ausgegeben und betreut von
Prof. Dr. Esfandiar Mohammadi

mit Unterstützung von
Yara Schütt

Lübeck, den 4. Juli 2024

Abstract

The concern about manipulation of public opinion through social media-supported disinformation campaigns makes recognizing coordinated user activities and detecting social bots an important research topic. With this work, a fundamental contribution to the analysis of orchestrated user accounts appearing in YouTube comment sections is made by creating a dataset suitable for authorship attribution with BERT models from YouTube comments. To verify the suitability of the developed dataset for authorship attribution of YouTube comments with BERT models, authorship attribution was interpreted as a sequence classification task, where each author is assigned a unique label. By comparing the accuracies determined for this task, the most performant configuration of model and fine-tuning dataset was identified. According to current knowledge, TuBERT is the first BERT model trained for authorship attribution on YouTube comments. Additionally, the attention distribution (attention weights) underlying TuBERT's classification was analyzed, and it was shown that the embeddings of the comments within the TuBERT model are suitable for clustering these comments by their writing style and the topics they address. Thus, this work also lays a foundation for the automated use of BERT-based LLMs for detecting coordinated commentators and social bots on YouTube.

Kurzfassung

Die Sorge vor einer Manipulation der öffentlichen Meinung durch Social-Media-gestützte Desinformationskampagnen, macht das Erkennen koordinierter Nutzeraktivitäten und die Detektion von Social Bots zu einem wichtigen Forschungsthema. Mit dieser Arbeit wird, durch das Erstellen eines für die Authorship Attribution mit BERT-Models geeigneten Datensatzes aus YouTube-Kommentaren, ein grundlegender Beitrag zur Analyse orchestriert auftretender Nutzeraccounts in YouTubes Kommentarspalten gelegt. Um die Eignung des entwickelten Datensatzes für die Authorship Attribution von YouTube-Kommentaren mit BERT-Models zu überprüfen, wurde die Authorship Attribution als Sequence Classification Task interpretiert, in der für jeden Autor ein eigenes Label vergeben wird. Durch den Vergleich der für diese Aufgabe ermittelten Genauigkeiten, konnte die performanteste Konfiguration aus Model und Finetuning-Datensatz ermittelt werden. Nach derzeitigem Kenntnisstand ist TuBERT das erste BERT-Model, das für die Authorship Attribution auf YouTube-Kommentaren trainiert wurde. Zusätzlich wurde die TuBERTs Classification zugrundelegende Aufmerksamkeitsverteilung (Attention Weights) analysiert und gezeigt, dass sich die Einbettungen der Kommentare innerhalb des TuBERT-Models dafür eignen, diese Kommentare nach ihrem Schreibstil und den in ihnen behandelten Themen zu clustern. Damit legt diese Arbeit ebenfalls eine Grundlage für den automatisierten Einsatz BERT-basierter LLMs zur Erkennung koordinierter Kommentatoren und Social Bots auf YouTube.

Erklärung

Ich versichere an Eides statt, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Hilfsmittel angefertigt zu haben.

Lübeck, 04. Juli 2024

Danksagung

Zuallererst danke ich Yara Schütt und Prof. Esfandiar Mohammadi für die umfassende und geduldige Betreuung. Darüber hinaus gilt Prof. Mohammadi und dem Institut für IT-Sicherheit Dank für das Zurverfügungstellen der VM und Hardwareumgebung für die Datensammlung und Verarbeitung sowie für den Zugang zu ChatGPT und LanguageTools. Im Zuge der Erstellung dieser Arbeit wurde ChatGPT eingesetzt, um bei der Programmierung zu unterstützen. So wurden die händisch geschriebenen Python-Programme für die Datensatzerstellung (Kapitel 3.1) und Vorbereitung der Daten für das Model-Finetuning (Kapitel 3.3) durch ChatGPT optimiert. Der Quellcode für die statistische Textanalyse (Kapitel 5.3.2) sowie die Visualisierung von Attention Weights (Kapitel 5.3.3) und Token-Embedding-basiertem Clustering (Kapitel 5.3.5) wurde mit Anpassungen von ChatGPT übernommen. LanguageTools wurde für das Auffinden und Korrigieren von Schreibfehlern verwendet. Abschließend möchte ich meiner Frau Justine für ihre unerschütterliche Unterstützung, insbesondere in den vergangenen Wochen, danken.

Inhaltsverzeichnis

1 Einleitung

Die mit der weitverbreiteten Nutzung Sozialer Medien einhergehende, zunehmende Ersetzung der institutionalisierten Massenmedien, wie dem Öffentlich-rechtlichen Rundfunk, als Bezugsquelle für den persönlichen Nachrichtenkonsum, führte zu großen Veränderungen in der sozialen Öffentlichkeit [18]. Besonders aktive Nutzer können die Rolle von Meinungsführern übernehmen, Agenden setzen und damit für andere Nutzer ein alternatives Bild von der Wirklichkeit erzeugen [21]. Dabei kommt es immer öfter zu accountübergreifend koordinierten Aktionen, deren Ziel die Manipulation der öffentlichen Meinung ist [26]. Häufig werden dafür automatisierte Nutzeraccounts, die softwaregesteuert in den Sozialen Medien aktiv sind, sogenannte *Social Bots*, eingesetzt [25]. Dadurch ist die absichtliche Verbreitung unwahrer Informationen (*Desinformation*) möglich, die unter anderem aufgrund der Beeinflussung von Wahlen, wie bei der US-Präsidentschaftswahl 2016 [2], eine Gefahr für demokratische Gesellschaften sein kann. Diese große gesellschaftliche Relevanz macht das Erkennen koordinierter Aktivitäten und insbesondere das Identifizieren von Social Bots zu einem wichtigen Forschungsthema.
Mit den jüngsten Fortschritten im Bereich der Künstlichen Intelligenz (KI) und *Large Language Models* (LLMs) wächst auch die Sorge vor immer effektiveren Desinformationskampagnen. Andererseits lassen sich LLMs auch dafür einsetzen, große Textmengen zu analysieren, um etwa die Autoren von Social-Media-Beiträgen zu bestimmen (*Authorship Attribution*). Dabei ist das Verständnis koordinierter Aktivitäten je nach Plattform sehr unterschiedlich ausgeprägt. Während insbesondere die Microblogging-Plattform *X* (ehemals und hier weiterhin *Twitter*) in dieser Hinsicht breit untersucht wurde, ist die Videoplattform *YouTube*, trotz ihrer großen Nutzerschaft, in der aktuellen Forschungsliteratur noch deutlich unterrepräsentiert [31].
In dieser Arbeit wird das Verhalten besonders aktiver Kommentatoren auf YouTube untersucht, um eine Grundlage für die automatisierte Erkennung koordinierter Aktivitäten auf dieser Plattform zu legen. Dafür wurde ein Datensatz (*NP-Set*) aus YouTube-Kommentaren zu Videos auf Kanälen aus dem *News & Politics*-Bereich erstellt und mit diesem das LLM *BERT* gefinetuned (*TuBERT*). Hierin liegt auch der Schwerpunkt dieser Arbeit. Weiterhin wurde die Herausforderung der Authorship Attribution auf YouTube-Kommentaren als *Classification Task* für ebendieses Model angewandt, um die Autoren auf Grundlage ihres Schreibstils in Gruppen einteilen zu können. Die performanteste Kombination aus Datensatz und Finetuning-Konfiguration für die Classification Task

wurde experimentell ermittelt. TuBERT ist nach derzeitigem Kenntnisstand das erste BERT-Model, das für die Authorship Attribution auf YouTube-Kommentaren trainiert wurde.
Um die Performance von TuBERT für NP-Set bewerten zu können, wurde ein weiterer Datensatz erstellt: *Gen-Set*. Dieser besteht aus YouTube-Kommentaren zu Videos auf Kanälen, die nicht dem News & Politics - Bereich zugeordnet sind. Aus dem Vergleich der Datensätze, wie die Analyse der Kommentare einzelner Autoren, konnten einige Erkenntnisse über das Kommentierverhalten besonders aktiver Accounts auf YouTube gewonnen werden. Beispielsweise drücke die im News & Politics - Bereich verwendeten Emoticons deutlich weniger häufig positive Emotionen aus, als es im Gen-Set der Fall ist. Dabei wurde neben der Häufigkeitsanalyse einzelner Wörter, Satzzeichen und Emoticons auch die Aufmerksamkeitsverteilung von TuBERT auf einzelnen Teilworten (*Tokens*) am Kommentarbeispiel betrachtet (*Attention Weights*). Mit TuBERT lassen sich in ihrer Wortwahl von der Masse an Kommentaren innerhalb der Finetuning-Sets abweichende Autoren einfach ermitteln. Darüber hinaus konnten wir die Einbettungen der jeweils betrachteten Kommentarteile (*Token-Embeddings*) in TuBERT nutzen, um unterschiedliche Autoren auf der Grundlage ihres Schreibstils visuell zu clustern. So lässt sich ein Überblick über Kommentare und Autoren zu bestimmten Themen und deren Anteil am gesamten Datensatz bekommen. Damit leistet diese Arbeit einen wichtigen Beitrag zum Erkennen orchestriert auftretender Accounts und somit auch zur Bekämpfung Social-Media-gestützter Desinformationskampagnen.

Zu Beginn dieser Arbeit wird in Kapitel 2 der fachliche Hintergrund erläutert. Es folgt das Kapitel zur Datensatzerstellung (Kapitel 3), in dem das, im Rahmen dieser Arbeit dafür entwickelte Verfahren von der Gewinnung der noch rohen Kommentardaten über die verschiedenen Verarbeitungsschritte, bis hin zur Vorbereitung der Trainingsdaten für das BERT-Model Finetuning detailliert beschrieben wird. Zur Überprüfung der Eignung des erstellten Datensatzes NP-Set werden in Kapitel 4 mehrere Versuche mit Finetuning-Durchläufen unter variierenden Rahmenbedingungen untersucht. Die Auswertung dazu erfolgt in Kapitel 5. Anschließens wird eine weitergehende Analyse der erstellten Datensätze, NP- und Gen-Set, vorgenommen. Kapitel 6 dient der Diskussion von Methodik und Ergebnissen. Eine Betrachtung der ethischen Durchführbarkeit des in dieser Arbeit verfolgten Forschungsthemas nach Maßgabe des Gemeinsamen Ausschuss zum Umgang mit Sicherheitsrelevanter Forschung[1] ist der Diskussion vorangestellt.

[1] Gemeinsamer Ausschuss zum Umgang mit Sicherheitsrelevanter Forschung: `https://www.sicherheitsrelevante-forschung.org/`

Weiterhin wurden im Rahmen dieser Arbeit folgende Forschungsfragen analysiert:

1. Anders als bei X (ehemals Twitter) lassen sich von YouTube nicht einfach alle Beiträge eines Nutzers extrahieren. Wie gelingt es, einen Datensatz aus YouTube-Kommentaren für die Authorship Attribution zu erstellen? (Kapitel 3.1)

2. Was muss ein Datensatz bestehend aus YouTube-Kommentaren erfüllen, damit er für die Authorship Attribution mit einem BERT-Model geeignet ist? (Kapitel 3.3)

3. Was muss beachtet werden, um ein BERT-Model auf dem Datensatz zu Finetunen? (Kapitel 4.2)

4. Wie lässt sich die Eignung eines Datensatzes für die KI-gestützte Authorship Attribution mit BERT experimentell überprüfen? (Kapitel 4.3)

5. Erzielt das auf Tweets gepretrainte BERT-Model BERTweet eine höhere Genauigkeit in der Authorship Attribution von YouTube-Kommentaren, als das auf Fließtexten trainierte BERT-base? (Kapitel 4.3)

6. Enthalten auch sehr kurze Kommentare Informationen über den Schreibstil des jeweiligen Autors, durch die sich die Genauigkeit eines BERT-Models in der Authorship Attribution auf YouTube-Kommentaren erhöht? (Kapitel 4.3)

7. Wie unterscheiden sich Kommentare aus dem News & Politics - Bereich im NP-Set von den allgemeinen Kommentaren aus dem Gen-Set? (Kapitel 5.3.2)

8. Können wir TuBERT nutzen, um orchestriert auftretende Kommentarautoren zu clustern? (Kapitel 5.3.3)

2 Hintergrund

Dieses Kapitel gibt zunächst eine Einführung zu den Themen Social Media und Desinformation, der Videoplattform YouTube, der Erkennung koordinierter Aktivitäten in Sozialen Medien und der stylometriebasierten Authorship Attribution mit Large Language Models (LLMs). Anschließend werden die in dieser Arbeit verwendeten Techniken zur Sammlung von YouTube-Kommentaren, LLMs auf der Basis von BERT sowie Arbeiten mit Twitter-Postings (*Tweets*) beschrieben. Diese stammen wie die YouTube-Kommentare aus der Domäne der Sozialen Medien. Eine dieser Arbeiten behandelt die Authorship Attribution auf CNNs. In zwei weiteren Arbeiten wurde sich mit der Erkennung koordiniert auftretender Twitter-Accounts befasst. Im Rahmen dieser Arbeit eingesetzt wurden das Social-Media-Analysetool *Mozdeh* (siehe Kapitel 2.5.1) und die Large Language Models (LLMs) *BERT* und *BERTweet* (siehe Kapitel 2.5.2 und 2.5.2). *BertAA* wiederum untersuchte ebenfalls die Authorship Attribution mit BERT-Models (siehe Kapitel 2.5.2) und diente für die Versuchsgestaltung als Orientierung und für die Diskussion in Kapitel 6 als Vergleich.

2.1 Social Media und Desinformation

Soziale Medien werden als Sammelbegriff für Plattform-Angebote, die es den Nutzern basierend auf digitalen Netzwerktechnologien ermöglichen, Informationen zu teilen und Beziehungen zu anderen Nutzern aufzubauen und zu unterhalten, definiert. In den vergangenen zwanzig Jahren wurden gemeinsam mit der steigenden Verbreitung des Internets auch Social Media immer beliebter [18]. 2021 haben bereits 72 Prozent der US-Amerikaner bei einer Befragung des *Pew Research Center* angegeben, diese jemals genutzt zu haben [5]. Die damit einhergehende, wenigstens teilweise Ersetzung der institutionalisierten Massenmedien, wie dem öffentlich-rechtlichen Rundfunk als Bezugsquelle für den persönlichen Nachrichtenkonsum, führte zu großen Veränderungen in der sozialen Öffentlichkeit. [18]
Papakyriakopoulos et al. [21] haben gezeigt, dass Social-Media-Nutzer, die an politischen Online-Diskussionen teilnehmen und ihre Ansichten verbreiten, die Art und Weise beeinflussen können, in der die ursprüngliche Information von anderen Nutzern aufgenommen wird. Hyperaktive Nutzer machen zwar nur einen kleinen Teil der gesamten Nutzer aus, haben aber signifikanten Anteil an politischen Diskussionen. Sie übernehmen die Rolle von Meinungsführern und können Agenden setzen, wodurch sie für weitere Nutzer ein

alternatives Bild der öffentlichen Meinung erzeugen. [21]
Das Ziel des Betreibens von Social Media ist nicht, eine Plattform zum ausgewogenen politischen Austausch zu schaffen. Sie sollen ihre Nutzerzahl maximieren und die Nutzenden so lange wie mögliche auf der jeweiligen Plattform halten [21], indem ihnen im jeweils personalisierten Feed für sie möglichst relevante Inhalte gezeigt werden. Aufmerksamkeit ist die Währung zwischen den Produzenten und Konsumenten von Social-Media-Inhalten [15]. Die Algorithmen hinter Sozialen Medien bevorzugen Inhalte, die Nutzerinteraktionen begünstigen. Dafür müssen diese nicht unbedingt von vertrauenswürdigen Quellen stammen [25]. Böswillige Akteure können diesen Mechanismus nutzen, um gezielt Falschinformationen zu verbreiten. Im Falle der absichtlichen Verbreitung unwahrer Informationen wird von *Desinformation* gesprochen [10]. Dabei wird die Verbreitung von Desinformation unter anderem aufgrund der Sorge vor der Beeinflussung von Wahlen, wie bei der US-Präsidentschaftswahl 2016 [2], als Gefahr für demokratische Gesellschaften verstanden. Shao et al. haben gezeigt, dass dabei softwaregesteuerte Nutzeraccounts, die automatisiert Inhalte produzieren und mit anderen Nutzern interagieren (*Social Bots*) eine entscheidende Rolle spielen, da sie andere Social-Media-Nutzer täuschen können [25]. Neben der Infiltration des politischen Diskurses sind die Manipulation des Aktienmarktes sowie das Stehlen persönlicher Informationen weitere schadhafte Anwendungsfälle [9]. Dabei gilt das Blockieren beziehungsweise Entfernen vorsätzlich böswilliger Nutzer als effektiver Weg, um die Verbreitung von Desinformation und voreingenommenen Aussagen einzudämmen [21][25]. Daher ist das Erkennen von Social Bots ein wichtiges Forschungsthema, um diese Eindämmung automatisiert vornehmen zu können.

2.2 YouTube

Aufgrund der weiten Verbreitung und der Bedeutung für die öffentliche und persönliche Meinungsbildung liegt der Fokus in dieser Arbeit auf dem sozialen Werbemedium *YouTube*. Mit lokal angepassten Versionen in über 100 Ländern und Inhalten in 80 verschiedenen Sprachen erreicht YouTube jeden Monat über zwei Milliarden Nutzer weltweit [13]. Der Schwerpunkt der Plattform liegt auf Veröffentlichung, Konsum und Interaktion mit nutzergenerierten Videoinhalten. Daher gehört es zur Social-Media-Untergruppe der *user-generated-content plattforms* (*UGC-Plattforms*) und unterscheidet sich damit von *Sozialen Netzwerken* (wie *Facebook*) und Internetforen [18]. Aktuell gilt YouTube als größter Anbieter von nutzergenerierten Videoinhalten weltweit [33] und ist ebenfalls die am häufigsten genutzte Onlineplattform in den Vereinigten Staaten. 81 Prozent der US-Amerikaner gaben 2021 an, dass sie die Plattform schon einmal genutzt haben. Dies stellt einen deutlichen Anstieg seit den 73 Prozent im Jahr 2019 dar. Bei den 18- bis 29-Jährigen haben sogar

95 Prozent angegeben, die Plattform zu nutzen. [5]
Trotz der weiten Verbreitung ist YouTube, verglichen mit der *Microblogging-Plattform X* (ehemals und hier weiterhin *Twitter*), bei den wissenschaftlichen Publikationen eher unterrepräsentiert [31]. Gleichzeitig besteht aber ein hohes Potenzial zur missbräuchlichen Nutzung mit dem Ziel der Meinungsmache [21]. Deshalb wird in dieser Arbeit ein Ansatz zur Bestimmung der Autoren von YouTube-Kommentaren untersucht. Damit wird auch die Detektion koordinierter Aktivitäten in YouTube-Kommentarspalten vorbereitet. Diese könnte nicht nur der weiteren Erforschung zukünftiger Desinformationskampagnen auf YouTube dienen, sondern auch eine Grundlage für die automatisierte Löschung entsprechend böswillig agierender Nutzeraccounts schaffen.

2.3 Erkennung Koordinierter Aktivitäten

Immer häufiger kommt es in den Sozialen Medien zu koordinierten Aktionen, deren Ziel die Manipulation der öffentlichen Meinung ist [26]. Mit der fortlaufend steigenden Nutzerzahl und dem damit einhergehend steigenden Einfluss beliebter Plattforminhalte auf die öffentliche Debatte und des damit bestehenden Gefahrenpotenzials, wird auch das Erkennen koordiniert agierender Nutzeraccounts zur wichtigen Forschungsfrage. Sharma et al. [26] haben zwei charakteristische Merkmale dieser Koordination vorgestellt: *Strong hidden influence* und *highly concerted activities*. Es ist anzunehmen, dass sich diese Eigenschaften auch im ähnlichen Schreibstil und einer sich überschneidenden Themenwahl der beteiligten Accounts wiederfinden lassen [20]. Mit dieser Arbeit wird eine wissenschaftliche Grundlage gelegt, um diese Ähnlichkeiten zur Erkennung koordiniert operierender Accounts in den Kommentarspalten auf YouTube nutzen zu können. Dazu wird ein *stylometrischer* Ansatzes verwendet.

2.4 Stylometriebasierte Authorship Attribution mit LLMs

Authorship Analysis ist der Teil im Bereich der *Natural Language Processing (NLP)*, bei dem die Eigenschaften von Texten analysiert werden, um Informationen über deren Autoren zu gewinnen [8]. Stylometriebasierte Authorship Attribution bezeichnet die Aufgabe zur Bestimmung von Autoren auf der Grundlage ihrer unterscheidbaren Schreibstile und der Themen, über die sie schreiben. Dabei lassen sich stilistische Informationen sowohl auf morphologischer, als auch auf lexikaler und syntaktischer Ebene finden [27].
Für die automatisierte Analyse großer Datenmengen sind Deep Learning-Methoden erwiesenermaßen sehr gut geeignet, insbesondere, wenn es um Klassifizierungsaufgaben wie die Authorship Attribution geht [13]. Dabei bildet die Modellierung natürlicher Sprache die Grundlage zur Sprachanalyse und -generierung und wurde in den vergangenen

20 Jahren intensiv untersucht [36]. Sogenannte *Large Language Models (LLMs)* sind für die Bewältigung verschiedener Aufgaben, wie beispielsweise der Übersetzung oder Generierung von Texten, geeignet. Immer häufiger werden sie in Forschung und Industrie eingesetzt [6]. Zuletzt wurden auf großen Textkörpern gepretrainte *Transformer Models* (siehe Kapitel 2.5.2) vorgestellt, die sehr gute Ergebnisse in der Bearbeitung verschiedenster NLP-Tasks erzielen.
In dieser Arbeit wird das LLM *BERT* (siehe Kapitel 2.5.2) sowie dessen Variante *BERTweet* (siehe Kapitel 2.5.2) eingesetzt, um die Autoren von YouTube-Kommentaren basierend auf ihrem Schreibstil zu bestimmen.

