Universität Zürich[UZH]

Bachelorarbeit
zur Erlangung des akademischen Grades
Bachelor of Arts
der Philosophischen Fakultät der Universität Zürich

Computergestützte Erkennung von Textwiederverwendung in Schweizer Stellenanzeigen

Verfasserin: Daniela Hofmann
Matrikel-Nr: 99-907-511

Betreuer: Dr. Simon Clematide
Institut für Computerlinguistik

Abgabedatum: 01.07.2023

Zusammenfassung

Den Untersuchungsgegenstand dieser Arbeit stellen deutschsprachige Stellenanzeigen der Jahre 2014 bis 2022 aus dem Korpus des Stellenmarkt-Monitor Schweiz dar. Regelmässig erfasst der Stellenmarkt-Monitor Schweiz Stellenanzeigen des Schweizer Arbeitsmarktes zur kontinuierlichen Entwicklungsanalyse. Textwiederverwendung als konzeptionelles Werkzeug soll im Rahmen dieser Arbeit typische, formelhafte Wendungen im Korpus sichtbar machen. Zur Detektion von wiederverwendeten Textpassagen kommt Passim zum Einsatz. Umfangreiche Experimente erfolgen, um passende Passim-Konfigurationen zu finden, auch für die Deduplikation im Vorfeld. Erste Experimente geben jedoch den Anlass, zusätzlich in der Vorverarbeitung eine Satzsegmentierung zu implementieren. Abschliessend lässt sich sagen, dass Passim keine kurzen Textbausteine direkt ausgibt. Dennoch kann die resultierende Datenstruktur im Weiteren genutzt werden und beinhaltet u. a. Metadaten zur Textähnlichkeit und zum Vokabular der gefundenen Textpassagen. Ergänzend wäre z. B. eine Kollokationsanalyse wünschenswert, um gezielter kurze, typische und formelhafte Textbausteine in den Stellenanzeigen zu erfassen.

Abstract

The subject of this study are job advertisements in German from the years 2014 to 2022 from the corpus of the Swiss Job Market Monitor. The Swiss Job Market Monitor regularly collects job advertisements from the Swiss labor market for continuous development analysis. Text reuse as a conceptual tool should make typical, formulaic phrases in the corpus visible in the context of this work. Passim is used to detect reused text passages. Extensive experiments are carried out to find suitable Passim configurations, also for deduplication in advance. However, first experiments give the reason to implement a sentence segmentation in the preprocessing as well. In conclusion, it can be said that Passim does not output short text passages directly. Nevertheless, the resulting data structure can be used further and includes among other things metadata on the text similarity and the vocabulary of the found text passages. In addition, e.g. a collocation analysis would be desirable in order to capture short, typical and formulaic text phrases in the job advertisements in a more targeted manner.

Danksagung

Mein ausserordentlicher Dank gilt allen Personen, die mir diese Bachelorarbeit ermöglicht haben.

Allen voran meinem Betreuer Dr. Simon Clematide. Er hatte immer ein offenes Ohr für meine Fragen und stand mir mit wertvollem Rat zur Seite. Seine Anregungen waren fortwährend Motivation.

Für Ihren unerschütterlichen Glauben an mich, ein liebes Merci an meine Tochter Alexandra sowie an meinen Lebenspartner Christian fürs Korrekturlesen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

BLAST	Basic Local Alignment Search Tool
COMHIS	Computational History and the Transformation of Public Discourse in Finland
DNA	Desoxyribonukleinsäure
JSON	JavaScript-Object-Notation
JSONL	JavaScript-Object-Notation-Lines
KiTAB	Knowledge, Information Technology & the Arabic Book
LIFO	Last In First Out
MWA	Mehrwortausdrücke
METER	MEasuring TExt Reuse
OCR	Optical Character Recognition
POS	Part-of-Speech
SMM	Stellenmarkt-Monitor Schweiz
STTS	Stuttgart-Tübingen-Tagset
UPOS	Universal-Part-of-Speech

1 Einleitung

Wenn man von Textwiederverwendung (engl. *text reuse*) spricht, ist das weite Konzept von Text-Recycling gemeint. Das geistige Produkt „Text“ wird dem Lese-Kreislauf erneut zugeführt. Text, der als Ganzes oder in Teilen wiederverwendet wird, ob mit oder ohne Hinweis auf den eigentlichen Urheber.