2.5 Relevante Techniken

In diesem Kapitel werden die für diese Arbeit verwendeten Techniken sowie weitere für die Diskussion wichtigen Arbeiten erläutert. Zunächst werden die Techniken zur Sammlung und Analyse von YouTube-Kommentaren dargestellt. Es folgen detaillierte Ausführungen zu relevanten BERT-Models. Abschließend werden weitere Techniken zur Erkennung koordinierter Accounts in der Twitter-Domäne vorgestellt.

2.5.1 YouTube Datengewinnung

In dieser Sektion wird zuerst die YouTube-API und deren Verwendung beschrieben (Kapitel 2.5.1). Anschließend wird der Einsatz des im Rahmen dieser Arbeit verwendeten Social-Media-Analysetools *Mozdeh* erklärt (Kapitel 2.5.1), bevor weitere Techniken zur Sammlung und Analyse von YouTube-Kommentaren vorgestellt werden (Kapitel 2.5.1).

YouTube Application Programming Interface (API)

Um ein besseres Verständnis vom Verhalten besonders aktiver Kommentatoren auf YouTube durch das Finetuning eines LLMs gewinnen zu können, wird zunächst ein geeigneter Datensatz benötigt. Um aber Datensätze aus YouTube-Kommentaren erstellen zu können, benötigen wir wiederum eine Möglichkeit, Kommentardaten in großem Maßstab zu sammeln.
Eine Programmierschnittstelle (*API*) ist eine Reihe von Protokollen und Spezifikationen, die es Anwendungen erlaubt, miteinander zu kommunizieren. In dieser Arbeit wird die YouTube-API verwendet, da diese das systematische Extrahieren von Kommentaren zu YouTube-Videos ermöglicht. Ebenfalls lassen sich durch die API videobezogene Daten wie Nutzer-Interaktionen sammeln, wozu auch Kommentare gehören. [13]

Mit dem Ziel, Daten über die YouTube-API abzufragen, wurde ein Account in der *Google Cloud console*[2] angelegt. Der dort erstellte *API-KEY* ermöglicht das Stellen von *data requests* an die API. Standardnutzer sind dabei auf ein bestimmtes Anfragenkontingent begrenzt. Anstatt nun aber die für die Kommentargewinnung benötigten HTTP-Anfragen an die YouTube-API selbst zu generieren, wurde in dieser Arbeit das Social-Media-Analysetool *Mozdeh* eingesetzt, dass diese Anfragen für zuvor spezifizierte YouTube-Kanäle stellt.

Mozdeh - Big Data Analysis

Mozdeh[3] ist eine kostenlose Desktopanwendung für Windows-Systeme [30]. Entwickelt wurde sie von der *Statistical Cybermetric Research Group* an der University of Wolverhampton. Neben Tweets und Reddit-Posts lassen sich mit Mozdeh auch Nutzerkommentare zu YouTube-Videos sammeln. Darüber hinaus bietet die Software die Möglichkeit zur detaillierteren Analyse der Daten, wie einer *Sentiment Analyse*, bei der die Stimmung untersucht wird, die der jeweilige Eingabetext ausdrückt. [13]
In dieser Arbeit wurde Mozdeh aber zur reinen Gewinnung der Kommentar-Daten verwendet. Um die gewonnenen Daten, wie in Kapitel 2.4 beschrieben, für die Authorship Attribution auf dem BERT-Model nutzen zu können, bedarf es einiger Verarbeitungsschritte, welche in Kapitel 3 ausgeführt werden.

Weitere Techniken zur Sammlung von YouTube-Kommentaren

Neben der Sammlung von YouTube-Kommentaren über die API besteht auch die Möglichkeit des *Web-* oder *Screen Scrapings*. Screen Scraping beinhaltet das Auslesen der Inhalte über die Benutzeroberfläche. Auch wenn dieser Prozess durch *Web-Crawler* automatisiert werden kann, ist er häufig schwerfällig und zeitaufwendig, da enorm große Datenmengen in sehr vielen Abfragen durchsucht werden müssen. Darüber hinaus werden mit dieser Technik häufig weniger Informationen abgegriffen, als durch die APIs gewonnen werden [13]. Ebenfalls lässt sich Screen Scraping ergänzend zur YouTube-API einsetzen, um die, aufgrund der Beschränkung der API auf eine gewisse Anzahl von Anfragen je Nutzer je Tag, lange Dauer bis zum Erhalt größerer Kommentardatensätze zu verkürzen (vgl. Kapitel 2.5.1). Dieses Vorgehen wurde beispielsweise von Boddapati et al. erfolgreich umgesetzt. [4]

Die von der *Digital Methods Initiative* der Universität Amsterdam entwickelten YouTube Data Tools[4], sowie das vom *Social Media Lab* der kanadischen Ryerson University ent-

[2] Google Cloud console: `https://console.cloud.google.com` (17.06.24)
[3] Mozdeh - Big Data Analysis: `http://lexiurl.wlv.ac.uk/searcher/youtube.html` (30.04.24)
[4] YouTube Data Tools: `https://ytdt.digitalmethods.net`

wickelte, cloud-basierte NETLYTIC[5] können genauso wie Mozdeh (siehe Kapitel 2.5.1) für die Sammlung von YouTube-Daten genutzt werden [13]. Aufgrund der bedienerfreundlichen Benutzeroberfläche wurde sich in dieser Arbeit für die Nutzung von Mozdeh (siehe Kapitel 2.5.1) entschieden.

Sentimentanalyse auf YouTube-Kommentaren

Neben der stylometrischen Authorship Attribution gibt es noch weitere Methoden zur Analyse von YouTube-Kommentaren. Boddapati et al. [4] haben 2022 vorgestellt, wie sie die von YouTube-Kommentaren ausgedrückte Stimmung auf Grundlage von lexikonbasierten Techniken analysiert haben. Um das *Sentiment* der Kommentare zu bestimmen und diese als positiv, negativ oder neutral klassifizieren zu können, haben sie die Polarität der Sätze und Wörter, die ausschlaggebend für die Stimmung des Kommentars sind, bestimmt. [4]
Dieser Ansatz wird in Kapitel 5 aufgegriffen, um die Unterschiede zwischen NP-Set und Gen-Set zu analysieren.

2.5.2 BERT Language Models

In diesem Bereich werden die für diese Arbeit relevanten BERT-Models detailliert erklärt. Für die Versuchsdurchläufe in Kapitel 4.3 wurden BERT-base (siehe Kapteil 2.5.2) und BERTweet (siehe Kapitel 2.5.2) verwendet. TweetBert (siehe Kapitel 2.5.2) ist ein genauso wie BERTweet auf der BERT-Architektur basierendes Model, das für die Twitter-Domäne gepretrained wurde. *BertAA* (siehe Kapitel 2.5.2) diente für die Versuchsgestaltung in Kapitel 4.3 als Orientierung und für die Diskussion in Kapitel 6 als Vergleich. Auf die BERT-Models folgt eine Ausführung über die Authorship Attribution mit neuronalen Netzen vor der Verbreitung transformerbasierter Models. Dazu wird in Kapitel 2.5.3 das von Shreshta et al. [27] vorgestellte *Convolutional Neural Network* (CNN) zur Authorship Attribution kürzerer Texte detailliert beschrieben. Abschließend werden in Kapitel 2.5.3 weitere Arbeiten zur Erkennung koordiniert auftretender Twitter-Accounts vorgestellt.

BERT

BERT ist eine LLM-Architektur und wurde 2018 von Devlin, Chang, Lee und Toutanova veröffentlicht [7]. Die Abkürzung steht für ***B**idirectional **E**ncoder **R**epresentations from **T**ransformers*. Dabei dienen BERT-Models der Repräsentation natürlicher Sprache, die auf der von Vaswani et al. [32] vorgestellten Architektur zur Berechnung und Darstellung globaler Abhängigkeiten zwischen Ein- und Ausgabe (*Transformer*) aufgebaut wurde. Trans-

[5] Netlytic: `https://Netlytic.org`

former basieren ausschließlich auf einem *Attention Mechanism*, der als Abbildung von Anfragen und Schlüssel-Wert-Paaren beschrieben wird, deren Ausgaben jeweils Vektoren sind. Dabei wird jede Ausgabe aus der gewichteten Summe der entsprechenden Werte berechnet (*Attention Weights*). Diese dienen der Betrachtung der einzelnen Textelemente im Gesamtkontext der durch das Model repräsentierten Sprache. Dabei funktioniert dieses Verfahren unabhängig von der Distanz der betrachteten Ein- und Ausgabeelemente. Transformer sind parallelisierbar und somit auch bei langen Texteingaben performant. So zeigten Vaswani et al. beispielsweise, dass das Transformer Model Pretraining für Übersetzungsaufgaben gemessen am *BLEU-Score* (*BiLingual Evaluation Understudy*, eine Methode zur Evaluation automatisierter maschineller Übersetzung [22]) eine um bis zu 2 Punkte bessere Performance erreichte und dabei deutlich weniger Rechenleistung verbrauchte, als bisherige Architekturen, die auf *Recurrent Neural Networks* oder *Convolutional Neural Networks (CNNs)* basieren. [32]
Ausgehend von der Transformerarchitektur wurde BERT zum Erlernen tiefer natürlichsprachlicher Repräsentationen auf der Grundlage ungelabelter Texte entworfen. Gepretrained wird es auf einem Textkörper bestehend aus dem *BooksCorpus*[6] (850 Millionen Wörter) und dem englischsprachigen Wikipedia (2.500 Millionen Wörter). Dazu wurden *Masked LM*, ein Verfahren, bei dem zufällige Textelemente maskiert und vom Model erraten werden und die *Next Sentence Prediction*, bei der das Model den Folgesatz aus zwei Auswahlmöglichkeiten bestimmt, genutzt. Dies ermöglicht es dem Model, einen komplexeren beidseitigen Kontext in das Erlernen der Sprache einzubeziehen, als wenn die Trainingsdaten aus einzelnen Sätzen bestehen würden [32]. Beide Verfahren gehören zum Machine-Learning-Bereich des *Unsupervised Learnings*. Diese zeichnen sich dadurch aus, dass das Model Eingabedaten ohne die gewünschten Ausgabedaten erhält. Das Ziel dabei ist es, durch das Erkennen von Mustern eine möglichst vollständige Repräsentationen der Eingabedaten zu bekommen, welche sich beispielsweise für die Entscheidungsfindung oder die Vorhersage weiterer Eingaben nutzen lassen [11].
Ein gepretraintes BERT-Model kann trotz seiner einheitlichen Architektur für verschiedene NLP-Aufgaben gefinetunet werden. Dabei verbraucht das Finetuning im Vergleich zum Pretraining relativ wenig Rechenleistung. In dieser Arbeit wird BERT eingesetzt, um die Eignung des Datensatzes NP-Set für LLM-gestützte Authorship Attribution experimentell zu überprüfen. Dafür wurde die Authorship Attribution Task in eine *Sequence Classification Task*[7] übersetzt, in der den Eingabekommentaren das entsprechende Autoren-Label zugeordnet werden muss. Die Ein- und Ausgaben werden für die Verarbeitung im Model durch Token-Embeddings dargestellt. Als *Token* verstehen wir analog

[6] BookCorpus: `https://paperswithcode.com/dataset/bookcorpus` (22.06.24)
[7] Sequence Classification: `https://huggingface.co/transformers/v3.4.0/task_summary.html#sequence-classification` (22.06.24)

zu *Word Pieces* nach Wu et al. übliche Teilworte [34]. Diese begünstigen die Verarbeitung von seltenen, dem Model unbekannten Wörtern in Ein- und Ausgabetexten, da sich diese aus mehreren Tokens zusammensetzen lassen. Das Vokabular dazu umfasst 30.000 dieser *Tokens*. Der Beginn jeder Textsequenz wird durch ein spezielles Klassifikations-Token *[CLS]* markiert. Der letzte, diesem Token zugehörige *hidden state* wird als aggregierte Sequenzdarstellung für Klassifizierungsausgaben verwendet (siehe Abbildung2.2). Das Ende jeder Sequenz wird durch den speziellen Token *[SEP]* gekennzeichnet. Für jeden Token einer Sequenz wird die Eingabe als Summe aus den zugehörigen Token-, Segment- und Positions-Embeddings dargestellt. Für das Finetuning müssen lediglich die aufga-

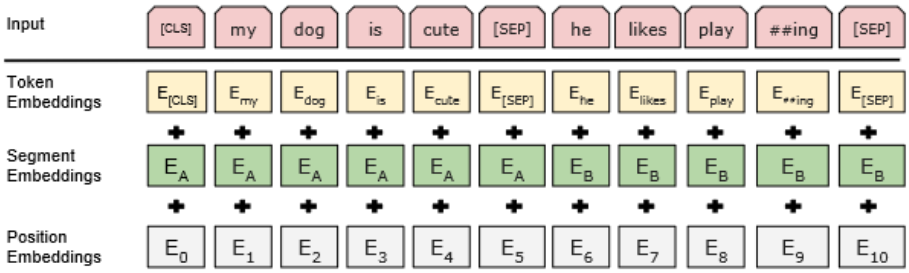


Abbildung 2.1: BERT Eingaberepräsentation: Die Eingabe-Embeddings sind die Summe der Token-Embeddings, Segment-Embeddings und entsprechenden Positionen in der Eingabesequenz. (Abbildung 2 aus BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Devlin et al., 2019)

benspezifischen Ein- und Ausgaben an das BERT-Model übergeben und die Parameter entsprechend angepasst werden, da der *Self Attention Mechanism* das Transformer-Model BERT flexibel von den Daten lernen lässt. Für Classification Tasks wird nur die [CLS]-Repräsentation an das Output-Layer übergeben, auf dessen Grundlage dann die Vorhersage des Labels stattfindet. (siehe Abbildung 2.2). [7]

Da BERT-base auf englischsprachiger Literatur und Wikipedia gepretrained wurde, wurde zusätzlich ein weiteres BERT-Model, welches auf Twitter-Postings (*Tweets*) gepretrained wurde, für vergleichende Experimente herangezogen. Es kann vermutet werden, dass sich mit *BERTweet* eine höhere Genauigkeit in Classification Tasks erreichen lässt, da Tweets YouTube-Kommentaren vom Textformat ähnlicher als Fließtexte sind und das Model daher durch die erlernte Sprachrepräsentation in den Classification Tasks besser abschneiden könnte. Diese Vermutung wird in Kapitel 4 experimentell überprüft, indem sowohl BERT-base als auch BERTweet auf die gleiche Art und Weise gefinetuned und deren erzielte Genauigkeit in der Classification Task verglichen werden.

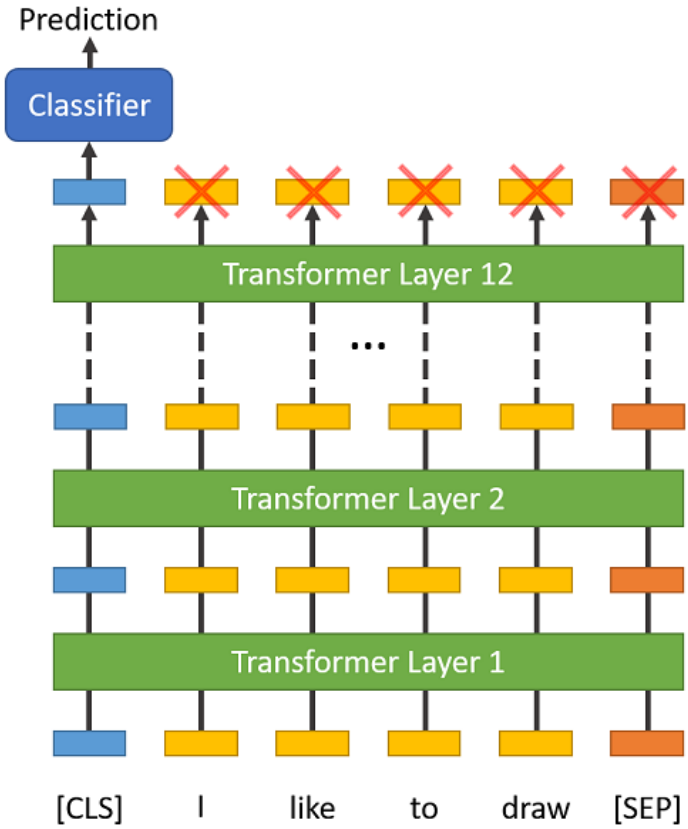


Abbildung 2.2: Schema der Verarbeitungsschritte der Token-Repräsentationen für die BERT Sequence Classification am Beispiel. (von Jaskaran Singhs BERT Fine-Tuning with PyTorch, 29.06.24)

BERTweet

BERTweet ist das erste öffentliche Sprachmodell, dass in großem Maßstab auf englischsprachigen Tweets gepretrained wurde. Das zugehörige Paper wurde 2020 von Nguyen, Vu und Nguyen veröffentlicht [19]. Es besteht aus der BERT-base - Architektur (siehe 2.5.2) und wurde auf 850 Millionen Tweets mit einer Länge zwischen 10 und 64 Wort-Tokens gepretrained. Dabei wurde nach der *RoBERTa*-pretraining-procedure trainiert, die in der 2019 von Liu et al. [16] veröffentlichten Arbeit zu RoBERTa vorgestellt wurde. In dieser Replikationsstudie zum BERT-pretraining (siehe Kapitel 2.5.2) wurden einige der Pretraining-Designentscheidungen von Devlin et al. [7] überprüft. Die Autoren fanden heraus, dass sich die Model Performance durch längeres Training, größere Batches, mehr Trainingsdaten und einiger weiterer Anpassungen substanziell steigern lässt. Damit beheben Nguyen et al. mit BERTweet einige Schwächen in der Anwendung von auf konventionellen Textkörpern gepretrainten Sprachmodellen für die Analyse deutlich kürze-

rer Texter, wie Tweets. Das Finetuning findet über 30 Epochen und einer *Batchsize* von 32 statt. Die Autoren zeigen experimentell, dass BERTweet das auf gleiche Art gefinetunete Model RoBERTa [16] in verschiedenen NLP Tasks auf Tweets an Genauigkeit übertrifft. [19]
Damit haben Nguyen et al. gezeigt, dass es für die Lösung von NLP Tasks von Vorteil ist, auf Models zurückzugreifen, die den, für Finetuning und Classification verwendeten Eingabedaten möglichst ähnlich sind. Ob das Pretraining auf Tweets bei BERTweet dieses Model in die Lage versetzt, die Authorship Attribution von YouTube-Kommentaren mit höherer Genauigkeit durchzuführen, als das auf Fließtexten trainierte BERT-base, wird in Kapitel 4.3 experimentell überprüft.

TweetBERT

TweetBERT ist genauso wie BERTweet (Kapitel 2.5.2) ein auf der BERT-Architektur (Kapitel 2.5.2) basierendes LLM, das für die Twitter-Domäne trainiert wurde. Veröffentlicht wurde es 2020 von Qudar und Mago [23] in zwei unterschiedlichen Versionen. TweetBERT (v1) wurde auf 140 Millionen und TweetBERT (v2) auf 540 Millionen Tweets von den 100 Twitter-Accounts mit den meisten Followern und den 100 reichweitenstärksten Twitter-Hashtags trainiert. Dabei wurden die Trainingsdatensätze von der *Big Data Analytics Platform* der kanadischen *Lakehead University* [17] erstellt.
Gefinetuned wurde das Model für die Sentiment-Analyse und Classification Tasks. Dabei zielt diese Classification nicht auf die Autorenbestimmung, sondern auf andere Klassifizierungsaufgaben wie die *gender classification* ab. In Ihren Experimenten ist es den Autoren gelungen, auf Fließtexten gepretrainte BERT-Models wie AIBERT [14] für die Twitter-Domäne deutlich zu übertreffen. [23] TweetBERT fand in dieser Abschlussarbeit keine Verwendung, da die Nutzung von BERTweet ausführlicher erläutert wird[8] und über die *Huggingface*-Transformer-API in Python einfach importierbar ist. Somit konnte die Implementierung des Finetuning-Procedures analog zu der mit BERT-base erfolgen.