Bei meinen Recherchen zu dieser Arbeit habe ich viele verschiedene einleitende Umschreibungen gelesen. Zwei Sichtweisen möchte ich hier anführen.

Man kann Sinnzusammenhänge heranziehen, da verschiedene Begriffe Textwiederverwendung konkretisieren. Romanello und Hengchen (2021) stellen Plagiate den literarischen Phänomenen wie Anspielungen, Paraphrasen und wörtlichen Zitaten gegenüber. Im Verlags- oder Unterrichtskontext sehen sie die Plagiate, wenn Textteile einer anderen Person ohne angemessene Quellenangabe wiedergegeben werden und im literaturwissenschaftlichen Kontext Textwiederverwendung oft als ein Sinnbild für eine offene, quellengestützte Bezugnahme.

Nach Büchler et al. (2016) tritt Wiederverwendung von Text auf, wenn ein Autor Text von einem früheren oder zeitgenössischen Autor entlehnt oder wiederverwendet. Folglich sprechen sie von dem Entleiher oder dem zitierenden Autor, der den Text des zitierten Autors wortwörtlich wiedergibt oder komplett umformulieren kann und nennen diese Form der Entlehnung „beabsichtigte“ Textwiederverwendung. Unter „unbeabsichtigter“ Textwiederverwendung subsumieren sie Redewendungen oder geflügelte Worte, deren Ursprung unbekannt ist und Teil des allgemeinen Sprachgebrauchs geworden sind.

Für mich fallen weitere Möglichkeiten in dieses Reproduktionsschema. Redaktionelle Systeme z. B. für Übersetzungen und Erstellung von technischen Dokumentationen leben von Datenbanken mit allerlei Textbausteinen. Werbung benutzt den Wiedererkennungseffekt von Mustern in Phrasen und Bildern gezielt für ihre Zwecke.

So sind im Zuge der Digitalisierung Stellenanzeigen als Gebrauchstexte länger und expliziter geworden (Gnehm, Bühlmann & Clematide, 2022) und „heute mehr als nur ein Informationsträger für offene Stellen. [Sie sind] ein Werbeträger für das Firmenimage, eine Visitenkarte, der man entsprechende Aufmerksamkeit bei der Gestaltung widmet“ (Bendel, 1999, S. 7).

1.1 Motivation

Folgendes Beispiel könnte so auch heute auf dem Stellenmarkt zu finden sein. Die Wortwahl wäre wohl etwas anders:

> „Gesucht ein Knabe als Ausläufer welcher mit Ostern von der Schule entlassen wird. Velofahrer bevorzugt. Eintritt 1. April oder später bei Georg Häntzler's Erben, Metzgerei, Seefeldstrasse 181, Zürich 8.“ (Tages-Anzeiger, 17. März 1921)[1]

Beispielhaft zeigt diese Anzeige, dass Fahrradkuriere bereits vor 100 Jahren gefragt waren. Jugendliche beendeten ihre Schulzeit folglich zu Ostern und vermutlich liessen sich noch andere Dinge herauslesen.

Auch Stellenanzeigen wirken ganz klar als Spiegel der Gesellschaft und ihrer Entwicklung, weil sie aufzeigen, wie die Veränderungen aller Lebensbereiche Texten ihren Stempel aufdrücken und dennoch bleibt die Textfunktion erhalten (Meier, 2018, S. 358). Da Textbausteine in Anzeigen offensichtlich schon länger Usus sind (Bendel, 1999), muss ein reicher Fundus vorhanden sein bezüglich Geschlechterrollen, Berufsbildern oder Anforderungen. Wender und Peter (1999) forderten allerdings schon früh eine bessere Darstellung der „verwirrenden Fülle an Metainformationen“.

In ihrer Einleitung werfen Roth et al. (2017) Fragen auf, wie die Sprache die Gesellschaft durchwirkt und begrüssen die Entwicklung zu mehr interdisziplinärer Forschung. Die Analyse grosser Korpora zu ermöglichen ist wichtig, um breitflächig Informationen zur Verfügung zu stellen. So wird die Erforschung gesellschaftsrelevanter Themen unterstützt und deren Ergebnisse können Verbesserungen im Bildungswesen zur Folge haben (Cordella et al., 2020) oder das kulturelle Gedächtnis in Bibliotheken erschliessen (Stäcker, 2022) oder verloren Geglaubtes elegant wieder sichtbar machen (Büchler et al., 2014).

1.2 Forschungsfragen

Aus den vorangestellten Erläuterungen lassen sich folgende Fragestellungen ableiten, die in dieser Arbeit untersucht werden:

- Auf welcher Ebene findet Textwiederverwendung in Stellenanzeigen statt?
  - Ist damit der Sprachwandel erforschbar? Oder braucht es dazu Kollokationen?
- Lassen sich typische, formelhafte Textbausteine mit Passim finden?
  - Welche Vorverarbeitungsschritte braucht Passim?

[1] https://www.stellenmarktmonitor.uzh.ch/de/infrastructure/examples.html (Letzter Besuch: 23. Mai 2023).

- Welche sinnvollen Konfigurationen für Passim sind geeignet?
- Wie lassen sich die Resultate von unterschiedlichen Passim-Konfigurationen quantitativ charakterisieren?
- Welche Metadaten können wir bereitstellen, um anderen Disziplinen eine sinnvolle Untersuchungsplattform zu bieten?

1.3 Aufbau der Arbeit

Nach diesem einführenden *ersten Kapitel* gibt das *zweite Kapitel* einen exemplarischen Abriss der verwandten Arbeiten im Bereich der Textwiederverwendung. Der Fokus liegt auf dem Phänomen der Textviralität. Die medialen Inhalte stammen sehr oft aus Zeitungen und Zeitschriften, die grossen Global-Player vor dem Zeitalter der Digitalisierung und des Internets. Die „Kultur des Nachdruckes“ (D. A. Smith et al., 2013, S. 1) wird auch im Sinne eines Vergleichs zwischen literarischem Ursprungstext und Zielquellen beleuchtet. Es werden in diesem Kapitel zusätzliche Korpora und Implementierungen vorgestellt. Einige Bespiele in diesem Abschnitt gehen auf Arbeiten zu Stellenanzeigen ein.

Das *dritte Kapitel* fasst die theoretischen Grundlagen zusammen. Dabei werden linguistische Aspekte zum Sprachwandel, zu formelhafter Sprache und zu Mehrwortausdrücken (MWA) aufgegriffen.

Im Weiteren folgen im *vierten Kapitel* Beschreibungen zum verwendeten Korpus und den gewählten Tools.

Anschliessend führt das *fünfte Kapitel* detailliert die vorgenommenen Experimente und Ergebnisse aus.

Schlussbetrachtung und ein Ausblick auf geplante und mögliche Folgeprojekte und Fragestellungen bilden das *sechste Kapitel*.

2 Verwandte Arbeiten

2.1 Textviralität

2.1.1 Pressetexte

Gehen Inhalte heutzutage viral, so denken wir an eine rasend schnelle, breitflächige und ungefilterte Ausbreitung von Informationen, Bildern oder Video-Clips in den sozialen Medien. Gerade im historischen Forschungskontext sind Zeitungen und Zeitschriften zu den Medien der Wahl avanciert. Sie werden immer häufiger digital zugänglich gemacht und so immer attraktiver als Informationsquellen.

Clough, Gaizauskas, Piao und Wilks (2002) schliessen ihre Publikation zu *MEasuring TExt Reuse* (METER) noch mit den Worten: „Zusammenfassend lässt sich sagen, dass die Messung der Textwiederverwendung ein spannender, neuer Bereich ist, der eine Reihe von Anwendungen bieten wird, insbesondere, aber nicht beschränkt auf die Beobachtung und Überprüfung des von einer Nachrichtenagentur erstellten Nachdruckes“. Zudem stellten sie das notwendige Korpus noch von Hand zusammen (Clough, Gaizauskas & Piao, 2002, S. 1680).