BertAA

Weder in der Arbeit zu BERT von Devlin et al. [7], noch in der Arbeit zu BERTweet von Nguyen et al. [19] wurde die Performance der vorgestellten Models für eine Authorship Attribution Task untersucht. *BertAA* [8] wiederum ist ein gefinetunetes BERT-Model (vgl. Kapitel 2.5.2), welches zur Authorship Attribution erweitert wurde. Es wurde von Fabien, Villatoro-Tello, Motlicek und Parida 2020 vorgestellt [8] und ist einer der ersten Ansätze

[8] BERTweet verwenden: `https://huggingface.co/docs/transformers/en/model_doc/bertweet`, (29.06.24)

zur Untersuchung der Genauigkeit gefinetuneter LLMs für eine größere Anzahl an Autoren (bis zu 100). Da in dieser Abschlussarbeit ebenfalls das Finetuning eines BERT-Models zur Authorship Attribution von kürzeren Texten vorgenommen wird, ist die Arbeit von Fabien et al. an dieser Stelle als relevante Arbeit aufgeführt.
Fabien et al. haben ihre Experimente auf drei verschiedenen Datensätzen durchgeführt: E-Mails, Filmbewertungen und Blogbeiträgen. Dazu wurden jeweils die Autoren mit den meisten Texten für die Betrachtung ausgewählt. 20 Prozent der für das Finetuning genutzten Daten wurden als Test-Datensatz verwendet. Insgesamt hat diese Untersuchung gezeigt, dass BertAA auch auf relativ kurzen Eingabetexten und bei Unausgewogenheit in der Text-Anzahl je Autor eine hohe Genauigkeit erzielt, solange die Anzahl von Texten je Autor ausreichend groß ist. Bei der Auswahl der zu betrachtenden Autoren, der 8:2-Aufteilung der Finetuning-Sets in Trainings- und Testdaten und dem späteren Verwerfen von Kommentaren mit weniger als 10 Wörtern, wurde sich in dieser Abschlussarbeit an Fabien et al. orientiert. In Kapitel 6 werden einige der Versuchsergebnisse mit denen von BertAA verglichen. [8]

2.5.3 Weitere Techniken zur Detektion koordinierter Accounts auf Twitter

In dieser Sektion wird zunächst die Authorship Attribution mit CNNs anhand eines Beispiels beschrieben, bevor weitere Ansätze zur Erkennung koordiniert agierender Twitter-Nutzer erläutert werden.

Authorship Attribution mit CNNs

Bevor BERT und weitere Transformer Models in den Vordergrund rückten, waren *Convolutional Neural Networks (CNNs)* ein sehr performanter Ansatz zur Bestimmung von Autoren. Beispielsweise haben Shrestha, Rosso und weitere 2017 ein eigenes CNN über *character n-grams* zur *Authorship Attribution* auf kurzen Texten vorgestellt [27]. Dabei werden die *character* und *word n-grams* durch Erfassen von Syntax und Stil bei der Autorenbestimmung genutzt. Die von Shreshta et al. vorgestellte Netzwerk-Architektur erlernt die Text-Repräsentation von der Ebene der einzelnen Zeichen an und nimmt dabei die lokalen Interaktionen zwischen diesen Zeichen, um sie später zu komplexeren Mustern zu aggregieren, aus welchen wiederum der Schreibstil des Autors bestimmt wird.
Der dem Einsatz von CNNs dabei inhärente sequenzielle Ablauf verhindert eine weitreichende Parallelisierung des Trainingsprozesses, wodurch insbesondere bei längeren Eingabesequenzen ein deutlich höherer Bedarf an Rechen- und Speicherkapazität im Vergleich zur Transformer-Architektur besteht. Die Berechnung der Abhängigkeiten der jeweiligen Elemente ist von der Distanz innerhalb der Ein- und Ausgabesequenzen ab-

hängig, sodass auch die Speicherbegrenzungen zur geringeren Performance gegenüber Transformer-basierten Architekturen wie BERT (siehe Kapitel 2.5.2) beiträgt. [32]
Evaluiert haben Shreshta et al. ihren Ansatz auf einem Datensatz aus rund 9.000 Twitter-Nutzern mit jeweils bis zu 1.000 Tweets. Dabei erzielte das CNN-Bigram Model die besten Ergebnisse. Auch wenn die Autorenbestimmung für steigende Autorenzahlen und weniger Tweets je Autor schwieriger wird, so haben sie doch für die Bestimmung von 1.000 Autoren noch eine Genauigkeit von über 36 Prozent erzielt. In ihrer Arbeit haben Shreshta et al. herausgefunden, dass sich fast 30 Prozent der Kommentar-Autoren durch wiederholende Muster aus Neuigkeiten, Werbung und Links wie automatisiert verhielten. Weiterhin wurde untersucht, welche Art von n-grams für das Model insgesamt die größte Bedeutung hat. Viele dieser top n-grams waren ungewöhnliche Versionen von Emoticons, die wohl zur persönlichen Vorlieben der jeweiligen Autoren gehören. [27]
Um die grundsätzliche Wirksamkeit von NP-Set und der in den Versuchen eingesetzten Models BERT-base und BERTweet zu überprüfen, wurde die Genauigkeit eines n-gram-betrachtenden CNNs für dieselbe Authorship Attribution Task ermittelt (siehe Kapitel 4).

Weitere Ansätze zur Erkennung koordinierter Twitternutzer

Im Vergleich zu YouTube-Kommentaren besteht eine sehr große Bandbreite an Forschung zu und mit Tweets [31]. Orlov und Litvak haben in ihrer 2019 veröffentlichten Arbeit [20] einen Ansatz zur Erkennung von Propagandisten und Desinformanten auf Twitter vorgestellt. Dabei definieren sie Propagandisten als Gruppe von Nutzern, die mutwillig Desinformation oder voreingenommene Aussagen verbreiten. Um Nutzer als solche zu erkennen, wurde sowohl das Nutzungsverhalten, als auch der Inhalt ihrer Texte in einem unbeaufsichtigten Verfahren (siehe *Unsupervised Learning* in Kapitel 2.5.2) untersucht. Zur Verarbeitung der Texte haben die Autoren die gesammelten Tweets normalisiert, indem sie Stoppwörter (häufig vorkommende, für den Textinhalt meist nicht relevante Wörter), Ziffern, Symbole, Emoticons und Links entfernt haben. Zusätzlich wurden die verbliebenen Wörter durch sogenanntes *Stemming* in ihre Grundform überführt (beispielsweise von *plays* zu *play*) [28]. Die Tweets in den so gestalteten Datensätze wurden dann nach ihren Veröffentlichungszeitpunkten in zeitliche Rahmen untergliedert und innerhalb jedes Rahmens geclustert. Als Nutzergruppen wurde dann die Mengen an Nutzern identifiziert, die wiederkehrend nah beieinander im Cluster aufgetreten sind. [20]

Einen weiteren Ansatz zur Erkennung koordiniert agierender Social-Media-Nutzer haben Sharma, Zhang, Ferrara und Liu 2021 vorgestellt [26]. Das generative Model AMDN-HAGE (*Attentive Mixture Density Network with Hidden Account Group Estimation*) modelliert Nutzeraktivitäten und *hidden group behaviours* gemeinsam. Es ist ihnen gelungen, in-

härente Eigenschaften der Koordination aufzuzeigen, denn koordiniert auftretende Nutzer beeinflussen ihre Aktivitäten gegenseitig und müssen somit kollektiv vom Verhalten der Masse aller Nutzer abweichen. Das Model trifft auf der Grundlage der bisherigen Aktivitäten eines Nutzers Vorhersagen über dessen zukünftige Aktivitäten und modelliert dabei die Nutzerverteilung. Während sich damit mutmaßliche Gruppenzugehörigkeiten der einzelnen Nutzer lernen lassen, wird dabei ebenfalls das kollektiv abweichende Verhalten erfasst. Um ihren Forschungsansatz zu verifizieren, haben die Autoren ihr Model auf Twitter-Daten getestet, die in Verbindung mit koordinierten Kampagnen der *Russian Internet Research Agency* zur Beeinflussung der US-Präsidentschaftswahl 2016 stehen. Dabei konnten sie bestätigen, dass der durchschnittliche wechselseitige Einfluss von denjenigen Nutzern am höchsten ist, die auch an diesen Kampagnen beteiligt waren. [26]

3 Datensatzerstellung

In diesem Kapitel werden folgende Forschungsfragen beantwortet: Wie gelingt es, einen Datensatz aus YouTube-Kommentaren für die Authorship Attribution zu erstellen (Kapitel 3.1 und 3.2) und was muss ein Datensatz bestehend aus YouTube-Kommentaren erfüllen, damit er für die Authorship Attribution mit einem BERT-Model geeignet ist (Kapitel 3.3)? Dazu wird die Erstellung des Hauptdatensatzes NP-Set und des Vergleichsdatensatzes Gen-Set schrittweise nachvollzogen. Zunächst wird die Sammlung der noch rohen Kommentardaten beschrieben und wie die gewonnenen Daten insbesondere aufgrund der personenbezogenen Anteile darunter geschützt werden. Da die Datensätze für das Finetuning verschiedener BERT-Models genutzt werden, wurde die Kommentar-Verarbeitung in dieser Arbeit noch einmal in *Vorverarbeitung* (Kapitel 3.2) und *Aufbereitung der Daten für das Finetuning* (Kapitel 3.3) untergliedert. Im Bereich zur Vorverarbeitung wird zunächst das dazu erstellte Python-Programm beschrieben. Innerhalb des Kapitels zur Aufbereitung der Daten für das Finetuning führen wir ebenfalls durch den entsprechenden Programmablauf. Aus beiden Programmen ergeben sich die notwendigen Schritte zu Nutzbarmachung der im Rahmen dieser Arbeit gewonnenen Kommentar-Daten für das Finetuning von BERT-Models.

3.1 Rohdaten-Gewinnung

In dieser Sektion wird das in dieser Arbeit verwendete Verfahren zur Gewinnung der, für die Erstellung für das Finetuning von BERT-Models auf YouTube-Kommentaren geeigneten Datensätze erläutert. Dazu wird zunächst die Nutzung der YouTube-API beschrieben, bevor geschildert wird, wie die YouTube-Kanäle für die Datensammlung ausgewählt wurden. Anschließend wird der Einsatz des Social-Media-Analysetools Mozdeh (siehe Kapitel 2.5.1) beschrieben, bevor dargelegt wird, wie die persönlichen Daten der Kommentarautoren im Rahmen dieser Arbeit geschützt wurden.

3.1.1 Nutzungsstrategie für die YouTube-API

In Kapitel 2 wurde bereits ausgeführt, dass für das Sammeln von YouTube-Kommentaren die YouTube API verwendet werden kann. Nichtkommerziellen Nutzern wird für ihren persönlichen *API-Key* eine begrenzte Anzahl an Abfrage-Tokens pro Tag gewährt. Durch das Strecken der Kommentar-Gewinnung über einen längeren Zeitraum von mehreren

Wochen konnten die Kommentare zur Erstellung der Datensätze kostenfrei über Google-Developers und die Programmierschnittstelle gewonnen werden. Anders als bei Boddapati et al. (vgl. Kapitel 2.5.1) wurde für die Sammlung der YouTube-Kommentare nicht auf ergänzendes Web-Crawling zurückgegriffen, da die Menge der über die YouTube-API über einige Zeit erhaltenen Kommentare für die Zielsetzung dieser Arbeit als ausreichend angenommen wurde. Außerdem liegen auf diese Weise alle Kommentardaten im selben Format vor, was die einheitliche Verarbeitung ermöglicht.

3.1.2 Auswahl der YouTube-Kanäle

Im Gegensatz zu Twitter lässt sich auf YouTube nicht einfach auf alle Texte eines Autors über dessen Profil zugreifen. Stattdessen müssen die Kommentare zu einzelnen Videos oder Kanälen abgefragt werden. Aufgrund des umfangreichen Forschungsstands bezüglich des Einsatzes von LLMs auf der englischen Sprache und entsprechenden Tweets (unter anderem [19], [23], [27], [26]) sowie der aktuellen Relevanz im Zuge der Sorge vor Desinformationskampagnen vor der anstehenden US-Präsidentschaftswahl, wurde sich in dieser Arbeit für eine Betrachtung von amerikanischen YouTube-Kanälen entschieden, die sich mit aktuellen Nachrichten und Politik beschäftigen. Diese Arbeit ist eine Grundlage zur Detektion koordinierter Nutzeraccounts und Social Bots, um Vorgehen und Wirken besser analysieren zu können und somit einen Beitrag zur Verhütung des missbräuchlichen Einsatzes Sozialer Medien zu leisten (vgl. Kapitel 2.1). Ziel war es also, zunächst Kanäle zu finden, auf denen möglichst viel kommentiert wird und deren Inhalte Nachrichten und Politik behandeln. Dazu wurde die Influencer Marketing-Plattform *Hypeauditor*[9] eingesetzt. Dort wurden die meistkommentierten Kanäle aus dem News & Politics Bereich in den Vereinigten Staaten ausgewählt, um aus den dort gesammelten Kommentaren den Hauptdatensatz NP-Set erstellen zu können. Für den Vergleichsdatensatz Gen-Set wurden die Kanäle mit den insgesamt meisten Kommentaren gewählt, allerdings ohne die Filterung nach dem News & Politics – Bereich. Tabelle 3.1 listet die jeweils gewählten YouTube-Kanäle. Wie dort zu sehen ist, ist kein Kanal in beiden Sets enthalten, sodass es auch keine Schnittmengen bei den enthaltenen Kommentaren geben kann. Mittels eines zweiten Unterstützungstools *Comment Picker*[10],wurden zu den gewählten Kanälen die jeweils einmaligen Channel-IDs bestimmt, welche wir für die präzise Abfrage der Kommentare durch das von uns eingesetzte Datengewinnungstool Mozdeh über die YouTube-API benötigen.

[9] Hypeauditor: `https://hypeauditor.com/` (16.06.24)
[10] Comment Picker: `https://commentpicker.com/youtube-channel-id.php` (16.06.24)

NP-Set	Gen-Set
Nexpo	llegendaryjay
BrandonHerrera	J97imjack
Thought Slime	Adamsfyp
PeterSantenello	GLITCH
Thunderf00t	Adams_fypReacts
AlexBeckersChannel	kendricklamar
OrdinaryThings	MrBeast
SecondThought	ExploreWithUs
MrBallen	cyclopentasiloxane
MarkDice	macklemore
rossmanngroup	jaidenanimations
DonutOperator	SpindleHorse
AuditTheAudit	JennyNicholson
DontWalkRun	samandcolby
BetterBachelor	MelanieMartinez
blancolirio	jordanmatter
PhilipDeFranco	KaiCenat
AmagansettPress	DanTDM
MrMobile	alanbecker
TheOfficerTatum	jschlattLIVE
CaspianReport	miniminuteman773
DailyMailRoyals	SMLMovies
JakeBroe	@BlacktailStudio
LegalEagle	SyntheticMan

Tabelle 3.1: Übersicht über die gewählte YouTube-Kanäle für NP-Set (links) und Gen-Set (rechts)

3.1.3 Datengewinnungs-Tool

Für die Untersuchung von YouTube-Kommentaren gibt es eine Reihe von bestehenden Tools, auf die Wissenschaftler zurückgreifen können, um die YouTube-API automatisiert abzufragen [3], [13]. In dieser Arbeit wurde sich aufgrund der einfachen Bedienbarkeit für den Einsatz von Mozdeh (siehe Kapitel 2.5.1) zum Stellen der Anfragen an die YouTube-API entschieden. Aufgrund des in der Anfragemenge begrenzten Zugangs zu Kommentardaten über die YouTube-API, wurden jeweils aus den gewählten Kanälen (siehe Tabelle 3.1) für alle ab dem ersten Januar 2024 hochgeladenen Videos die Kommentare mit Mozdeh gesammelt. Weiterhin wurde die Abfrage so konfiguriert, dass mehrere Kommentare je Nutzer zugelassen bleiben und auch eine nicht anonymisierte Kopie der originalen Kommentare erhalten wird. Standardmäßig ist von Mozdeh die Analyse der anonymisierten Daten vorgesehen. Die weitere Verarbeitung der gewonnenen Daten wurde für diese

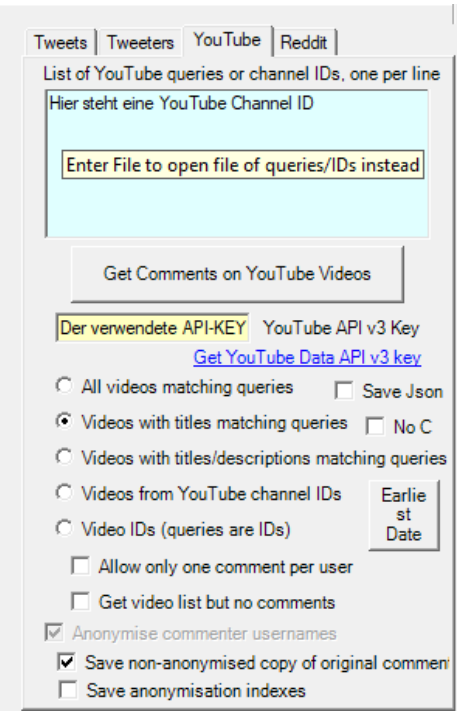


Abbildung 3.1: Screenshot einer beispielhaften Mozdeh Anfragenkonfigurierung

Arbeit aber auf den nicht anonymisierten Backup-Daten vorgenommen worden, weil die Mozdeh-Pseudonymisierung einfach die Nutzernamen durch einen entsprechend aufsteigenden Index als Pseudonym ersetzt. Aufgrund der Limitierung des nichtkommerziellen API-Zugangs wurden im Rahmen dieser Arbeit aber viele einzelne Abfragen gestartet, für die Mozdeh jeweils angefangen hätte, neu zu indizieren. Auf diese Weise wäre es nicht mehr ohne Weiteres möglich gewesen, die Autoren im Gesamtdatensatz zu unterscheiden und ihre Labels für die Classification Task wählen zu können. Während der API-Anfragenstellung durch Mozdeh sind einige Kanäle aufgefallen, für die keine Datengewinnung erfolgen konnte. Möglicherweise lässt der Betreiber für diese den API-Zugriff blockieren. Besonders die Gewinnung der Kommentare für das Gen-Set wurde hierdurch erschwert. Die betroffenen Kanäle wurden für die weitere Arbeit verworfen und sind auch in Tabelle 3.1 bereits nicht enthalten. Da über die YouTube-Accounts der kommentierenden Autoren eine Identifizierung der tatsächlichen Personen möglich sein kann, wurden die gesammelten Kommentardaten als personenbezogene Daten betrachtet, welche es zu schützen gilt.

3.1.4 Schutz peronenbezogener Daten

Um die persönlichen Daten der Kommentarautoren zu schützen, wurde im ersten Nachverarbeitungsprozess eine Pseudonymisierung eingebaut. Anders als bei Mozdeh implementiert, wird für jeden Autor ein globales und zufällig generiertes Pseudonym erstellt, welches die Nutzer-ID für die weitere Verarbeitung ersetzt. Somit blieb eine individuelle Zuordnung der Kommentare zu einem bestimmten Nutzer auch dann erhalten, wenn dieser auf verschiedenen Kanälen kommentiert hat und diese Kommentare in verschiedenen Abfragen gesammelt wurden. Alle im Verlauf dieser Arbeit eingesetzten BERT-Models erhielten nur die gelabelten Kommentare ohne Pseudonym oder Nutzer-ID. Darüber hinaus sind alle gesammelten Informationen auf verschlüsselten Systemen innerhalb der Universität zu Lübeck gespeichert, sodass sich, außer für die Forschenden, keine Möglichkeit zum Zugriff auf diese Daten ergibt. Um den Schutz der aus den Kommentaren gewinnbaren persönlichen Daten der Nutzer (wie beispielsweise politische Ansichten) sicherzustellen, werden weder die erstellten Datensätze noch die mit ihnen trainierten Models veröffentlicht.

3.2 Daten-Vorverarbeitung

Um die gesammelten Kommentare zu für das BERT-Finetuning nutzen zu können, ist eine mehrschrittige Verarbeitung notwendig. Zunächst wurden die aus den verschiedenen Mozdeh-Abfragen stammenden nicht-anonymisierten Backup-Files in ein gemeinsames Verzeichnis überführt. Alle Kommentare wurden normalisiert und dann zu einem gemeinsamen Datensatz kombiniert. Dazu wurde ein Python-Programm geschrieben, welches auf GitHub vollständig einsehbar ist und hier zur Nachvollziehbarkeit als auf das Wesentliche reduzierter Pseudocode abgebildet ist. Nicht enthalten sind in der reduzierten Variante unter anderem die Programmteile zur Ermittlung der Anzahl der Kommentare je Autor zur Bestimmung der besonders aktiven Nutzer.

Listing 3.1: Vorverarbeitung

```
1  INITIALIZE source_dir = 'path_to_source_files\\'
2  INITIALIZE target_dir = 'path_to_output_location\\'
3  INITIALIZE max_length = 126
4
5  FUNCTION process_file(filename):
6      READ LINES FROM source_dir + filename INTO source_data, SKIPPING FIRST LINE
7      FOR EACH line IN source_data:
8          SET comment, author = process_line(line)
9              IF author NOT IN pseudonym_dict:
10                 ASSIGN RANDOM VALUE BETWEEN 1 AND 10000000 TO pseudonym_dict[author]
```

```
11              SET pseudo_author = pseudonym_dict[author]
12              REPLACE author IN line WITH pseudo_author
13          SET line = dsc.cosmetics(line)
14          IF validate_line(line):
15              APPEND line TO output_lines
16          ELSE:
17              SPLIT comment USING dsc.len_split(comment, pseudo_author, max_length)
18              FOR EACH split_comment IN split_comments:
19                  APPEND split_comment + '\t' + pseudo_author TO output_lines
20      CALL save_output(output_lines, filename)
21
22  FUNCTION main():
23      OPEN target_dir + output_filename FOR APPENDING AS complete_output
24      FOR EACH filename IN LIST OF FILES IN source_dir:
25          IF filename ENDSWITH '.bak':
26              CALL process_file(filename)
```

Im Programm müssen zunächst das Quellverzeichnis mit den nicht anonymisierten Mozdeh-Dateien und das Zielverzeichnis für den dann erstellten Datensatz spezifiziert werden [Z.1-2]. Anschließend ist noch die maximale Anzahl an Tokens (*max_length*) anzugeben [Z.3], um schon in diesem Verarbeitungsschritt eine erste Aufteilung zu langer Kommentare vornehmen zu können. Das im Rahmen dieser Arbeit zunächst eingesetzte BERT-base Model (siehe Kapitel 2.5.2) fasst eine Eingabelänge von bis zu 512 Tokens. Weil in dieser Arbeit ebenfalls das BERTweet-Model (siehe Kapitel 2.5.2) einsetzt wird, welches auf 128 Eingabetokens ausgelegt ist, wurde die Länge aller Kommentare sowohl im NP-, als auch im Gen-Set auf 128 Tokens begrenzt, um eine einheitliche Verarbeitung zu ermöglichen. Wie in Kapitel 2.5.2 beschrieben, müssen die Kommentare für die Verarbeitung in BERT-Models in Tokens aufgeteilt werden. Im Zuge dieser *Tokenization* werden jeweils noch die BERT-eigenen Special-Tokens ([CLS] zum Beginn der Sequenz und [SEP] am Ende) ergänzt. Daher ist die hier konfigurierte Länge 128 – 2 = 126. Nacheinander geht das Programm jede Quelldatei im Quellverzeichnis durch [Z.24f]. Zunächst werden die Dateien zeilenweise eingelesen und die Kopfzeile mit den Spalten-Bezeichnungen entfernt [Z.6]. Da die Mozdeh-Abfragen neben den für die Experimente lediglich benötigten Kommentar-Texten und den Autoren-Kennungen noch eine Reihe Metadaten mitliefern, werden die Daten aus den nicht benötigten Spalten verworfen [Z.8]. Dafür wird in der Methode *process_line()* jede Zeile spaltenweise reduziert, bis nur noch Kommentar und Autor übrig sind, welche dann getrennt zurückgegeben werden (siehe Abbildung 3.2). Nun folgt die Pseudonymisierung der Nutzer-IDs. Dazu wird ein Dictionary für die IDs und die entsprechend zugehörigen Pseudonyme angelegt. Sollte eine Nutzer-ID noch nicht verzeichnet sein, so wird für diese eine zufällige Zahl zwischen 1 und 10.000.000 als Pseudonym bestimmt und entsprechend gespeichert. Die zum aktuellen Kommentar

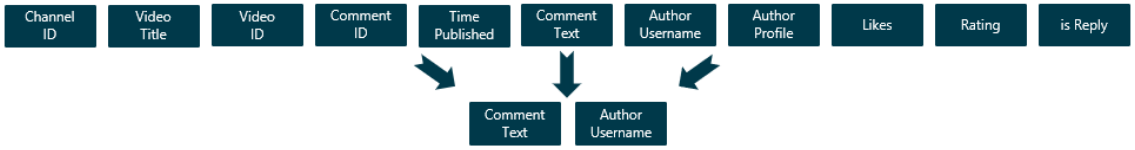


Abbildung 3.2: Reduzierung der gesammelten Kommentardaten auf Kommentar-Text und Author (Vereinfachte Darstellung)

gehörende Nutzer-ID wird nun durch das entsprechende Pseudonym ersetzt [Z.9-12]. Weiterhin wird nun eine weitere Hilfsfunktion *cosmetics()* aufgerufen [Z.13]. Diese ist von der Tweet-Normalisierung von BERTweet[11] inspiriert. In der Funktion werden einige Sonderzeichen entfernt, alle Emoticons in eine Klartextrepräsentation überführt, URLs durch „httpurl“ ersetzt und Direktantworten und Erwähnungen mit dem Format „@user“ oder „@@user“ zu „_user“ vereinheitlicht (siehe Listing 3.2). Im Gegensatz zu anderen Arbeiten ([19], [20]) wurde mit dieser eher behutsamen Normalisierung versucht, möglichst viel vom ursprünglichen Kommentar und somit für die Models nutzbare Informationen zu erhalten. So bleiben Emoticons und Interpunktion bestehen und die URLs werden generalisiert, anstatt sie einfach zu entfernen.

Listing 3.2: cosmetics() als vereinfachter Pseudocode

```
1  Function cosmetics(lines: string) -> string:
2      Replace all '@@' with '@' in lines
3      Remove all '\u200b' and '\u003c3' from lines
4      Apply demojize on lines
5      If 'http' is present in lines:
6          Extract link from 'http' to the tab character ('\t') in lines
7          Truncate link at the first space, if present
8          Replace link with 'httpurl' in lines
9      While '@' is present in lines:
10         Extract mention from '@' to the tab character ('\t') in lines
11         Replace mention with '_user' in lines
12     Return lines
```

Da angenommen wurde, dass auch sehr kurze Kommentare zum individuellen Kommentierstil eines Autors beitragen und somit für das Model-Finetuning relevante Informationen enthalten, wurde sich zunächst, anders als bei BERTweet (siehe Kapitel 2.5.2) und BertAA (siehe Kapitel 2.5.2) dagegen entschieden, Kommentare unterhalb einer gewissen Minimallänge zu verwerfen. In Kapitel 4.3 wurde diese Annahme überprüft, indem die Models jeweils auf Datensatzvarianten inklusive und exklusive der sehr kurzen Kom-

[11] BERTweet Tweet Normalizer: `https://github.com/VinAIResearch/BERTweet/blob/master/TweetNormalizer.py`, (30.06.24)

mentare für die Authorship Attribution gefinetuned wurden. Allerdings war die erzielte Genauigkeit bei den Label-Predictions der Modelle ohne sehr kurze Kommentare im Finetuning-Set höher (vgl. Tabelle 4.2), weshalb die Annahme widerlegt wurde.
Wenn ein Kommentar nun aus weniger Wörtern als die zuvor festgelegte Maximallänge besteht, wird die aktuelle Zeile direkt der Zieldatei hinzugefügt [Z.14-15]. Ist der Kommentar länger als die zuvor definierte *max_len*, wird die Funktion *len_split()* aufgerufen. In dieser wird der Kommentar anhand seiner Satzzeichen in möglichst sinnvolle, nun kürzere Kommentare unterteilt [Z.16-19]. Über alle gesammelten Kommentare wird zusätzlich noch eine Datei angelegt, in der alle Pseudonyme jeweils mit der Anzahl ihrer Kommentare im Datensatz gespeichert sind. Außerdem wird noch eine Liste mit allen Autoren, von denen mindestens 100 Kommentare in der Datei vorhanden sind, erstellt. Diese wird für die Vorbereitung unserer Datensätze für das Finetuning benötigt, da wie bei BertAA (vgl. Kapitel 2.5.2) für die Finetuning-Sets zur Authorship Attribution nur diejenigen Kommentare der Autoren infrage kommen, für die genügend Kommentare vorliegen, um den individuellen Stil eines Autors zu bestimmen. Für die Versuche im Rahmen dieser Arbeit standen nur begrenzte Rechenkapazitäten zur Verfügung (siehe Kapitel 4.1). Mit steigender Anzahl der Classification Label nimmt aber auch die Rechenkomplexität für das Finetuning zu, da vom Model für jedes Label eine eigene Repräsentation im globalen Kontext erlernt und verarbeitet werden muss. Daher ist eine Begrenzung der Anzahl der betrachteten Autoren notwendig, um jedem für die Classification Task ein eigenes Label zuweisen und die Versuche in der für diese Arbeit zur Verfügung stehenden Hardwareumgebung durchführen zu können.

Das NP-Set aus Kommentaren zu Videos auf dem News & Politics - Bereich besteht nun aus 4.422.537 Kommentaren mit jeweiligem Autorenpseudonym. Von den 1.444.038 gelisteten unterschiedlichen Autoren haben lediglich 745 mindestens 100 Kommentare verfasst. Das entspricht einem Anteil von 0,052 Prozent, was den Befund von Papakyriakopoulos et al. bestätigt, nachdem nur ein sehr kleiner Teil der Nutzer Sozialer Medien zu den *hyperaktiven Nutzern* gezählt werden kann [21].
Das Gen-Set aus Kommentaren zu Videos ohne thematische Eingrenzung umfasst 1.453.465 Kommentare von 552.998 Autoren.

3.3 Aufbereitung der Daten für das Finetuning

Um die Datensätze für das Finetuning der BERT-Models nutzen zu können, sind noch weitere Verarbeitungsschritte nötig, für das ebenfalls ein eigenes Python-Skript erstellt wurde. Es ist hier aus Gründen der Nachvollziehbarkeit als reduzierter Pseudocode auf-

geführt und vollständig auf GitHub hinterlegt:

Listing 3.3: Testset Creation

```
INITIALIZE source_dir = 'path_to_source_files\\'
INITIALIZE target_dir = 'path_to_output_location\\'
INITIALIZE model_name = 'bert-base'
INITIALIZE max_input_length = 128
INITIALIZE label_amount = 745

FUNCTION create_active_subset(author_path, source_path, target_path, amount_of_authors):
    READ author list FROM author_path
    TRIM author list TO amount_of_authors
    READ dataset FROM source_path

    SET target_file = target_path + 'active_set.tsv'
    FOR EACH line IN dataset:
        SET temp_author = get_author(line)
        IF temp_author IN per_author:
            WRITE line TO target_file
    RETURN target_file

FUNCTION relabel(source_path, label_amount):
    READ comments FROM source_path
    EXTRACT authors FROM comments
    CREATE author_list WITH UNIQUE authors
    MAP author_list TO NUMERIC LABELS
    SET target_path = source_path WITH '_labeled.tsv' SUFFIX
    FOR EACH comment IN comments:
        REPLACE author WITH NUMERIC LABEL IN comment
        WRITE labeled_comment TO target_path
    RETURN target_path

FUNCTION main():
    CALL create_active_subset
    CALL force_maxlength
    CALL relabel
    CREATE full_dataset_frame FROM relabeled dataset
    SPLIT full_dataset_frame INTO training_set AND testing_set
    WRITE training_set TO CSV
    WRITE testing_set TO CSV WITH IDS
CALL main
```

Zunächst muss das Model spezifiziert werden, damit sichergestellt werden kann, dass die maximale Tokenzahl je Kommentar für den jeweiligen Tokenizer eingehalten wird [Z.3]. Dieser Schritt ist notwendig, da die im Rahmen dieser Arbeit verwendeten Models BERT

und BERTweet unterschiedliche Tokenizer einsetzen und somit derselbe Eingabetext in unterschiedlich viele Tokens untergliedert repräsentiert wird. Ein Überschreiten der Maximallänge bei nur einem einzigen Kommentar hätte bereits zur Folge, dass das Model den Datensatz nicht verarbeiten kann. Quell- und Zielverzeichnis werden ebenso definiert [Z.1-2], wie die maximale Eingabelänge (*max_input_length*) und die Anzahl der Label (*label_amount*) als Anzahl der Autoren [Z.4-5]. Für das NP-Set ist 745 die Anzahl aller Autoren mit mindestens 100 Kommentaren. In der Main-Methode [Z.30f] wird zunächst die Funktion *create_active_subset()* aufgerufen, in der für jeden Eintrag, der in der Vorverarbeitung erstellten Liste aktiver Autoren der Datensatz zeilenweise durchsucht und eine Datei erzeugt wird, die ausschließlich aus den Kommentaren besteht, die von diesen Autoren geschrieben wurden [Z.7f]. Danach wird mittels der Methode *force_maxlength()* sichergestellt, dass kein Kommentar inklusive der Special Tokens für das spezifizierte Model aus mehr als 128 Tokens besteht [Z.32]. Einträge, die trotz der entsprechenden Schritte in der Vorverarbeitung nach der Tokenization zu lang sind, beispielsweise weil sie aus Schriftzeichen bestehen, die das BERT-Model nicht kennt und daher jeweils mit dem [UNK] – Token übersetzt, werden für die weitere Verarbeitung verworfen, um das Training mit den übrigen Kommentaren zu ermöglichen. Anschließend wird die Funktion *relabel()* ausgeführt, in der die Pseudonyme aufsteigend iteriert bis zur gewünschten Anzahl durch das jeweilige Label ersetzt werden [Z.19f]. Es ist nicht möglich, einfach die Pseudonyme als Label zu nutzen. Diese sind zwar ebenfalls nur einmalig vergeben, die BERT-Implementierungen erwarten aber eine durchgängig aufsteigende Beschriftung der Eingabetexte. Der jetzt deutlich kleinere, aktive und mit Labeln versehene Datensatz wird nun als Dataframe eingelesen [Z.34]. Da sowohl Trainingsdaten zum Finetunen des Models, als auch Testdaten zu dessen Bewertung benötigt werden, wird der Datensatz, analog zum Vorgehen von Fabien et al. bei BertAA (siehe Kapitel 2.5.2), im Verhältnis 8:2 in Training-Set und Test-Set aufgeteilt [Z.35]. Somit wird gewährleistet, dass das Model die Kommentare, für die es die Autoren testweise bestimmen soll, nicht bereits im Training mit ihrem korrekten Label gelernt hat. Dies würde die Genauigkeit der Label-Prediction verfälschen. Zur Vorbereitung der Versuchsauswertung wird noch einmal eine Datei mit der Anzahl der Kommentare pro Autor in den jeweiligen Sets angelegt.

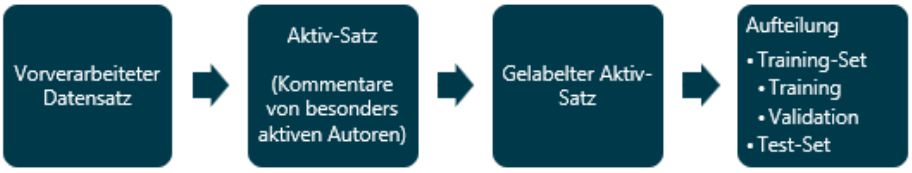


Abbildung 3.3: Datenaufbereitungsschritte für das Fine-Tuning

4 BERT Finetuning auf eigenen Datensätzen

In diesem Kapitel werden die folgenden Forschungsfragen beantwortet: Was muss beachtet werden, um ein BERT-Model auf NP- und Gen-Set finetunen zu können (Kapitel 3.3), wie lässt sich die Eignung des NP-Sets für die BERT-gestützte Authorship Attribution experimentell überprüfen (Kapitel 4.3), erzielt BERTweet bei gleichem Finetuning eine höhere Genauigkeit als BERT-base und enthalten auch sehr kurze Kommentare Informationen über den individuellen Schreibstil eines Autors, die in einer höheren Genauigkeit in der Classification Task bei der Einschluss in die Datensätze resultieren? Dazu werden die mit NP- und Gen-Set und den verschiedenen BERT-Models durchgeführten Versuche beschrieben. Nachdem zu Beginn die Versuchsumgebung dokumentiert ist (Kapitel 4.1), wird nachfolgend die Implementierung von BERT-base und BERTweet für das Finetuning erläutert. Abschließend werden durch die vier Versuchsdurchläufe geführt und deren Ergebnisse gelistet.

4.1 Versuchsumgebung

Alle Versuche wurden über eine Virtuelle Maschine auf Hardware des Instituts für IT-Sicherheit an der Universität zu Lübeck durchgeführt:

Systemhersteller & Model: VMware Inc. (7.1)

Betriebssystem & Typ: Microsoft Windows 11 Enterprise (x64)

CPU & RAM: AMD EPYC 7513 (64 GB)

GPU & GRAM: GRID A100-20C (20 GB)

Zu Beginn des Erstellungsprozesses dieser Arbeit wurde versucht, mit deutlich weniger Hardwareressourcen zurechtzukommen. Allerdings wurde schnell deutlich, dass die Beantwortung der formulierten Fragestellungen deutlich ressourcenintensivere Berechnungen erfordern würde, als mit der kleineren Hardwareumgebung durchführbar sein würde. Programmierung und Versuchsdurchführung fanden innerhalb von JetBrains PyCharm[12] (2024.1.1 Professional Edition) statt. Für das Finetuning wurde ein virtuelles Conda-Environment angelegt, um die Berechnungen auf der GPU zu ermöglichen.

[12] PyCharm Professional: `https://www.jetbrains.com/pycharm/`, (30.06.24)

4.2 Implementierung

Zur Erstellung der Jupyter Notebooks zum BERT-Finetuning haben wir uns an der Kaggle-Vorlage von Jaskaran Singh orientiert[13]. Für die Implementierung von BERTweet konnte dieselbe Vorlage mit einigen Anpassungen verwendet werden. Der vollständige, für das Finetuning eingesetzte Quellcode kann auf GitHub eingesehen werden.

4.2.1 BERT-base

Zunächst werden jeweils die Rahmenparameter für die Finetuning-Durchläufe definiert. Die Anzahl der Labels wird dem Model ebenso übergeben, wie die Anzahl der Trainings-Epochen. In den Versuchen wurde festgestellt, dass bis einschließlich in die sechste Epoche Steigerungen in der Genauigkeit erreicht werden. Ab der sechsten Epoche steigt dann aber ohne weiteren Genauigkeitsgewinn der *Validation-Loss*, es kommt also zum *Overfitting* des verwendeten Models. Entsprechend wurde die Anzahl der Epochen für alle Versuchsdurchläufe auf sechs festgelegt. Die BERT-Autoren empfehlen eine *Batch-Size* von 16 oder 32 [7]. Um das Finetuning möglichst performant durchzuführen, wurde sich für alle Versuche für die Batch-Size von 32 entschieden. In Versuchen mit dem BERT-base-Model wurde der *BertTokenizer* (*BertTokenizer .from_pretrained ('bert-base-uncased', do_lower_case = True)*) eingesetzt und der Tokenisierungs-Schritt mit den folgenden Parametern versehen: *add_special_tokens = True*, *max_length = max_len*, *pad_to_max_length = True*, *truncation = True*, *return_attention_mask = True*, *return_tensors = 'pt'*. Folglich werden in allen Kommentaren die in Kapitel 2.5.2 erklärten Special Tokens hinzufügt. Kommentare, die inklusive der Special Tokens weniger als die spezifizierten maximalen 128 Tokens umfassen, werden mit [PAD]-Tokens aufgefüllt, sodass sie für die Berechnung gleichförmig vom Model geladen werden können. In der übergebenen *attention_mask* werden die Positionen der [PAD]-Tokens markiert, sodass diese vom Attention Mechanism (siehe Kapitel 2.5.2) ignoriert werden und somit nicht in die Klassifizierung einfließen. Der Parameter *return_tensors='pt'* legt das Ausgabeformat als *Pytorch Tensor* fest. Anschließend wird der jeweilige Datensatz als *TensorDataset* eingelesen. Die Kommentartexte werden jetzt als Auflistung der IDs der enthaltenen Tokens repräsentiert. Verarbeitet werden sie neben den Labels auch mit ihren jeweiligen *attention_masks*. Für das Unsupervised Learning (siehe Kapitel 2.5.2) des verwendeten Models wird der Datensatz jetzt zufällig im Verhältnis 9:1 in Trainings- und Validierungs-Datensatz aufgeteilt.