Wie der Name des *Viral Texts*[2] Projekt andeutet, kann durchaus jede Form von mehrdimensionaler Ausbreitung mit Viralität umschrieben werden. Das Viral Texts Projekt (D. Smith et al., 2015; Cordell et al., 2015) untersucht die zeitliche und räumliche Verbreitung von Zeitungsartikeln in verschiedenen zeitgenössischen Print-Medien Mitte des 19. Jahrhunderts, v. a. in den USA und ausserhalb.

Aus der ursprünglichen Idee wiederverwendete Textpassagen mit weniger Restriktionen bezüglich ihrer Verbindung zum Kontext oder ihrer eigenen Ausprägung finden zu können, entstand ein „effizienter Algorithmus . . . für die Cluster-Detektion von Textpassagen, die eingebettet sein können in grossen Dokumentensammlungen mit Texten von schlechter OCR-Qualität“ (D. A. Smith et al., 2013, S. 8) und hat unter dem Namen *Passim* im Viral Texts Projekt als Open-Source-Tool seine erste Verwendung gefunden. Leider gibt es keine ausreichende Dokumentation zu Passim. Romanello und Hengchen (2021) geben aber in Form eines kurzen Tutorials eine hilfreiche erste Einführung dazu, das auch für die vorliegende Arbeit als Ausgangspunkt diente.

Was bei Cordell et al. (2023) unter dem Titel *Going the Rounds* zu einem Buch nun vereint

[2] https://viraltexts.org/ (Letzter Besuch: 20. Mai 2023).

werden soll, hat im Viral Texts Projekt seinen Ursprung. Das Buch wird einen Einblick in das interdisziplinäre Projekt bieten und einige Aspekte daraus näher vorstellen.

Die Erkenntnisse aus dem umfangreichen Projekt will Blankenship (2021) nun ebenfalls nutzen, um eine noch recht unbeachtete Textsorte zu untersuchen, nämlich Kochrezepte aus der gleichen Epoche. Nach Blankenship (2021) eine sehr reichhaltige Quelle, da zu jener Zeit viele Menschen aus dem häuslichen Umfeld nur so eine gewisse Präsenz in der Öffentlichkeit erreichen konnten.

In diesem Zusammenhang ebenfalls erwähnenswert ist das Projekt *Computational History and the Transformation of Public Discourse in Finland* (COMHIS) einer Forschergruppe aus Finnland. In einer neuen Publikation zum Projekt geben Paju et al. (2023) einen zusammenfassenden Rückblick auf die Untersuchung des „Cycles of Information Flows“ in finnischen Zeitungen und Journalen über den Zeitraum vom Ende des 18. bis Anfang des 20. Jahrhunderts. Sie geben zugleich bekannt im Folgeprojekt *Information Flows over the Baltic Sea* eine Kollektion von schwedischen Zeitungen dazuzunehmen, um den transnationalen Informationsaustausch zu untersuchen.

Für ihre Untersuchungen verwendeten sie das *Basic Local Alignment Search Tool* (BLAST). Da BLAST (McGinnis & Madden, 2004) ein Analyse-Tool aus der biomedizinischen Forschung ist, sprich zur Analyse von DNA-Sequenzen entwickelt wurde, mussten Anpassungen vorgenommen werden. Das historische, qualitativ-schlechte Optical Character Recognition (OCR)-Textkorpus sequenzierten Vesanto, Ginter et al. (2017) wie Aminosäuren in DNA-Strängen anhand 23 eindeutiger und am häufigsten vorkommender Kleinbuchstaben (Vesanto, 2018, S. 26 ff.). Vesanto, Nivala et al. (2017) konnten trotz 60% divergierender Zeichen in den OCR-Daten (Torget, 2023, S. 64) genügend wiederverwendete Textpassagen finden, die interessante Ansätze für ein Weiterverfolgen boten.

In Salmi et al. (2019) wird betont wie zwei auffallende Merkmale der Untersuchung zum Schluss führten, dass die „Gewohnheit des Nachdruckes“ eine Verbindung zwischen Vergangenheit und Gegenwart herstellt und Zeitungen wie ein Langzeitgedächtnis der Gesellschaft agieren, die Kontinuität zwischen den Epochen ebenso wie die Unterschiede darstellen können und so die Meinungen der Zeit mitgestalten.

Auch sehen Salmi et al. (2021, S. 14) in den Print-Medien des 19. Jahrhunderts die ganz eigene Form von Big-Data und das Wiederverwenden von Text als einen essenziellen Bestandteil der Presse jener Zeit. Hier ergeben sich Parallelen zur vorliegenden Arbeit. Stellenanzeigen bieten ebenfalls eine kontinuierliche Quelle. Sie waren vermutlich schon von Beginn der Industrialisierung an neben Zeitungsmeldungen sporadisch Bestandteil von Anzeigen in Zeitungen (Meier, 2018, S. 355). Auch diese Textsorte unterliegt dem Wandel der Zeit in Sprache und Ausgestaltung.

Salmi et al. (2021, S. 26) geben als Limitierungsfaktor ihrer Arbeit die Sprachgrenzen an. Sie

agieren zwischen Finnisch und Schwedisch. BLAST kann nicht ohne Weiteres Sprachgrenzen überwinden und so sprachübergreifend wiederverwendete Textpassagen als solche erkennen. Dieser limitierende Aspekt sollte mit Passim kein Hindernis darstellen, was mit ein Grund ist, für diese Arbeit Passim einzusetzen.

Ein weiterer Grund stellt das *impresso - Media Monitoring of the Past*[3] Projekt dar. Hier wurde Passim auf mehreren Verarbeitungsebenen eingesetzt (Ehrmann et al., 2020, S. 964). Nicht nur, dass es wie oben angedeutet multilingual einsetzbar ist, es lässt sich sowohl für die Deduplikation als auch für Detektion von Textwiederverwendung einsetzen. Diese Rückschlüsse lassen die Angaben der breiten Möglichkeiten, die impresso bietet, zu.

Ehrmann et al. (2020, S. 966) haben mit Hilfe von Zeitungskorpora, linguistischen Annotationen und Sprachmodellen in Französisch, Deutsch, Luxemburgisch und Englisch Datensätze zusammengestellt und zu einem grossen Ganzen vereint. Sie decken einen Zeitraum von ca. 200 Jahren ab und erwähnen das Problem von diachronen Untersuchungen von Sprache (Ehrmann et al., 2020, S. 959), da diese stetem Wandel unterworfen ist. Damit sind auch die Ziele der vorliegenden Arbeit umrissen. Aus einer grossen, noch wachsenden und i. e. S. unzusammenhängenden Textmenge eine syntaktisch annotierte, mit Metadaten angereicherte und explorierbare Datenstruktur zu schaffen.

2.1.2 Literarische Texte

Das Detektieren von wiederverwendetem Text ist aber keineswegs eine nur auf die Welt der Presse beschränkte Fragestellung. Im Zuge der fortschreitenden Entwicklungen werden immer mehr Textsorten aufgegriffen.

Bei der Auswahl spielen oft sehr individuelle Aspekte eine Rolle. Beachtung finden Datensammlungen aus Judikative und Administration (Wilkerson et al., 2015; Koolen & Hoekstra, 2022) oder der Finanzwelt (Radford et al., 2009), aber auch aus den Bereichen Theologie (Tiepmar et al., 2014), Architektur (Ball, 2022), Medizin (D'hondt et al., 2016), Informatik (X. Xu et al., 2021; Loose et al., 2008) oder der Wissenschaft allgemein (Citron & Ginsparg, 2015; Potthast et al., 2013).