[13] BERT Finetuning with Pytorch: `https://www.kaggle.com/code/jaskaransingh/bert-fine-tuning-with-pytorch` (16.06.24)

Listing 4.1: Tokenisierungsschritte am Beispiel

```
Original:
You could hear the host basically gasp once I said it hahaha.
Tokenized:
['you', 'could', 'hear', 'the', 'host', 'basically', 'gasp', 'once', 'i',
'said', 'it', 'ha', '##ha', '##ha', '.']
Token IDs:
[2017, 2071, 2963, 1996, 3677, 10468, 12008, 2320, 1045,
2056, 2009, 5292, 3270, 3270, 1012]
Token IDs padded:
tensor([ 2017, 2071, 2963, 1996, 3677, 10468, 12008, 2320, 1045,
         2056, 2009, 5292, 3270, 3270, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Als gepretraintes Model wurde *BertForSequenceClassification*[14] (*BertForSequenceClassification.from_pretrained ("bert-base-uncased", num_labels = number_of_labels, output_attentions = True, output_hidden_states = True)*) für die Durchläufe mit BERT-base eingesetzt. Entsprechend wird die Anzahl der zu betrachtenden Autoren (*number_of_labels*) übergeben. *Attention* und *Hidden States* repräsentieren aus den Kommentaren in den Finetuning-Schritten gewonnene Informationen innerhalb der Model-Verarbeitung und sind für die Analyse der Ergebnisse in Kapitel 5 relevant. Für das Finetuning wurden die jeweiligen Models aufgrund der größeren Kapazität zum parallelen Rechnen in die GPU geladen. Somit konnten die Versuche in deutlich kürzerer Zeit abgeschlossen werden, als es beim exklusiven Einsatz der CPU der Fall gewesen wäre. Jede der sechs Trainingsepochen dauerte für den großen Gesamtdatensatz über die 745 Autoren (vgl. Kapitel 3.2) jeweils etwa zehn Minuten, sodass ein Finetuning-Durchlauf nach etwa einer Stunde abgeschlossen war. Durchläufe auf kleineren Eingabe-Sets benötigten entsprechend weniger Zeit. Die Tokenization der Testdaten wird mit denselben Parametern konfiguriert, wie zuvor die Trainingsdaten. Die Classification Label bleiben ebenfalls gleich. Am Ende der Testphase werden die Predictions des gefinetuneten Models als Datei gespeichert. Diese werden wiederum zur automatischen Evaluation der Testergebnisse verwendet. Neben der Gesamtgenauigkeit (Anzahl der richtigen Vorhersagen geteilt durch Anzahl der Test-Kommentare)

[14] BERT für Sequence Classification: `https://huggingface.co/transformers/v3.0.2/model_doc/bert.html#bertforsequenceclassification`, (30.06.24)

wird noch die Genauigkeit für jedes Label ausgeben. Somit ist bekannt, für welche Labels die Authorship Attribution wie zuverlässig funktioniert hat. Zusätzlich zum Stil bestimmter Autoren kann durch die Betrachtung der Attention Weights je Layer die Verarbeitung der Kommentare im Model einzeln nachvollzogen werden. In dieser Arbeit wurde dies für einen Beispielkommentar mit dem für die Prediction ausschlaggebendsten Layer zur Untersuchung der Aufmerksamkeitsverteilung von TuBERT durchgeführt (siehe Kapitel 5.3.3). Für jeden Versuchsdurchlauf werden die Rahmendaten als Textdatei ausgeben, um die einzelnen Versuche besser vergleichen zu können. Diese Versuchsprotokolle (siehe Listing 4.2.1) beginnen mit allgemeinen Informationen über das Experiment, wie Model, Tokenizer und Umfang des Finetuning-Sets [Z.1-20]. Ebenfalls werden die jeweilige Finetuning-Statistik ausgeben [Z.21-30] und die Evaluationsergebnisse insgesamt und je Autor aufgelistet [Z.31 ff.].

Listing 4.2: Versuchsprotokoll mit Rahmendaten

```
1  === Experiment: bert-base_all\===
2  Used device: GRID A100-20C
3
4  Used tokenizer:
5  BertTokenizer.from_pretrained('bert-base-uncased', do_lower_case=True)
6  Max tokens: 128
7
8  Number of comments: 106,788
9  Classification labels: [0, ..., 745]
10 Number of training comments: 96109
11 Number of validation comments: 10679
12
13 Operating model:
14 BertForSequenceClassification.from_pretrained ("bert-base-uncased",
15 num_labels=number_of_labels, output_attentions=True,
16 output_hidden_states=True)
17
18 Batch size: 32
19 Epochs: 6
20
21 FINE-TUNING STATS:
22        Training Loss  Valid. Loss  Valid. Accur. Training Time Validation Time
23 epoch
24   1        4.95           4.07         0.26         0:10:23         0:00:23
25   2        3.58           3.50         0.35         0:10:23         0:00:22
26   3        2.81           3.25         0.39         0:10:23         0:00:22
27   4        2.23           3.18         0.40         0:10:23         0:00:22
28   5        1.80           3.17         0.42         0:10:22         0:00:22
29   6        1.52           3.19         0.42         0:10:23         0:00:22
```

```
30
31 EVALUATION:
32 Predictions:     Correctly classified:   Misclassified:   Accuracy:
33    26698                 11256                15442         0.4216
34
35 per Author:
36 label   total   correct   incorrect   accuracy
37   1       26        2          24         0.08
38   2       42        33         9          0.79
39   3       30        7          23         0.23
40   4       35        12         23         0.34
41   5       29        20         9          0.69
42                     ...
```

4.2.2 BERTweet

Die Versuchsimplementierung mit BERTweet (siehe Kapitel 2.5.2) wird analog zu der mit BERT-base vollzogen. Als gepretraintes Model wird *RobertaForSequenceClassification*[15] (*RobertaForSequenceClassification .from_pretrained ("vinai/bertweet-base", num_labels = number_of_labels, output_attentions = True, output_hidden_states = True)*), sowie der zugehörige Tokenizer *AutoTokenizer .from_pretrained('vinai/bertweet-base', do_lower_case = True)* geladen. Es wird genauso wie für BERT-base parametrisiert. Die Tokenization wird genauso wie für die BERT-base-Durchläufe, nur mit dem BERTweet-Tokenizer, durchgeführt. Allerdings wurde in Testläufen festgestellt, dass BERTweet noch über die sechste Trainingsepoche hinaus bis in die achte Epoche eine Verbesserung in der Genauigkeit erreicht. Daher wurden diese Versuche mit acht Epochen durchgeführt. Die Aufteilung der Datensätze in Trainings- und Testsets erfolgt analog zu den bisherigen Versuchen. Aufgrund der unterschiedlichen Tokenization der Kommentare, die ja bereits in die Datenverarbeitung eingeflossen ist, ist es zu kleinen Abweichungen in der Zusammensetzung der nun tatsächlich eingesetzten Sets gekommen. So enthält das BERT-Set ohne Kommentare mit Überlänge nur 133.486 Kommentare anstelle der 134.283 Kommentare im BERTweet-Set. Da sich die abweichenden Kommentare auf die 745 aktiven Autoren verteilen, lässt sich davon ausgehen, dass die Versuchsergebnisse davon nicht signifikant beeinflusst werden. Die Durchführung der Trainings- und Testläufe sowie die Ausgabe der Testergebnisse erfolgte ebenfalls analog zu den bisherigen Versuchen, sodass Leistung und Ergebnis aller durchgeführten Versuche vergleichbar sind.

[15] RoBERTa für Sequence Classification: `https://huggingface.co/docs/transformers/v4.42.0/en/model_doc/roberta#transformers.RobertaForSequenceClassification`, (30.06.24)

4.3 Versuchsdurchläufe

In dieser Sektion werden die verschiedenen Finetuning-Versuchsdurchläufe zur Ermittlung der besten Daten-Model-Konfiguration beschrieben und deren Ergebnisse abgebildet. Um Aussagen über die Erstellung eines bestmöglich geeigneten Datensatzes für die Authorship Attribution auf YouTube-Kommentaren treffen zu können, wurden die Versuchsdurchläufe so systematisiert, dass sich unsere Forschungsfragestellungen und einige Annahmen aus der Datensatzerstellung gezielt überprüfen lassen. In Versuchsdurchlauf A (siehe Kapitel 4.3.1) wird zunächst nachvollzogen, ob die Annahme richtig ist, dass es sich positiv auf die Genauigkeit der Authorship Attribution auf YouTube-Kommentaren mit BERT-Models auswirkt, wenn möglichst viele Kommentare in das Finetuning-Set einfließen, auch wenn dies dazu führt, dass die Kommentare je Autor stark unterschiedlich verteilt sind. Versuchsdurchlauf B (siehe Kapitel 4.3.2) wiederum diente zunächst der Feststellung der generellen Eignung des NP-Sets und des in dieser Arbeit angewandten Finetuning-Procederes für die Authorship Attribution auf YouTube-Kommentaren. Dazu wurden die BERT-Models in der Classification Task gegen ein n-gram Model antreten gelassen. Weiterhin wurde die Forschungsfrage untersucht, ob sich durch die Einbeziehung sehr kurzer Kommentare in das Finetuning-Set eine höhere Genauigkeit bei der Authorship Attribution erreichen lässt. Dazu wurden sowohl BERT-base, als auch BERTweet jeweils mit und ohne Kommentare mit weniger als 10 Wörtern gefinetuned. Die Konfiguration mit der höchsten Genauigkeit, BERTweet als Model ohne sehr kurze Kommentare im Finetuning-Set, wurde TuBERT genannt. Mit dieser wurden die weiteren Versuche durchgeführt. Im Versuchsdurchlauf C (siehe Kapitel 4.3.3) ging es darum, die Performance von TuBERT auf den beiden Datensätzen NP- und Gen-Set zu testen. Analog zum vorherigen Versuch wurden Messungen für Finetuning-Sets mit 745, 100 und 50 Autoren-Labels durchgeführt, um Aussagen über das Verhältnis von Genauigkeit und Anzahl zu unterscheidender Labels treffen zu können. Zusätzlich haben wir TuBERT noch auf den kombinierten Daten aus NP- und Gen-Set (*Com-Set*) gefinetuned, um zu untersuchen, ob sich ausgehend von einem größeren Datensatz und der dann größeren Anzahl an Kommentaren je Autor eine noch höhere Genauigkeit erreichen lässt.

4.3.1 Versuchsdurchlauf A

In Versuchsdurchlauf A ging es zunächst darum, die Entscheidung zu prüfen, nach der das Finetuning für einen möglichst großen Datensatz durchgeführt wird, wobei die Kommentare je Autor ungleich verteilt sind. Dazu wurden aus NP-Set zwei Subsets erstellt, das eine mit allen Kommentaren der 745 aktiven Autoren und das andere mit lediglich 100 Kommentaren für ebendiese Autoren. Die Genauigkeit in der Classification haben

wir sowohl für BERT-base, als auch für BERTweet bestimmt (siehe Tabelle 4.1).

Durchlauf	Kommentare	BERT-base	BERTweet
1	alle	42 %	43 %
2	100	30 %	30 %

Tabelle 4.1: Versuchsdurchlauf A: NP-Set, alle Kommentare je Autor vs. 100 Kommentare je Autor, BERT-base vs. BERTweet, Genauigkeit der Authorship Attribution in Prozent

4.3.2 Versuchsdurchlauf B

In Versuchsdurchlauf B ging es zunächst darum, die generelle Eignung von NP-Set und unserer Implementierung für die Classification Task festzustellen (siehe Tabelle 4.2). Dafür wurde als Vergleichswert die Classification Task ebenfalls mit einem n-gram Model für das NP-Set durchgeführt. Der vollständige Quellcode dazu ist auf GitHub hinterlegt. Im eingesetzten n-gram Model werden die Kommentare zunächst in Tokens aufgeteilt, um aus diesen die n-grams zu bestimmen. Anschließend wird mit dem *Naive Bayes Classifier*[16] die Wahrscheinlichkeit der verschiedenen Labels auf der Grundlage der n-gramme ermittelt. Dabei wurde der gerundete Mittelwert der n-gram Model Durchläufe mit n = [3,4,5] in der Tabelle als Vergleichswert hinzugefügt. Weiterhin wurde die Fragestellung überprüft, ob das auf, den YouTube-Kommentaren ähnlicheren Tweets gepretrainte BERTweet besser für unsere Classification Task geeignet ist, als BERT-base und somit eine höhere Genauigkeit erzielt. Dazu wurden beide Models nach gleichem Schema gefinetuned. In der Datensatzerstellung wurde angenommen, dass es von Vorteil für die Authorship Attribution ist, auch kürzeste Kommentare einzubeziehen, da diese die Anzahl von Kommentaren je Autor erhöhen und weitere Informationen über deren Schreibstil liefern könnten. Dies würde sich analog zum Versuchsdurchlauf A in einer höheren Genauigkeit bei der Authorship Attribution widerspiegeln (vgl. Tabelle 4.1). Um das zu überprüfen, wurden die Versuche sowohl mit den vollen Subsets inklusive kürzester Kommentare (+) als auch mit den Subsets durchgeführt, die nur Kommentare enthalten, die mindestens 10 Worte umfassen (-). Für alle Models und Subsets wurden die Versuche jeweils für 745, 100 und 50 Autoren durchgeführt, um Erkenntnisse über die Performance des jeweiligen Models im Verhältnis zur Anzahl von Autoren zu gewinnen. Für die vollen 745 Label konnte kein Vergleichswert mit dem n-gram Model berechnen werden, da die im Rahmen dieser Arbeit zur Verfügung stehenden Rechenkapazität für die dafür zu allokierenden 33 GB nicht

[16] Naive Bayes Classifier: `https://scikit-learn.org/stable/modules/naive_bayes.html`, (03.07.24)

ausreichen. Die Konfiguration mit der höchsten Genauigkeit, BERTweet als Model ohne sehr kurze Kommentare im Finetuning-Set, wurde TuBERT genannt.

D.	Labels	n-gram (-)	BERT-base (+)	BERT-base (-)	BERTweet (+)	BERTweet (-)
1	745	-	42 %	44 %	43 %	44 %
2	100	47 %	73 %	76 %	75 %	76 %
3	50	58 %	80 %	84 %	81 %	85 %

Tabelle 4.2: Versuchsdurchlauf B: NP-Set, Labels = Autoren, n-gram Model vs. BERT-base vs. BERTweet, mit (+) und ohne (-) Kommentare mit weniger als 10 Wörtern, Genauigkeiten der Authorship Attribution in Prozent

4.3.3 Versuchsdurchlauf C

Im Versuchsdurchlauf C ging es nun darum, die Performance von TuBERT auf den beiden Datensätzen (NP-Set und Gen-Set) zu testen. Analog zum vorherigen Versuchsdurchlauf wurden dabei Messungen für 100 und 50 Autoren für jeweils alle zu diesen Autoren gesammelten Kommentaren durchgeführt. Da in Gen-Set nur 211 Autoren mindestens 100 Kommentare verfasst haben, wird kein Durchlauf für 745 Label betrachtet. Um zu untersuchen, welche Auswirkung ein längerer Kommentarsammlungsprozess, resultierend in mehr Kommentaren je Autor gehabt hätte, haben wir abschließen noch TuBERT auf dem kombinierten *Com-Set* aus NP- und Gen-Set gefinetuned (siehe Tabelle 4.3). Dabei hat die Kombination beider Datensätze zu einer Erhöhung der Kommentare je Autor geführt, da für das Finetuning-Set die Autoren mit den absolut meisten Kommentaren ausgewählt wurden. Die Versuche werden in Kapitel 5 ausgewertet und NP- und Gen-Set weitergehend analysiert.

D.	Labels	TuBERT (NP)	TuBERT (Gen)	TuBERT (Com)
1	100	76 %	80 %	83 %
2	50	85 %	85 %	88 %

Tabelle 4.3: Versuchsdurchlauf C: TuBERT, Labels = Autoren, NP-Set vs. Gen-Set vs. Com-Set, Genauigkeiten der Authorship Attribution in Prozent

5 Versuchsauswertung und Datensatzanalyse

In diesem Kapitel wird zunächst noch einmal die Zielsetzung dieser Arbeit mit allen Forschungsfragen dargelegt (Kapitel 5.1). Es folgt die Auswertung der Ergebnisse der in Kapitel 4.3 beschriebenen Experimente. Danach folgt die Analyse der im Rahmen dieser Arbeit erstellten Datensätze NP- und Gen-Set. Nach einer kurzen Erläuterung der dazu angewandten Methoden (Kapitel 5.3.1), folgt zunächst der Vergleich der Datensätze, jeweils auf Ebene der Kommentartexte (Kapitel 5.3.2) und der Model Attention (Kapitel 5.3.3), um zu ergründen, inwiefern sich Kommentare aus NP-Set von denen aus dem Gen-Set unterscheiden. Abschließend wird das Kommentierverhalten der Autoren in den jeweiligen Sets wiederum zunächst auf Textebene (Kapitel 5.3.4) und dann auf Model Ebene (Kapitel 5.3.3) untersucht.

5.1 Zielsetzung

Ziel dieser Arbeit ist die Erstellung eines Datensatzes von YouTube-Kommentaren für das Finetuning eines BERT-Models zur Authorship Attribution. Dazu wurden die folgenden Forschungsfragen formuliert:

1. Wie gelingt es, einen Datensatz aus YouTube-Kommentaren für die Authorship Attribution zu erstellen?
2. Was muss ein Datensatz bestehend aus YouTube-Kommentaren erfüllen, damit er für die Authorship Attribution mit einem BERT-Model geeignet ist?
3. Was muss beachtet werden, um ein BERT-Model auf einem Datensatz zu Finetunen?
4. Wie lässt sich die Eignung eines Datensatzes für die Authorship Attribution mit einem BERT-Model experimentell überprüfen?

Um darüber hinaus einige im Zuge der Beantwortung dieser Fragen gemachten Annahmen zu überprüfen, wird in dieser Arbeit ebenfalls untersucht:

5. Erzielt das, auf Tweets gepretrainte BERTweet-Model eine höhere Genauigkeit in der Authorship Attribution auf YouTube-Kommentaren als das auf Fließtexten trainierte BERT-base-Model und lässt sich daraus eine stilistische Ähnlichkeit von Tweets und YouTube-Kommentaren schlussfolgern?

6. Enthalten auch sehr kurze Kommentare Informationen über den Schreibstil des jeweiligen Autors, durch die sich die Genauigkeit eines BERT-Models in der Authorship Attribution auf YouTube-Kommentaren erhöht?

7. Unterscheiden sich YouTube-Kommentare zu Videos aus dem News & Politics - Bereich stilistisch von Kommentaren zu allgemeinen Videos?

8. Liefert diese Arbeit, mit dem für die Authorship Attribution auf YouTube-Kommentaren gefinetuneten BERT-Model TuBERT, eine Grundlage für Detektion und Clustering orchestriert auftretender Kommentarautoren auf YouTube?

5.2 Versuchsauswertung

Zur besseren Übersichtlichkeit wurden die Werte in den Ergebnis-Tabellen (siehe Kapitel 4.3) auf ganze Prozente gerundet. Die Versuchsdurchläufe werden in alphabetischer Reihenfolge ausgewertet. Versuchsdurchlauf A diente der Bestimmung, ob es für die Authorship Attribution günstiger ist, alle gesammelten Kommentare zu nutzen oder nur eine gleiche Anzahl von Kommentaren je Autor in die Finetuning-Sets einzubeziehen. In Versuchsdurchlauf B wurde die grundsätzliche Eignung des angewendeten Verfahrens zum Finetuning der BERT-Models zur Authorship Attribution überprüft, bevor ermittelt wurde, ob BERT-base oder BERTweet besser für die Classification von YouTube-Kommentaren geeignet ist. Anschließend wurde untersucht, ob sich die zusätzliche Betrachtung von Kommentaren mit weniger als zehn Wörtern positiv auf die durch die Models erzielte Genauigkeit auswirkt. Die performanteste Konfiguration wurde TuBERT genannt. In Versuchsdurchlauf C wurde die Performance von TuBERT auf NP-Set, Gen-Set und dem aus beiden kombinierten *Com-Set* untersucht.

5.2.1 Versuchsdurchlauf A

Ziel von Versuchsdurchlauf A (Ergebnisse in Tabelle 4.1) war es, die Entscheidung zu überprüfen, nach der alle im NP-Set enthaltenen Kommentare der Autoren mit mindestens 100 Kommentaren (*aktive Autoren*) in das Finetuning-Set eingeschlossen werden, anstatt gleich viele Kommentare je Autor einzubeziehen. Dazu wurden für BERT-base (siehe Kapitel 2.5.2) und BERTweet (siehe Kapitel 2.5.2) jeweils zwei Finetuning-Durchläufe mit anschließender Evaluation der erzielten Genauigkeit für die Authorship Attribution Task durchgeführt. Im ersten Durchlauf wurde das Finetuning auf einem NP-Subset aus allen im NP-Set enthaltenen Kommentaren der 745 aktiven Autoren durchgeführt. Für den zweiten Durchlauf wurde die Anzahl der Kommentare für jeden der 745 Autoren auf 100 reduziert. Die erzielten Genauigkeiten für die Durchläufe mit BERT-base sind in Spalte

3, für die mit BERTweet in Spalte 4 dargestellt (v.l.n.r.). Dabei erzielten beide Models mit 42 Prozent zu 43 Prozent im ersten und jeweils rund 30 Prozent im zweiten Durchlauf eine sehr ähnliche Genauigkeit, wobei BERTweet jeweils eine um bis zu 1 Prozent höhere Genauigkeit erreichte. Insgesamt ist die erzielte Genauigkeit relativ niedrig, was auf die Schwere der Classification Task für eine so große Label-Zahl zurückzuführen ist. Die deutlich höhere Genauigkeit im ersten Durchlauf gegenüber der im zweiten Durchlauf bestätigt die Entscheidung, möglichst viele Kommentare je Autor in das Finetuning für die Authorship Attribution mit aufzunehmen, auch wenn dies zu einer Unausgewogenheit der Kommentarzahl je Autor führt.
Es ist anzunehmen, dass unter den Autoren, die überproportional häufig kommentiert haben, auch solche sind, die besonders gut durch die Models wiedererkannt werden. Um diese Annahme zu überprüfen, wurden aus den für jeden Finetuning-Durchlauf protokollierten Rahmendaten (vgl. Listing 4.2.1) die Evaluationsergebnisse für die zehn Autoren mit den meisten Kommentaren im Test-Set betrachtet. Diese machen mit den von ihnen vorhandenen 2845 Kommentaren gegenüber den insgesamt 26698 Kommentaren im Test-Set gut 11 Prozent aller Kommentare aus. Da die Aufteilung in Training- und Test-Set (siehe Kapitel 3.3) zufällig vorgenommen wird, ist anzunehmen, dass sich dieses Verhältnis so auch annähernd im gesamten NP-Set für die 745 aktiven Autoren widerspiegelt. Von den zehn Autoren mit den meisten Kommentaren sind durchschnittlich jeweils 285 Kommentare in das Test-Set eingeflossen. Somit haben diese Autoren gegenüber den durchschnittlich 36 Kommentaren je Autor im gesamten Test-Set überproportional häufig kommentiert und wurden mit rund 74 Prozent Genauigkeit auch deutlich überdurchschnittlich präzise wiedererkannt.

5.2.2 Versuchsdurchlauf B

Ziel von Versuchsdurchlauf B (Ergebnisse in Tabelle 4.2) war es zunächst, die grundsätzliche Eignung des in dieser Arbeit angewandten Verfahrens zum Finetuning der BERT-Models zur Authorship Attribution auf NP-Set zu überprüfen (a). Weiterhin wurde untersucht, ob BERTweet für die Authorship Attribution von YouTube-Kommentaren besser geeignet ist, als BERT-base (b). Außerdem wurde die Annahme überprüft, nach der auch sehr kurze Kommentare zusätzliche Informationen über den Schreibstil eines Autors beinhalten, durch die sich die Genauigkeit eines BERT-Models in der Authorship Attribution erhöht (c). Es wurden Versuche mit drei verschiedenen Models durchgeführt. Zur Prüfung der grundsätzlichen Eignung (a) wurde die Authorship Attribution auf NP-Set mit einem n-gram Model durchgeführt. Die Ergebnisse der Classification Task für das n-gram Model sind in Spalte 3, die für BERT-base in den Spalten 4 und 5 sowie für BERTweet in den Spalten 6 und 7 wiedergegeben (v.l.n.r). Dabei zeigt das (+) in der Spaltenbezeich-

nung an, dass das Model auch mit Kommentaren unter einer Länge von zehn Wörtern im Set gefinetuned wurde. Das (-) wiederum zeigt an, dass entsprechend keine sehr kurzen Kommentare im Finetuning-Set einbezogen wurden. Um ebenfalls einen Eindruck von der Classification Performance in Abhängigkeit der Anzahl der zu unterscheidenden Label zu bekommen, wurden jeweils drei Finetuning-Durchläufe durchgeführt. Im ersten Durchlauf wurde die Authorship Attribution für alle 745 Autoren vollzogen, wobei hier keine Authorship Attribution mit dem n-gram Model durchgeführt werden konnte, da die im Rahmen dieser Arbeit eingesetzten Hardwareressourcen nicht über genügend Rechenleistung dafür verfügten. Es folgte ein Durchlauf für Finetuning-Set und Classification für 100 Autoren und ein weiterer Durchlauf für 50 Autoren. Grundsätzlich war zu erwarten, dass die Genauigkeit mit kleiner werdender Autorenzahl zunimmt, da die Classification Task für weniger Label leichter wird [8]. Wenn die Implementierung des BERT-Finetuning-Procederes (siehe Kapitel 4.2) im Rahmen dieser Arbeit korrekt war, ist aufgrund der stärkeren Transformer-Architektur eine deutlich höhere Genauigkeit der Authorship Attribution bei den BERT-Models im Vergleich zum n-gram Model zu erwarten (siehe Kapitel 2.5.2 und 2.5.3).
Das n-gram Model erreichte für 100 Autoren eine Genauigkeit von 47 Prozent, und für 50 Autoren von 58 Prozent in der Authorship Attribution Task. BERT-base mit kurzen Kommentaren erzielte für 745 Autoren eine Genauigkeit von 42 Prozent, für 100 Autoren von 73 Prozent und für 50 Autoren von 80 Prozent. BERT-base ohne kurze Kommentare erreichte in den Durchläufen 44, 76 und 84 Prozent. Mit BERTweet sind es 43, 75 und 91 Prozent mit und 44, 76 und 85 Prozent Genauigkeit ohne kurze Kommentare.

a) Ist das im Rahmen dieser Arbeit eingesetzte Finetuning-Procedure mit BERT-Models grundsätzlich für die Authorship Attribution von YouTube-Kommentaren geeignet?

   Ja. Mit allen BERT-Model-Varianten wird eine um mindestens 26 Prozentpunkte höhere Genauigkeit für 100 Autoren und eine um mindestens 22 Prozentpunkte höhere Genauigkeit für 50 Autoren, verglichen mit dem n-gram Model, erreicht.

b) Ist BERTweet für die Authorship Attribution von YouTube-Kommentaren besser geeignet, als BERT-base?

   Ja. In allen Finetuning-Durchläufen wird mit BERTweet eine mindestens so hohe Genauigkeit erreicht, wie mit BERT-base. Allerdings beträgt die höchste Differenz lediglich 2 Prozentpunkte im Durchgang für 100 Autoren inklusive kurzer Kommentare. BERTweet scheint mit den sehr kurzen Kommentaren geringfügig besser umgehen zu können, als BERT-base. Dies könnte auf die unterschiedlichen Pretraining-Datensätze zurückzuführen sein, nach denen BERT-base auf Fließtexten und BERTweet auf Tweets trainiert wurde. Der Genauigkeitsgewinn von den Durchläufen mit

BERTweet ist aber insgesamt zu niedrig, um dies sicher annehmen zu können, da auch aufgrund der verschiedenen Pretraining-Procedures mit einer besseren Performance von BERTweet zu rechnen ist (siehe Kapitel 2.5.2).

c) Führt die zusätzliche Einbeziehung sehr kurzer Kommentare zu einer höheren Genauigkeit bei der Authorship Attribution mit BERT-Models?

Nein. In allen Durchläufen ohne Kommentare mit weniger als zehn Wörtern Länge wurde eine höhere Genauigkeit von bis zu vier Prozentpunkten erreicht, als in denen mit sehr kurzen Kommentaren. Weiterhin scheint BERTweet im Verhältnis etwas besser mit sehr kurzen Kommentaren zurechtzukommen. Sehr kurze Kommentare enthalten zu wenige Tokens, um deren Embeddings im globalen Kontext gewichten zu können, sodass diese für auf der Transformer-Architektur basierende LLMs keine zusätzlichen, für die Authorship Attribution nützlichen Informationen liefern können (siehe Kapitel 2.5.2).

Die Model-Datensatz-Konfiguration, mit der die höchste Performance erreicht wurde, das ohne sehr kurze Kommentare gefinetunete BERTweet, wird im Weiteren mit *TuBERT* bezeichnet

5.2.3 Versuchsdurchlauf C

Ziel von Versuchsdurchlauf C (Ergebnisse in Tabelle 4.3) war es, die Performance von TuBERT bei der Authorship Attribution auf NP-Set (Spalte 3) mit der auf Gen-Set (Spalte 4) zu vergleichen. Anders als in Versuchsdurchlauf B wurden nur zwei Läufe, einer für 100 und einer für 50 Autoren durchgeführt, da im Gen-Set nur 211 Autoren jeweils mehr als 100 Kommentare verfasst haben. Zusätzlich wurde TuBERT noch auf den kombinierten Daten von NP- und Gen-Set, dem *Com-Set* getestet (Spalte 5). Auf NP-Set erzielte TuBERT eine Genauigkeit von 76 Prozent für 100 und eine Genauigkeit von 85 Prozent für 50 Autoren. Auf Gen-Set erreichte TuBERT für 100 Autoren sogar eine Genauigkeit von 80 Prozent, während diese für 50 Autoren ebenfalls bei 85 Prozent lag. TuBERT lag auf Com-SET mit 83 und 88 Prozent jeweils 3 Prozentpunkte in der Genauigkeit vor TuBERT auf Gen-Set. Da für die Finetuning-Sets jeweils die Anzahl von Autoren mit den je meisten Kommentaren gewählt werden, ist es nicht überraschend, dass auf Com-Set noch einmal eine höhere Genauigkeit erzielt wird. Um zu untersuchen, warum TuBERT für 100 Autoren auf Gen-Set immerhin 4 Prozentpunkte vor dem Finetuning-Durchlauf für 100 Autoren auf NP-Set lag, wird erneut die autorenspezifische Evaluation in den protokollierten Rahmendaten (siehe Kapitel 4.2) betrachten. Analog zur Auswertung von Versuchsdurchlauf A wurden jeweils die zehn Autoren genauer betrachtet, die am häufigsten kommentiert

haben. Dabei haben deren Kommentare in NP-Set einen Anteil von 30 Prozent an allen Test-Kommentaren und in Gen-Set von 36 Prozent. Durchschnittlich sind von jedem dieser Autoren in NP-Set 227 und in Gen-Set 256 Kommentare gesammelt worden. Auch wenn diese Autoren mit lediglich 85 Prozent im Gen-Set Genauigkeit gegenüber den 84 Prozent im NP-Set korrekt klassifiziert wurden, begründet sich doch im deutlich höheren Anteil dieser Autoren im Gen-Set die höhere Genauigkeit des Durchlaufs für 100 Autoren. Dass TuBERT auf NP-Set für 50 Autoren wiederum gleichauf ist, liegt an der größeren Varianz der Genauigkeiten je Autor im Gen-Set. Diese ist mit 0,074 gegenüber der 0,064 im Durchlauf auf NP-Set größer ausgeprägt, wodurch sich schwerer für alle Kommentare eines Autors gemeinsame Eigenschaften finden lassen.

5.3 Datensatzanalyse

Um die Ergebnisse der im Rahmen dieser Arbeit durchgeführten Versuche (siehe Kapitel 4.3 und 5.2) besser einordnen zu können, werden NP- und Gen-Set im folgenden Kapitel genauer untersucht. Dabei wird sowohl anhand der Kommentartexte, als auch anhand von TuBERT untersucht, wie sie Kommentare aus dem News & Politics Bereich im NP-Set von allgemeinen Kommentaren im Gen-Set unterscheiden. Zunächst beginnt dieses Kapitel mit einer Sektion über die für diese Untersuchung genutzten Methoden (Kapitel 5.3.1). Es folgt der Vergleich der Datensätze anhand aller Kommentartexte in Kapitel 5.3.2 und der Attention Weights (siehe Kapitel 2.5.2) anhand eines Beispielkommentars in Kapitel 5.3.3. Abschließend werden die Kommentare der Autoren, die mit mindestens 90 Prozent Genauigkeit besonders gut wiedererkannt wurden, händisch analysiert und in Untergruppen eingeteilt (Kapitel 5.3.4). Um nachzuvollziehen, ob sich diese händische Gruppierung auch durch TuBERT automatisiert durchführen ließe, wurden für die Kommentare zweier unterschiedlich eingestufter Autoren die Token-Embeddings visualisiert (Kapitel 5.3.5). In diesem Kapitel werden also die Forschungsfragen untersucht, inwiefern sich YouTube-Kommentare aus NP-Set von den allgemeinen Kommentaren in Gen-Set unterscheiden und ob TuBERT nutzbar für die weitere Detektion und das Clustering orchestriert auftretender Kommentarautoren auf YouTube ist.

5.3.1 Methodik

In dieser Sektionen werden die zur Analyse der Datensätze eingesetzten Methodiken erläutert. Dafür wird zunächst das Verfahren zur statistischen Kommentaranalye erklärt. Anschließen wird die Berechnung der für die Classification ausschlaggebendsten Layer-Head-Kombination dargestellt, bevor die *Principal Component Analysis* (PCA) definiert wird. Die Berechnung der Token-Embeddings lasst sich in Kapitel 2.5.2 nachvollziehen.

Verfahren zur statistischen Analyse der Kommentartexte

Zur statistischen Analyse der Kommentartexte wurde in dieser Arbeit ein Python-Skript eingesetzt, das alle Kommentare aus dem zuvor spezifizierten Datensatz ausliest. Anschließend werden die durchschnittliche Kommentarlänge und deren Varianz, die Häufigkeitsverteilung der auftretenden Wörter und Satzzeichen und die Häufigkeitsverteilung der auftretenden Emoticons, jeweils für den gesamten Datensatz und je Autor, ermittelt. Abschließend werden die Anzahl von Satzzeichen und Emoticons im Verhältnis zur Anzahl der Zeichen beziehungsweise Wörter in den Kommentaren bestimmt, wieder jeweils für den gesamten Datensatz und je Autor. In einem weiteren Python-Skript werden die Unterschiede zwischen zwei bereits wie eben beschrieben analysierten Datensätzen, im Falle dieser Arbeit NP- und Gen-Set, untersucht. Dafür wird zunächst die Differenz der durchschnittlichen Kommentarlängen, deren Varianz sowie das Verhältnis von Satzzeichen zu Zeichen und Emoticons zu Wörtern in den Kommentaren ermittelt. Für die in beiden Sets enthaltenen Wörter werden diejenigen bestimmt, die nicht im jeweils anderen Datensatz vorkommen. Abschließend werden für jeden Datensatz die 1.000 häufigsten Wörter ermittelt, die nicht unter den 1.000 häufigsten Wörtern des jeweils anderen Datensatzes enthalten sind. Der dafür verwendete Quellcode ist auf GitHub nachvollziehbar.

Bestimmung der für die Classification ausschlaggebenden Layer-Head-Kombination

Für die Bestimmung der für die Classification ausschlaggebenden Layer-Head-Kombination wir der höchste Wert der *Importance Scores* (IS) für alle Layer und Heads bestimmt: [12][29]:

$$IS_{l,h} = \Sigma_{ij} |G_{l,h}[i,j] \cdot A_{l,h}[i,j]|$$

Wobei l der jeweilige Layer, h der jeweilige Head und $G_{l,h}$ die Gradientenmatrix der entsprechenden Attention Weights. Weiterhin ist $A_{l,h}$ die Attention Weights - Matrix, i und j die Indizes in der jeweiligen Matrix. Der entsprechende Quellcode ist vollständig auf GitHub hinterlegt.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA)[17], einen Algorithmus, der unter Erhalt relevanter Eigenschaften die Dimension hin zu einer erleichterten grafischen Darstellbarkeit reduziert.

[17] Principal Component Analysis (PCA): `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html` (23.06.24)

5.3.2 Datensätze im Vergleich: Textebene

Um die unterschiedliche Performance von NP-Set und Gen-Set zu untersuchen und dabei Erkenntnisse über mögliche Unterschiede im Kommentierverhalten für den News & Politics Bereich gegenüber allgemeinen Kommentaren zu gewinnen, wurden die Kommentartexte in den Datensätzen statistisch ausgewertet. Dazu wurde zunächst die durchschnittliche Kommentarlänge ermittelt, wofür der Tokenizer des *Natural Language Toolkits*[18], einem der am häufigsten eingesetzten Python-Toolkits zur Verarbeitung von Datensätzen natürlicher Sprache, verwendet wurde. Betrachtet wurden die auch für das TuBERT-Finetuning eingesetzten Datensatzversionen ohne die Kommentare mit weniger als 10 Wörtern Länge. Weiterhin wurden die Varianz der Kommentarlänge errechnet und das Verhältnis von verwendeten Interpunktionszeichen und Emoticons zur Anzahl an Wörtern ermittelt. Damit orientiert sich diese Arbeit an der Untersuchung schreibstilbasierter Eigenschaften nach Sari et al. [24]. Die Ergebnisse sind in Tabelle 5.1 dargestellt. Inspiriert von der *Comment Term Frequency Comparison* (CTFC), einer von Thelwall et al. [31] vorgestellten Methode zur Analyse des Kommentierverhaltens bei bestimmten Themen, wurde weiterhin die Häufigkeitsverteilung der in den Kommentaren verwendeten Wörter untersucht. Mit einer durchschnittlichen Länge von rund 138 zu 125 Word-Tokens sind die Kommentare aus NP-Set im Schnitt 13 Wort-Tokens länger als die aus Gen-Set. Die Varianz der Kommentarlängen ist mit 0.65 im Vergleich zu 0.67 im Gen-Set ein wenig stärker ausgeprägt. Gleichzeitig ist der Anteil verwendeter Interpunktion mit 0.0273 im NP-Set gegenüber den 0.025 im Gen-Set etwas höher, das Verhältnis verwendeter Emoticons mit 0.0016 gegenüber 0.0009 sogar deutlich höher.

Um eine Aussage über das, je Datensatz verwendete Vokabular treffen zu können, wurde

Dataset	Kommentare	d.Länge	Varianz	Punktuation	Emoticons
N-P	25751	138	0,65	0,027	0,0016
Gen	24177	125	0,67	0,025	0,0009

Tabelle 5.1: Kommentartexte von NP-Set und Gen-Set im Vergleich, durchschnittliche Länge als einfacher Durchschnitt, Punktuations- und Emoticon-Wert im Verhältnis zur Anzahl der Wörter

die Häufigkeitsverteilung der verwendeten Worte untersucht. Dabei haben NP- und Gen-Set einige Überschneidungen. So sind *the*, *to*, *and*, *a*, *you*, *of*, *that*, *it* und *is* jeweils unter den zehn häufigsten Worten, wie es für Datensätze aus Kommentaren in englischer Sprache nicht anders zu erwarten ist. Vermutlich ist dies ein wesentlicher Grund, warum das auf Fließtexten gepretrainte BERT im Vergleich zu BERTweet weiterhin zuverlässig funktioniert. Bei der Betrachtung der häufigsten Vokabeln, die nicht unter den 1.000 häufigsten

[18] Natural Language Toolkit Tokenizer: `https://www.nltk.org/api/nltk.tokenize.html` (15.06.24)

Worten im jeweils anderen Datensatz enthalten sind, werden aber Unterschiede deutlich. Während in den Kommentaren aus NP-Set häufig bewertende Adjektive (zum Beispiel *major*, *incorrect*, *obviously*, *definitely*) vorkommen, wie sie in Diskussionen zu erwarten sind, findet im Gen-Set häufig Umgangssprache (zum Beispiel *damn*, *gosh*) Verwendung. Ebenfalls finden sich im NP-Set politisierte Begriffe, wie *race*, *corruption*, *trump* oder *socialism*, welche im Gen-Set nicht vorkommen.
Weiterhin wurde für die vergleichende Emoticon-Analyse zunächst die Anzahl der verwendeten Emoticons für jeden der beiden Datensätze ausgegeben und die verwendeten Emoticons in die vier Klassen *engagement*, *positive*, *negative* und *neutral / unkown* eingeteilt (siehe Tabelle 5.2). Dabei fasst die Klasse *engagement* diejenigen Emoticons zusammen, die zu einer bestimmten Nutzerinteraktion motivieren sollen (beispielsweise *:backhand_index_pointing_up:* oder *:envelope_with_arrow:*). Während diese mit 45 Prozent den größten Anteil an die Emoticons im NP-Set haben (+), kommen sie im Gen-Set nicht vor. Insgesamt wurden 25 Prozent der Emoticons im NP-Set und 68 Prozent im Gen-Set der positiven Klasse zugeordnet. Im NP-Set sind 15 Prozent negativ und im Gen-Set 19. Werden die *engagement*-Emoticons aus der Betrachtung ausgeschlossen (-), so erhalten wir einen Anteil von 46 Prozent *positive* und 27 Prozent *negative* im NP-Set. Mit 46 Prozent *positive* im NP-Set gegenüber den 68 Prozent im Gen-Set fällt die durch Emoticons ausgedrückte Stimmung im Gen-Set im Schnitt deutlich positiver aus, als im NP-Set. Durch

Dataset	Emoticons	Verhältnis	Engagement	Positive	Negative	Neutral
NP+	7193	0,0016	45 %	25 %	15 %	15 %
NP-	3955	-	0 %	46 %	27 %	27 %
Gen	2774	0,0009	0 %	68 %	19 %	13 %

Tabelle 5.2: Vergleichende Emoticon-Analyse für NP- und Gen-Set (+ = inklusive *engagement*-Emoticons, - = exklusive), Absolute Anzahl von Emoticons im Set, das Verhältnis zur Anzahl von Wörtern im Set, Anteil der *engagement*-Emoticons, *positive*-Emoticons, *negative*-Emoticons und *neutral*-Emoticons

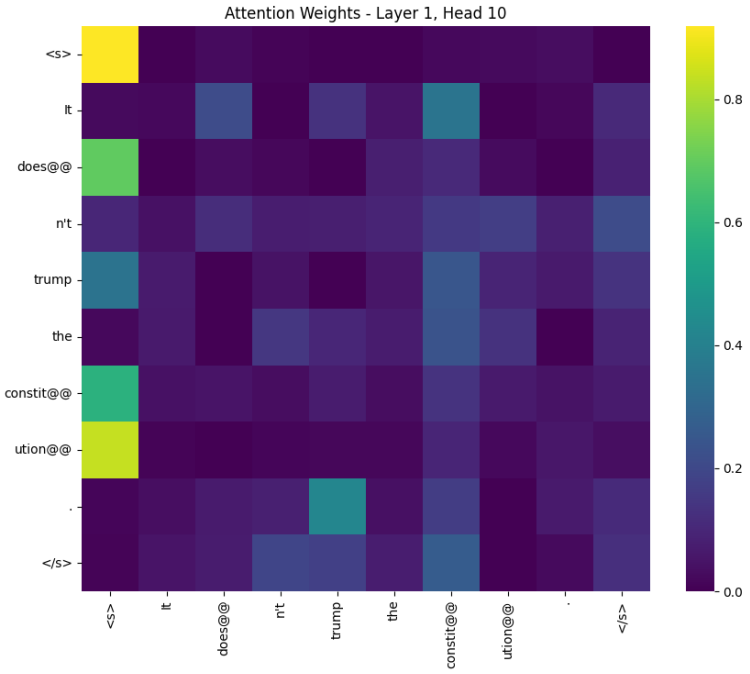
die Rückverfolgung der *engagement*-Emoticons im NP-Set konnte festgestellt werden, dass diese von nur drei verschiedenen Autoren, dafür aber in großer Zahl je Kommentar eingesetzt werden. Aufgrund des repetitiven Kommentierverhaltens dieser drei Autoren und dem offensichtlichen Ziel, weitere Nutzer zu einer bestimmten Interaktion zu bewegen, ist anzunehmen, dass es sich dabei um wenigstens teilautomatisierte Accounts handelt. Da diese auch mit 100 Prozent Genauigkeit in der TuBERT Authorship Attribution wiedererkannt werden, wurden diese zur Autorenklasse der *Sales-Bots* zusammengefasst (vgl. Kapitel 5.3.4). Da diese Autoren in ihrem Kommentierverhalten und dem inflationären Einsatz von Emoticons, die nur durch sie verwendet werden, das Verhältnis der statistischen Betrachtung der eingesetzten Emoticons nach deren ausgedrückter Stimmung ver-

zerren, wurden die NP-Emoticons auch ohne die *engagement*-Emoticons betrachtet (NP- in Tabelle 5.2). Dass der Anteil positive Emotionen ausdrückender Emoticons auch nach dem Ausschluss der *engagement*-Emoticons mit 68 Prozent im Gen-Set gegenüber den 46 Prozent im NP-Set mit noch 22 Prozentpunkten Differenz im Gen-Set so viel höher ist, lässt vermuten, dass sich die Grundstimmung auf YouTube-Kanälen je nach deren Genre unterscheidet. So scheint die generelle Stimmung im Schnitt positiver, als auf Kanälen aus dem News & Politics - Bereich.

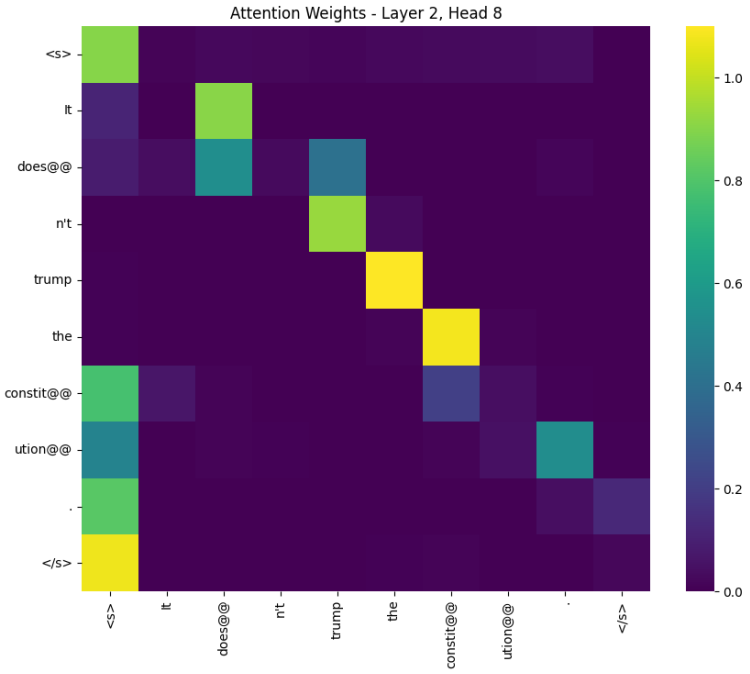
5.3.3 Datensätze im Vergleich: Model-Ebene

Um die Entscheidung von TuBERT für ein bestimmtes Label in der Classification Task genauer zu betrachten, werden in diesem Abschnitt die Attention Weights des Models (siehe Kapitel 5.3.1) anhand eines Beispielkommentars sowohl für das auf NP-Set, als auch auf Gen-Set gefinetunete TuBERT visualisiert. Dazu wurde der Kommentar *'It doesn't trump the constitution.'* aus NP-Set als Beispiel gewählt, da dieser nach der Tokenisierung aus genau 10 Tokens besteht, was die Übersichtlichkeit der Abbildung begünstigt und darüber hinaus mit '*trump*' und '*constitution*' zwei Worte enthalten sind, von denen aus der statistischen Analyse der Kommentartexte in Kapitel 5.3.2 bekannt ist, dass sie im Gen-Set nicht vorkommen. Es ist also anzunehmen, dass die Attention Weights, je nachdem ob mit NP- oder Gen-Set gefinetunet wurde, sehr unterschiedlich ausfallen. Das Ergebnis der Visualisierung ist in Abbildung 5.1 dargestellt, wobei Grafik (a) die Attention Weights für TuBERT auf NP-Set und Grafik (b) die Attention Weights für TuBERT auf Gen-Set für den Beispielkommentar abbildet. X- und y-Achse sind jeweils mit den Kommentar-Tokens belegt, wobei jeweils zu Beginn und am Ende des Kommentars die Special Tokens (siehe Kapitel 2.5.2) mit '<*s*>' wiedergegeben sind. Die Farben geben jeweils das Attention Weight für jeden Token wieder, wobei die Gewichtung höher ausfällt, je heller die Farbe in der Darstellung ist. Die entsprechende Farblegende findet sich rechter Hand der jeweiligen Grafik. Für die Abbildung wurde die jeweils für die Classification ausschlaggebende Layer-Head-Kombination gewählt (siehe Kapitel 5.3.1).

(a) In Grafik (a) werden die Attention Weights für den Beispielkommentar für TuBERTs Layer 1, Head 10 auf NP-Set dargestellt. Zu sehen ist ein klarer Aufmerksamkeitsschwerpunkt auf dem [CLS]-Token zum Beginn des Kommentars. Dies ist nicht verwunderlich, da für die Sequence Classification der Aufmerksamkeitsschwerpunkt auf dem [CLS]-Token und dessen Beziehungen zu den anderen Tokens liegt (siehe Abbildung 2.2). Weiterhin ersichtlich ist eine starke Gewichtung bei *constitution*, *trump* und *doesn't*, während *it*, *the* und *'.'* kaum betrachtet werden. Offenbar hat TuBERT gelernt, dass diese im englischen Sprachgebrauch in hoher Zahl vorkom-



(a) TuBERT auf NP-Set



(b) TuBERT auf Gen-Set

Abbildung 5.1: Visualisierung der TuBERT Attention Weights am Beispielkommentar *'It doesn't trump the constitution.'*, X- und y-Achse = Tokens, Special-Tokens als '<s>', Färbung nach Gewichtung: je heller, desto stärker

menden, und daher von vielen Autoren verwendeten Worte sich nicht dazu eignen, zwischen verschiedenen Autoren zu unterscheiden. Vielmehr scheinen Subjekt, Prädikat und Objekt im Fokus der Aufmerksamkeit zu liegen, zumal die Begriffe *trump* und *constitution* nur von bestimmten Autoren verwendet wurden. Aus der, im Rahmen der statistischen Textanalyse in Kapitel 5.3.2 erstellten Häufigkeitsverteilung der verwendeten Wörter ist bekannt, dass *constitution* seltener vorkommt und somit vermutlich von weniger Autoren verwendet wurde, als *trump*. Es ist anzunehmen, dass die Gewichtung von *constitution* deswegen höher ausfällt.

(b) In Grafik (b) werden die Attention Weights für den Beispielkommentar für TuBERTs Layer 2, Head 8 auf Gen-Set dargestellt. Neben der im Vergleich zur Grafik (a) ebenfalls starken Gewichtung für den [CLS]-Token liegt der Aufmerksamkeitsschwerpunkt hier gleichermaßen bei *trump* und *constitution*. Aus der statistischen Textanalyse in Kapitel 5.3.2 ist bekannt, dass beide Wörter im Gen-Set nicht vorkommen, weshalb TuBERT seine Aufmerksamkeit auf diese, für ihn besonderen Tokens konzentriert.

Anders als bei anderen Arbeiten, brauchen für die Datensatz-Erstellung für das BERT-Model-Finetuning die Stoppwörter nicht entfernt werden, da diese aufgrund des Attention Mechanism für die Classification keine Rolle spielen.

5.3.4 Autoren im Vergleich: Textebene

Ziel dieses Abschnittes ist die händische Einteilung einiger Autoren anhand ihrer Kommentartexte in NP-Set, um in Kapitel 5.3.5 untersuchen zu können, ob TuBERT auf Grundlage der Kommentar-Token-Embeddings für eine vergleichbare, automatisierte Einteilung nutzbar wäre. Betrachtet wurden alle Kommentare von allen Autoren, die bereits in der Authorship Attribution Task für 745 Autoren von BERT-base auf NP-Set mit mindestens 90 Prozent Genauigkeit oder weniger als 11 Prozent Genauigkeit wiedererkannt wurden. Von den 745 Autoren waren 44 mit über 90 Prozent und 230 mit unter 11 Prozent Genauigkeit klassifiziert. Es wurde angenommen, dass die mit sehr hoher Genauigkeit wiedererkannten Autoren leicht unterscheidbar sind und sie sich entsprechend gut einteilen lassen. Weiterhin wurden die Gründe untersucht, weshalb die Kommentare aller betrachteten Autoren besonders gut oder schlecht klassifiziert wurden.
In der händischen Analyse wurde vor allem das allgemeine Schriftbild betrachtet, also ob unter anderem die englische Grammatik weitestgehend korrekt angewendet und ob in sehr langen oder eher kurzen Sätzen kommentiert wurde. Darüber hinaus wurde die Einteilung auf Grundlage des durch die Kommentare beim Forschenden vermittelten Sentiments sowie den in den Kommentaren aufgegriffenen Themen vorgenommen. Eingeteilt

wurden die mit über 90 Prozent Genauigkeit wiedererkannten Autoren in die Gruppen *Sales-Bots*, *Community Manager*, *Missionaries*, *Trolls*, *Political Senders* und *Comfort Seekers*. Dabei zeichnen sich die Sales-Bots durch stark repetitive Kommentare und den ständigen Einsatz von Emoticons, die zu einer bestimmten Nutzerinteraktion oder -käufen anregen sollen, aus (vgl. Kapitel 5.3.2). Als Community Manager wurden diejenigen Autoren klassifiziert, die durch bestimmte Floskeln wie *'Vielen Dank für Ihren Kommentar!'* mutmaßlich um moderative Nutzerbindung bemüht waren. Die als Missionaries eingeteilten Autoren wiederum nutzen häufig Bibelzitate und deren Übertragung auf aktuelle Themen. Als Trolls wurden diejenigen Autoren bezeichnet, deren Kommentare eine als stark negativ empfundene Tonalität aufwiesen, häufig in Verbindung mit Diffamierungen bestimmter Personengruppen, wie beispielsweise Frauen oder Polizeibeamten. Political Senders wiederum zeichnen sich durch ein stark politisiertes Ausdrucksverhalten zugunsten eines bestimmten politischen Narrativs aus. Einige Autoren engagieren sich beispielsweise gegen den russischen Präsidenten beziehungsweise zugunsten der Ukraine oder gegen Rassismus. Als Comfort Seekers wurden die Accounts klassifiziert, die breit über eine persönliche Leidensgeschichte berichtet haben und dabei einen auffallenden Schreibstil verwendeten.
Die Autoren, die mit weniger als 11 Prozent Genauigkeit wiedererkannt wurden, wurden ebenfalls händisch in die Gruppen *Wild*, *Super Short* und *Extremely Variant* untergliedert. Die Autoren aus der Wild-Gruppe haben nicht in zusammenhängenden Sätzen kommentiert, häufig unter scheinbar willkürlichem Einsatz diverser Sonderzeichen. Als Super Short wurden die Autoren klassifiziert, deren Kommentare durchgängig auffallen kurz waren. Die Autoren der Extremely Variant - Gruppe sind mit Abstand am häufigsten unter den schlecht wiedererkannten Autoren vertreten. Ihre Kommentare charakterisiert ein in Kommentarlänge und Vokabular stark variierender Schreibstil, häufig entsprungen im Austausch mit weiteren Autoren.
Zwei Accounts sind besonders aufgefallen, da sich diese zwar händisch der Gruppe der Sales-Bots zuordnen lassen, diese aber zu 0 Prozent wiedererkannt wurden. Ihre Kommentare wurden ohne Ausnahme anderen Accounts aus der Sales-Bot Gruppe zugeschrieben. Es ist anzunehmen, dass diese beiden Accounts quasi-identisch zu den anderen Sales-Bots kommentiert haben, diese sich aber aufgrund der deutlich höheren Anzahl an im Datensatz enthaltenen Kommentaren in der Classification durchgesetzt haben.

5.3.5 Autoren im Vergleich: Model-Ebene

Nachdem in Kapitel 5.3.3 tiefergehend betrachtet wurde, wie die Attention Weights und damit die Classification in den im Rahmen dieser Arbeit gefinetuneten Models wirkt, wird nun die Unterscheidbarkeit einzelner Autoren anhand ihrer Token-Embeddings (siehe Ka-

pitel 2.5.2) im für die Classification entschiedenen letzten Model Layer betrachtet. Damit wird Forschungsfrage 8 untersucht, da, im Falle der guten Unterscheidbarkeit von Autoren aufgrund der geclusterten Token-Embeddings ihrer Kommentare, ein Mechanismus gefunden wurde, mit dem sich TuBERT für Detektion und Clustering orchestriert auftretender Kommentarautoren auf YouTube nutzen lässt. Dazu wird die PCA (siehe Kapitel 5.3.1) über den Token-Embeddings der Kommentare zweier Autoren aus NP-Set visualisiert, die im Kapitel 5.3.4 händisch in zwei unterschiedliche Gruppen eingeordnet wurden. Der dazu verwendete Quellcode ist auf GitHub hinterlegt. Beide gewählten Autoren gehörten zu denen, die mit über 90 Prozent Genauigkeit in der Authorship Attribution Task wiedererkannt wurden. Der erste Autor wurde händisch der Missionary-Gruppe und der zweite Autor der Trolls-Gruppe zugeordnet. Abbildung 5.2 zeigt ein Streudiagramm vom PCA der [CLS]-Token Embeddings. X- und y-Achse geben die nach PCA-Reduzierung verbliebenen Dimensionen an. Jeder Punkt entspricht einem Kommentar und die Farbe dem Label, den das Model für den Kommentar bestimmt hat. Eine entsprechende Farblegende findet sich rechter Hand der Abbildung. Die Häufung der Kommentare in den entgegengesetzten Ecken zeigt, dass der für TuBERT verwendete Tokenisierungsmechanismus grundsätzlich dazu in der Lage ist, die zuvor im Rahmen dieser Arbeit vorgenommene händische Einteilung in Autorengruppen automatisiert vorzunehmen. Wie gut das Clustering anhand der Token-Embeddings allerdings für eine größere Anzahl an Autoren und Kommentaren funktioniert, konnte nicht untersucht werden. Die Versuche dazu scheiterten an der, im Rahmen dieser Arbeit zur Verfügung stehenden, für diese Berechnungsschritte aber zu geringen Rechenleistung. Eine weitergehende Untersuchung empfiehlt sich aber für zukünftige Arbeiten.

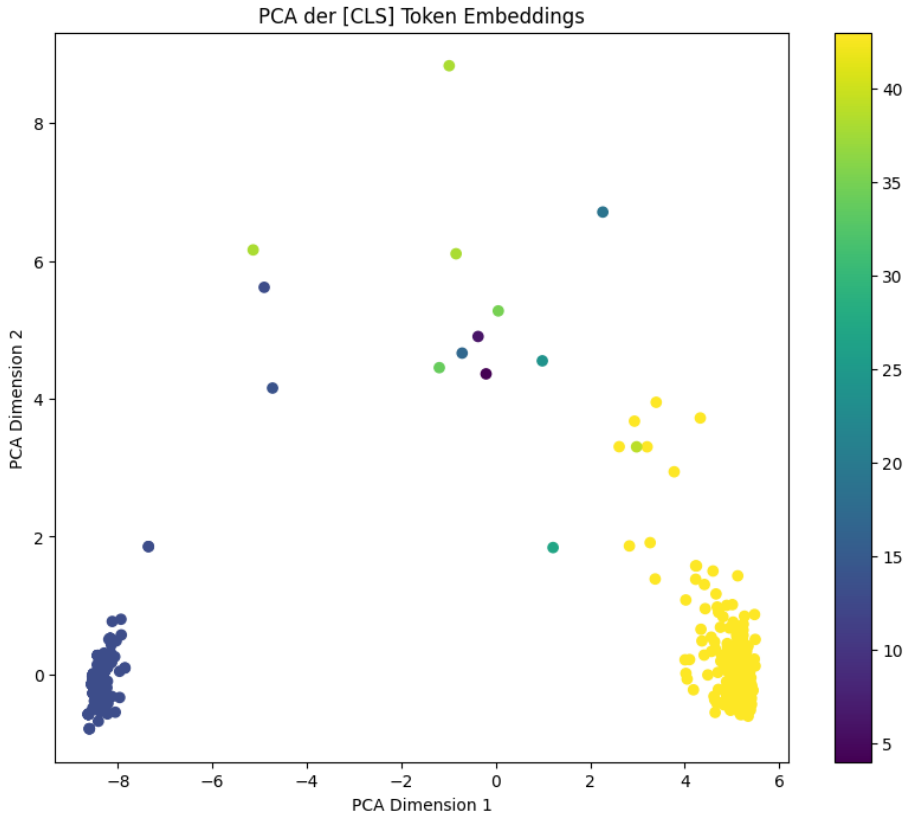


Abbildung 5.2: Visualisierung der PCA der [CLS]-Token Embeddings für die Kommentare zweier Autoren, die händisch zwei unterschiedlichen Autorengruppen zugeordnet wurden. X- und y-Achse sind durch die nach PCA verbliebenen beiden Dimensionen belegt. Jeder Datenpunkt stellt einen Kommentar dar, dessen Farbe das durch TuBERT erkannte Label.

6 Diskussion

In diesem Kapitel werden im Rahmen dieser Arbeit gemachte Annahmen und erzielte Ergebnisse diskutiert. Analog zum Aufbau der Arbeit werden dabei zunächst die Kommentarsammlung und Datensatzgestaltung betrachtet, bevor die Verarbeitung der Kommentare diskutiert wird. Abschließend wird noch die Eignung von TuBERT zur Detektion koordiniert auftretender Kommentarautoren und Social Bots behandelt. Gleich zu Beginn steht die Einordnung dieser Forschungsleistung als sicherheitsrelevante Forschung nach Maßgabe des Gemeinsamen Ausschuss zum Umgang mit Sicherheitsrelevanter Forschung[19].

Ethische Durchführbarkeit

In der Arbeit werden nur von den Personen selbst öffentlich gemachte Kommentare erhoben. Diese Kommentare werden allerdings so miteinander verknüpft, dass Personen anhand ihres Schreibstils mit Gruppen gegliedert werden. Diese Implementierung haben wir öffentlich verfügbar gemacht, aber die Modelle und die Daten sind nicht öffentlich verfügbar. Es könnten nun die Bedenken aufkommen, dass diese Art der Eingruppierung für diskriminierende Zwecke missbraucht werden können. Wir denken allerdings, dass die vorliegende Arbeit dieser Art des Missbrauchs keine unmittelbare Beihilfe schafft.
Nach Maßgabe des Gemeinsamen Ausschuss zum Umgang mit Sicherheitsrelevanter Forschung umfasst sicherheitsrelevante Forschung wissenschaftliche Arbeiten, bei denen die Möglichkeit besteht, dass sie Wissen, Produkte oder Technologien hervorbringen, die von Dritten missbraucht werden können, um Menschenwürde, Leben, Gesundheit, Freiheit, Eigentum, Umwelt oder ein friedliches Zusammenleben zu schädigen. Darunter fällt häufig auch Forschung an generativer Künstlicher Intelligenz, welche auch Gegenstand dieser Arbeit ist. Anhand der Leitfragen für die ethische Bewertung sicherheitsrelevanter Forschung[20] wurde auch das Thema dieser Arbeit diskutiert.

Der Einsatz von stylometriebasierter Authorship Attribution durch ein LLM in Sozialen Medien ließe sich neben der Untersuchung von Desinformationskampagnen theoretisch

[19] Gemeinsamen Ausschuss zum Umgang mit Sicherheitsrelevanter Forschung: https://www.sicherheitsrelevante-forschung.org/

[20] Leitfragen für die ethische Bewertung sicherheitsrelevanter Forschung: https://www.sicherheitsrelevante-forschung.org/wp-content/uploads/2024/08/2024_Informationsbroschuere_Sicherheitsrelevante_Forschung.pdf

auch dafür einsetzen, bestimmte politische Meinungen und deren Multiplikatoren gezielt zu unterdrücken oder zu verfolgen, was eine missbräuchliche Nutzung im obigen Sinne darstellen würde. Da die in dieser Arbeit beschriebenen Schritte nur sehr grundsätzlich die Eignung des gewählten Models für die Ausgangsfragestellung skizzieren und der Schwerpunkt auf der Erstellung von Datensätzen liegt, welche, wie auch die finegetuneten Models, nicht veröffentlicht werden, kann diese Arbeit nicht unmittelbar missbraucht werden.

Aufgrund der fehlenden Unmittelbarkeit hinsichtlich möglicher Missbrauchsszenarien lässt sich folgern, dass es sich bei dieser Arbeit nicht um besorgniserregende sicherheitsrelevante Forschung im Sinne des oben genannten Ausschusses handelt. Die Arbeit wurde ohne Kooperationspartner erstellt, weswegen diesbezüglich keine zusätzlichen Risiken bestehen. Durch die getroffen Maßnahmen zum Schutz persönlicher Informationen (siehe Kapitel 3.1.4) konnte der Einklang mit geltenden Datenschutzregeln gewährleistet werden.

Kommentarsammlung und Datensatzgestaltung

Ein essenzieller Teil dieser Arbeit behandelt die Entwicklung eines Verfahrens zur Erstellung von für die Authorship Attribution auf YouTube-Kommentaren geeigneten Datensätzen und deren Evaluation.
Da kein für die Authorship Attribution von YouTube-Kommentaren geeigneter Datensatz öffentlich zugänglich ist, mussten im Rahmen dieser Arbeit die Datensätze selbst erstellt werden. Diese Aufgabe ist umfangreich, da es einerseits eine große Menge an Kommentardaten auf YouTube gibt, der Zugriff auf diese durch die YouTube-API aber für kostenfreie Nutzer stark begrenzt ist (siehe Kapitel 2.5.1). Da ein Wettrüsten zwischen LLM-gestützten Methoden zur Entwicklung immer ausgefeilterer Social Bots und denen zur Entwicklung wirksamer Gegenmaßnahmen besteht, ist die fortlaufende Aktualisierung der Trainingsdaten und der angewendeten Erkennungstools weiterhin erforderlich [35].

Für zukünftige Arbeiten sind vergleichende Versuchsdurchläufe auf einer umfangreicheren Kommentar-Sammlung mit einem besseren Verhältnis von Kommentaren je Autor interessant.
Im Zuge dieser Arbeit hat sich herausgestellt, dass es, im Gegensatz zur Twitter-Domäne, schwierig ist, eine für das Finetuning notwendige, größere Anzahl an Kommentaren je Autor zu erhalten. Anders als bei Twitter können nicht einfach alle Postings eines Nutzers am Stück extrahiert werden. Stattdessen müssen die Kommentare über die Videos abge-

fragt werden, zu denen sie erstellt wurden. Für die weitere Forschung wäre es sicher interessant, durch eine umfangreichere Kommentar-Sammlung ein besseres Verhältnis von Kommentaren je Autor zu erhalten. Dies sollte die Genauigkeit der Autor-Klassifizierung verbessern. Aufgrund des begrenzten zeitlichen Umfangs dieser Abschlussarbeit konnte die Kommentarsammlung aber nicht über einen noch längeren Zeitraum betrieben werden (siehe Kapitel 3.1).

Die separate Betrachtung einzelner Social-Media-Plattformen ist notwendig.
Methoden zur Analyse von Social-Media-Texten sind für eine allgemeine, medienübergreifende Perspektive häufig nur schwer zu evaluieren, da die Unterschiede zwischen den einzelnen Medien groß sind. Dies kann eine separate Betrachtung mehrerer einzelner Medien notwendig machen, für die wiederum verschiedene Ansätze notwendig sein können [31]. Das befürwortet den in dieser Arbeit gewählten Ansatz, nur YouTube als Plattform alleinstehend zu betrachten und Kommentare aus dem News & Politics - Bereich zu sammeln.

Für das Erkennen und Clustern koordiniert auftretender YouTube-Accounts sind sowohl stilistische als auch inhaltliche Eigenschaften der jeweiligen Kommentare relevant.
Sari et al. [24] haben untersucht, wie die stilistischen und thematischen Eigenschaften der in den jeweiligen Datensätzen enthaltenen Texte die bei der Authorship Attribution erzielten Genauigkeiten beeinflussen. Unter anderem ging aus dieser Untersuchung hervor, dass themenbasierte Eigenschaften, wie die Verwendung bestimmter politisierter Begriffe, in genau den Datensätzen zur höheren Genauigkeit beitragen, deren Texte eine hohe thematische Vielfalt abbilden. Sollte also die Detektion orchestriert kommentierende Accounts zu bestimmten politischen Narrativen erfolgen, wie um eine mutmaßliche Desinformationskampagne zu einem bestimmten Thema zu analysieren, dürften die dieses Narrativ repräsentierenden Wortkombinationen genau dann in der Model-Attention stärker ausschlagen, wenn auch der untersuchte Datensatz eine große thematische Breite umfasst. Dieser Vermutung nachzugehen, wäre für eine zukünftige Arbeit interessant.
Im Rahmen dieser Abschlussarbeit wurden nur Kanäle einbezogen, die dem News & Politics - Bereich zugeordnet sind. Dass diese trotzdem ein breites inhaltliches Spektrum abbilden (vgl. Kapitel 3.1), erklärt, warum politisierte Begriffe weiterhin in der Betrachtung der TuBERT Attention Weights so stark gewichtet werden (siehe Kapitel 5.3.3).

Kommentare aus dem News & Politics - Bereich lassen sich nicht besonders gut erkennen.
Die Vermutung, dass sich Kommentare aus dem News & Politics - Bereich ihren Autoren besser zuschreiben lassen, als die allgemeinen Kommentare aus dem Gen-Set, wurde

nicht bestätigt (siehe Versuchsdurchlauf C in Kapitel 4.3). Beim Durchlauf für die Authorship Attribution mit 100 verschiedenen Autoren erzielte TuBERT auf dem Gen-Set sogar eine um 4 Prozentpunkte höhere Genauigkeit. Die Gründe dafür liegen aber bei Kommentierstil und überproportional vielen Kommentaren einiger bestimmter Autoren.

Es ist nicht ausgeschlossen, dass die erstellten Datensätze aus bereits bereinigten Kommentaren bestehen.
Ein möglicher Schwachpunkt des in dieser Arbeit verfolgten Ansatzes zur Sammlung der YouTube-Kommentare über die YouTube-API, ist das damit vorgeschossene Vertrauen in YouTube, die zur Verfügung gestellten Daten nicht zu verändern. Ein mögliches Motiv dazu könnte in der Wahrung der konzerneigenen Geschäftsinteressen liegen. Ferner liegen keine Erkenntnisse darüber vor, ob und inwieweit die Kommentare zu den Videos der im Rahmen dieser Arbeit betrachteten YouTube-Kanäle bereits durch YouTube automatisch gefiltert oder durch das kanalspezifische Community Management entfernt wurden. Es ist also nicht ausgeschlossen, dass die erstellten Datensätze aus bereits bereinigten Kommentaren bestehen.

Kommentardaten-Verarbeitung

Kommentare mit weniger als zehn Wörtern Länge verschlechtern die in der Authorship Attribution erzielte Genauigkeit.
Zu Beginn dieser Arbeit wurde angenommen, dass auch Kommentare aus weniger als 10 Wort-Tokens zusätzliche Informationen zum Schreibstil eines Autors beisteuern und Datensätze für das BERT-Model-Finetuning, welche zusätzlich auch diese sehr kurzen Kommentare enthalten, bei der Authorship Attribution eine höhere Genauigkeit erzielen. Dabei wurde die Grenze von 10 Tokens nach dem Vorbild von BertAA gewählt [8]. Die in Versuchsdurchlauf B ermittelten Genauigkeiten (siehe Kapitel 4.3) haben aber gezeigt, dass sowohl BERT-base als auch BERTweet in allen Durchläufen ohne die zusätzlichen kurzen Kommentare eine höhere Genauigkeit erzielen. Somit wurde diese Annahme widerlegt.

Lassen sich durch behutsame Normalisierung mehr stilistische Informationen aus den originalen Kommentartexten erhalten?
Es wäre interessant zu untersuchen, ob durch *behutsame Normalisierung* (siehe Kapitel 3.1) mehr stilistische Informationen aus den originalen Kommentardaten erhalten geblieben sind, welche es TuBERT ermöglichen, in der Classification Task eine höhere Genauigkeit zu erzielen. Dazu könnten in einer zukünftigen Arbeit, die bereits gesammelten rohen Kommentardaten mit dem in dieser Arbeit verwendeten Python-Skript zur Datensatzer-

stellung (siehe Kapitel 3.2), wie hier beschrieben, verarbeitet werden. Nur der kleine Programmteil zur Text-Normalisierung müsste abgeändert werden. Anschließend wäre die Authorship Attribution mit dem auf dem jeweiligen Datensatz gefinetuneten TuBERT durchzuführen und die erzielte Genauigkeit zu vergleichen. Im Falle einer höheren Genauigkeit von TuBERT auf dem behutsam normalisierten Finetuning-Set würde die in dieser Arbeit gemachte Entscheidung zum Einsatz ebendieser behutsamen Normalisierung, bestätigt.

Die Eignung von TuBERT zur Detektion koordiniert auftretender Kommentarautoren und Social Bots

Im Rahmen dieser Arbeit konnte gezeigt werden, dass TuBERT für die Authorship Attribution ähnliche Informationen verwendet, wie bei der händischen Einteilung.
Häufig weisen Methoden zur Erkennung koordinierter Accounts Schwächen auf, sobald es sich bei diesen um von Menschen betriebene Accounts handelt, die, anstelle von plumpen Falschbehauptungen und einem automatisierten Verhalten, Diffamierung und Polarisierung einsetzen, um Meinungen zu beeinflussen [26]. Durch die im Rahmen dieser Arbeit durchgeführten Untersuchungen zu TuBERTs Aufmerksamkeitsverteilung (Attention Weights) in Kapitel 5.3.3 und dem Clustering der für dessen Autoren-Klassifizierung ausschlaggebenden Token-Embeddings in Kapitel 5.3.5 konnte gezeigt werden, dass TuBERT für die Authorship Attribution ähnliche Informationen verwendet, wie bei der händischen Einteilung.

Tweets und YouTube-Kommentare ähneln einander kaum.
Da Tweets und YouTube-Kommentare beides durch Nutzer veröffentlichte, kürzere Texte in der Social-Media-Domäne sind, wurde angenommen, dass das auf Tweets gepretrainte BERTweet bei der Authorship Attribution von YouTube-Kommentaren eine deutlich höhere Genauigkeit erreicht, als das auf Fließtexten gepretrainte BERT-base. Tatsächlich wurde in den Finetuning-Versuchsdurchläufen mit BERTweet nur eine geringe Verbesserung der Genauigkeit gegenüber den Durchläufen mit BERT-base erzielt (siehe Kapitel 4.3). Neben den für das Pretraining verwendeten Datensätzen unterscheiden sich beide Models auch im Pretraining-Procedure, was ebenfalls eine höhere Genauigkeit von BERTweet gegenüber BERT-base bei der Classification von YouTube-Kommentaren erwarten lässt (siehe Kapitel 2.5.2). Daher kann auch die geringe Verbesserung nicht allein der vermuteten Ähnlichkeit zwischen Tweets und YouTube-Kommentaren zugeschrieben werden. Nguyen et al. [19] haben aber gezeigt, dass das auf Tweets gepretrainte BERTweet gegenüber weiteren, auf dem gleichen Datensatz gefinetuneten BERT-Models, für verschiedene NLP-Tasks auf Tweets eine höhere Genauigkeit erreicht. Daher ist anzuneh-

men, dass auch ein auf YouTube-Kommentaren gepretraintes BERT-Model für verschiedene NLP-Tasks auf YouTube-Kommentaren eine höhere Genauigkeit erzielen würde. Für das Pretraining eines LLMs werden aber deutlich größere Datensätze, als im Rahmen dieser Arbeit für das Finetuning erstellt wurden und deutlich größere Rechenkapazitäten (vgl. Kapitel 4.1), benötigt. Daher bleibt die Überprüfung dieser Vermutung Gegenstand zukünftiger Arbeiten.

Lässt sich durch den Einsatz von *Contrastive Learning* die Fähigkeit von TuBERT zu Einteilung verschiedener Autoren weiter verbessern?
In dieser Arbeit nicht betrachtet wurde der Ansatz von Ai et al. [1], Contrastive Learning für die Authorship Attribution mit BERT-Models einzusetzen. In der entsprechenden Veröffentlichung wurde gezeigt, dass die mit diesem Ansatz vorgestellte Methode *Contra-X* in der Lage ist, auch zwischen sehr ähnlich schreibenden Autoren zu unterscheiden. Es ist also anzunehmen, dass die Erweiterung der in dieser Abschlussarbeit skizzierten Methode zur Token-Embedding-basierten Einteilung von YouTube-Kommentarautoren (siehe Kapitel 5.3.5) durch die Erweiterung des Finetuning-Procederes um Elemente des Contrastive Learnings noch einmal verfeinert werden kann. Die Überprüfung dieser Annahme obliegt ebenfalls zukünftigen Arbeiten.

Die Token-Embedding-basierte Einteilung für eine größere Autorenzahl zu untersuchen, ist für zukünftige Arbeiten interessant.
Die im Rahmen dieser Arbeit zur Verfügung stehenden, begrenzten Rechenkapazitäten waren dazu jedoch nicht ausreichend (vgl. Kapitel 4.1).

Es ist anzunehmen, dass die Performance von TuBERT mit der von BertAA (siehe Kapitel 2.5.2) zu vergleichen ist. Fabien et al. [8] haben ebenfalls die BERT-Model-basierte Authorship Attribution für eine größere Zahl von Autoren untersucht. Vergleichbar mit den Ergebnissen aus Versuchsdurchlauf C in Kapitel 4.3 sind dabei die Messungen auf dem aus E-Mails bestehenden *Enron*-Datensatz. Dieser besteht aus im 130.000 E-Mails mit einer durchschnittlichen Länge von 150 Tokens. Genau wie in Versuchsdurchlauf C wurden keine E-Mails mitbetrachtet, die nicht wenigstens 10 Tokens lang sind. Insgesamt bleiben die mit TuBERT bei der Authorship Attribution erzielen Genauigkeit mit bis zu 83 Prozent für 100 und bis zu 88 Prozent für 50 Autoren (siehe Tabelle 4.3) deutlich unterhalb der von Fabien et al. ermittelten 97 Prozent für 100 und 98,1 Prozent für 50 Autoren. Dies liegt vermutlich auch an der höheren Anzahl an Kommentaren je Autor (3685 für das Set aus 50 Autoren gegenüber den 530 für Com-Set) sowie der durchschnittlich höheren Textlänge (183 Tokens gegenüber den 132 für Com-Set).

7 Zusammenfassung

Die Sorge vor einer Manipulation der öffentlichen Meinung durch Social-Media-gestützte Desinformationskampagnen, macht das Erkennen koordinierter Nutzeraktivitäten und insbesondere die Detektion von Social Bots zu einem wichtigen Forschungsthema [2] [21] [25]. Dabei ist die Videoplattform YouTube trotz ihrer weiten Verbreitung im Gegensatz zur Microblogging-Plattform X (ehemals Twitter) in der Forschungsliteratur noch deutlich unterrepräsentiert [31]. Mit dieser Arbeit wird durch das Erstellen eines für die Authorship Attribution geeigneten Datensatzes aus YouTube-Kommentaren und dem Finetuning des, auf der BERT-Architektur [7] basierenden, Large Language Models (LLM) TuBERT ein grundlegender Beitrag zur Detektion orchestriert auftretender Nutzeraccounts in YouTubes Kommentarspalten gelegt.
Dafür wurde das Sammeln der Kommentare mit dem Social-Media-Analysetool Mozdeh [30] über die YouTube-API durchgeführt und ein eigenes Verfahren für die Verarbeitung der rohen Kommentardaten hin zu, für das Finetuning von BERT-Models geeigneten Datensätzen entwickelt (Kapitel 3.1 und 3.3). Es wurde angenommen, dass koordinierte Aktivitäten zur Verbreitung von Desinformation am ehesten in den Kommentarspalten von Videos auftreten, die sich mit dem aktuellen politischen Geschehen befassen. Daher wurden für den im Rahmen dieser Arbeit erstellten Hauptdatensatz (NP-Set) YouTube-Kanäle aus dem News & Politics - Bereich betrachtet. Für den Vergleichsdatensatz (Gen-Set) wurden solche Kanäle ausgewählt, die nicht in diesen Bereich fallen, in denen aber im Betrachtungszeitraum am meisten kommentiert wurde. Im Zuge der im November 2024 anstehenden US-Präsidentschaftswahl wurde die Auswahl auf US-amerikanische Kanäle begrenzt.
Um die Eignung der erstellten Datensätze NP- und Gen-Set zu überprüfen, wurden mehrere Versuchsdurchläufe für das Finetuning der Models BERT-base [7] und BERTweet [19] auf unterschiedlichen Rahmenbedingungen durchgeführt (Kapitel 4.3). Dazu wurde die Aufgabe der Authorship Attribution als Sequence Classification Task betrachtet, bei der jeder Autor durch ein eigenes Label repräsentiert wird. Durch den Vergleich der in diesem Task erzielten Genauigkeiten konnte die performanteste Konfiguration aus Model und NP-Subset ermittelt werden. Die Model-NP-Set-Konfiguration, für die die beste Performance erzielt wurde, wurde TuBERT genannt. Nach derzeitigem Kenntnisstand ist TuBERT das erste BERT-Model, dass für die Authorship Attribution auf YouTube-Kommentaren trainiert wurde.

Um die Eignung von TuBERT für die automatische Erkennung koordiniert kommentierender Nutzeraccounts und Social Bot - Detection zu untersuchen, wurden die in NP- und Gen-Set enthaltenen Kommentartexte zunächst statistisch analysiert (Kapitel 5.3.2). Weiterhin wurden solche Kommentarautoren, die mit mindestens 90 Prozent Genauigkeit in der Authorship Attribution Task wiedererkannt wurden, auf der Grundlage ihres stilistischen und inhaltlichen Kommentierverhaltens, händisch in Gruppen eingeteilt (Kapitel 5.3.4). In der Visualisierung von TuBERTs Aufmerksamkeitsverteilung (Attention Weights) konnte gezeigt werden, dass das Model für die Classification mit dem Schwerpunkt auf bestimmten politisierten Begriffen wie *'trump'* oder *'constitution'* eine ähnliche Grundlage für die Klassifizierung nutzt, wie es für die händischen Einteilung der Fall war (Kapitel 5.3.3). Darüber hinaus wurde am Beispiel von zwei händisch in unterschiedliche Gruppen eingeteilten Autoren gezeigt, dass sich die von TuBERT für die Kommentar-Klassifizierung verwendeten Token-Embeddings nutzen lassen, um YouTube-Kommentare visuell zu clustern (Kapitel 5.3.5).
Damit liefert diese Arbeit eine Grundlage für den automatisierten Einsatz BERT-basierter LLMs für die Erkennung koordinierter Kommentatoren und Social Bots in der YouTube-Domäne. Die nächsten Schritte, wie die Betrachtung größerer Datensätze oder die Anwendung des Token-Embedding-basierten Clusterings, bleiben zukünftigen Arbeiten überlassen.

Literaturverzeichnis

[1] Bo Ai, Yuchen Wang, Yugin Tan, and Samson Tan. Whodunit? learning to contrast for authorship attribution. *arXiv preprint arXiv:2209.11887*, 2022.

[2] Sinan Aral and Dean Eckles. Protecting elections from social media manipulation. *Science*, 365(6456):858–861, 2019.

[3] Jane Arthurs, Sophia Drakopoulou, and Alessandro Gandini. Researching youtube. *Convergence*, 24(1):3–15, 2018.

[4] Mohan Sai Dinesh Boddapati, Madhavi Sai Chatradi, Sridevi Bonthu, and Abhinav Dayal. Youtube comment analysis using lexicon based techniques. In *International Conference on Cognitive Computing and Cyber Physical Systems*, pages 76–85. Springer, 2022.

[5] Monica Anderson Brooke Auxier. Social media use in 2021. *Pew Research Center*, 2021.

[6] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3), mar 2024.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[8] Maël Fabien, Esau Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. BertAA : BERT fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*. NLP Association of India (NLPAI), 2020.

[9] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Communications of the ACM*, 59(7):96–104, 2016.

[10] Frederik Ferreau. Desinformation als herausforderung für die medienregulierung. *AfP*, 52(3):204–210, 2021.

[11] Zoubin Ghahramani. *Unsupervised Learning*, pages 72–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[12] Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi. Improving deep learning interpretability by saliency guided training. *Advances in Neural Information Processing Systems*, 34:26726–26739, 2021.

[13] M Laeeq Khan and Aqdas Malik. Researching youtube: Methods, tools, and analytics. *The sage handbook of social media research methods*, pages 651–663, 2022.

[14] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019.

[15] Kayla Guangrui Li, Sunil Mithas, Zhixing Zhang, and Kar Yan Tam. How does algorithmic filtering influence attention inequality on social media? *ICIS 2019 Proceedings*, 2019.

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[17] Chetan Harichandra Mendhe, Nathan Henderson, Gautam Srivastava, and Vijay Mago. A scalable platform to collect, store, visualize, and analyze big data in real time. *IEEE Transactions on Computational Social Systems*, 8(1):260–269, 2021.

[18] Jan-Hinrik Schmidt Monika Taddicken. *Entwicklung und Verbreitung sozialer Medien*. 2017.

[19] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. Bertweet: A pre-trained language model for english tweets. *CoRR*, abs/2005.10200, 2020.

[20] Michael Orlov and Marina Litvak. Using behavior and text analysis to detect propagandists and misinformers on twitter. In *Information Management and Big Data: 5th International Conference, SIMBig 2018, Lima, Peru, September 3–5, 2018, Proceedings 5*, pages 67–74. Springer, 2019.

[21] Hegelich Papakyriakopoulos, Medina Serrano. Political communication on social media: A tale of hyperactive users and bias in recommender systems. *Online Social Networks and Media*, 15:100058, 2020.

[22] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[23] Mohiuddin Md Abdul Qudar and Vijay Mago. Tweetbert: a pretrained language representation model for twitter text analysis. *arXiv preprint arXiv:2010.11091*, 2020.

[24] Yunita Sari, Mark Stevenson, and Andreas Vlachos. Topic or style? exploring the most useful features for authorship attribution. In *Proceedings of the 27th international conference on computational linguistics*, pages 343–353, 2018.

[25] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Alessandro Flammini, and Filippo Menczer. The spread of fake news by social bots. *arXiv preprint arXiv:1707.07592*, 96(104):14, 2017.

[26] Karishma Sharma, Yizhou Zhang, Emilio Ferrara, and Yan Liu. Identifying coordinated accounts on social media through hidden influence and group behaviours. Association for Computing Machinery, 2021.

[27] Prasha Shrestha, Sebastian Sierra, Fabio González, Manuel Montes, Paolo Rosso, and Thamar Solorio. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674. Association for Computational Linguistics, 2017.

[28] Jasmeet Singh and Vishal Gupta. Text stemming: Approaches, applications, and challenges. *ACM Comput. Surv.*, 49(3), 2016.

[29] Xiaobing Sun and Wei Lu. Understanding attention for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3418–3428, 2020.

[30] Michael Thelwall. *Data Collection with Mozdeh*, pages 13–21. Springer International Publishing, 2021.

[31] Mike Thelwall. Social media analytics for youtube comments: potential and limitations. *International Journal of Social Research Methodology*, 21(3):303–316, 2018.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[33] Mirjam Wattenhofer, Roger Wattenhofer, and Zack Zhu. The youtube social network. *Proceedings of the International AAAI Conference on Web and Social Media*, 6(1):354–361, 2021.

[34] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[35] Kai-Cheng Yang, Onur Varol, Clayton A Davis, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Arming the public with artificial intelligence to counter social bots. *Human Behavior and Emerging Technologies*, 1(1):48–61, 2019.

[36] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.