Dabei hat die Entwicklung hin zum Open-Access von wissenschaftlichen Publikationen, die Möglichkeiten interdisziplinäre Korpora zu analysieren, voran gebracht (Gienapp et al., 2022). Mit dem Bereich von Big-Data assoziiert sind die sozialen Medien und Wikipedia als reiche Quellen für die Frage nach wiederverwendeten Textpassagen (S. Xu et al., 2014; Alshomary et al., 2018). In einem ganz anderen Rahmen bewegen sich Untersuchungen von Musiktexten (Meinecke & Jänicke, 2020; Meinecke et al., 2022).

[3] https://impresso-project.ch/ (Letzter Besuch: 20. Mai 2023).

Literarische Werke oder berühmte Autoren stossen ebenfalls interdisziplinär Projekte an. Eine im 19. Jahrhundert beliebte Form von Informationsträger waren die Enzyklopädien, die zusammen mit Wörterbüchern erwartungsgemäss aus vielen einzelnen, nicht immer genügend vermerkten Quellen zusammengesetzt wurden. Nicht selten verband man einzelne Autoren eng mit bestimmten Editionen. Hier tragen solche Untersuchungen dazu bei, „versteckte Wege“ zu finden und ein klareres Bild zu gewinnen (Olsen et al., 2011).

Denkt man an berühmte Autoren der Geschichte, so wird Shakespeare unter den Top-Ten auftauchen. Unter dem Namen *The Vectorian*[4] soll nun das gesamte Werk von Shakespeare durchsuchbar gemacht werden. Allerdings befindet sich das Projekt noch im Experimentierstadium (Liebl & Burghardt, 2020).

Soweit abgeschlossen ist dagegen das internationale Projekt von O'Neill et al. (2021). Die Forschergruppe hat sich gefragt, wie stark ein Autor von seinen eigenen Recherchen, in diesem Fall in einer Bibliothek, in seiner Arbeit und in seiner Meinung beeinflusst wird. Erneut kam Passim in diesem Projekt erfolgreich zum Einsatz.

Als Untersuchungsgegenstand nahm sich das Projekt den gesammelten Werken des britischen Autors, Philosophen, Politikers und Ökonomen John Stuart Mill an. Er gilt als einer der prägendsten Denker des 19. Jahrhunderts bezüglich der heute immer noch oder wieder aktuellen Themen wie Frauenemanzipation, Vermögensverteilung sowie Presse- und Meinungsfreiheit[5].

Sein Lebenswerk ist dabei eng mit *The London Library* verbunden (O'Neill et al., 2021, S. 1015). Die beschriebene Passim-Konfiguration (O'Neill et al., 2021, S. 1019) diente dazu von Mill ausgeliehene Bücher mit seinen Publikationen abzugleichen. Passim zeigte wieder seine Stärke gegenüber OCR-Fehlern.

Interessant im Zusammenhang mit der vorliegenden Arbeit ist, dass Passim viele „Redewendungen“ fand, was O'Neill et al. (2021) im Zusammenhang mit ihrer Forschung als „falschpositive“ Treffer bezeichnen. Sie betonen, dass die Passim-Funde noch viel manuelle Analyse erfordern werden.

Eine Haupterkenntnis, die sie beschreiben, zeigt Mill jetzt schon als äusserst gewissenhaften Kennzeichner seiner Quellen (O'Neill et al., 2021, S. 1021).

Als Co-Autor lanciert D. Smith (2019) ein Folgeprojekt[6], eine Untersuchung digitaler Sammlungen im Bibliotheksmassstab zu kommentierten Seiten historischer Bücher, mit dem Ziel die erste automatisch generierte Datenbank handschriftlicher Anmerkungen zu erstellen.

Latein galt lange als die Sprache der Gelehrten und viele bekannte historische Autoren verfassten ihre Texte in dieser heute nicht mehr im Alltag gebräuchlichen[7] Sprache. Somit bleibt

[4] https://github.com/poke1024/vectorian (Letzter Besuch: 16. Mai 2023).
[5] https://www.britannica.com/biography/John-Stuart-Mill/Public-life-and-writing (Letzter Besuch: 17. Juni 2023).
[6] https://app.dimensions.ai/details/grant/grant.8385506 (Letzter Besuch: 19. Mai 2023).
[7] https://www.ethnologue.com/language/lat/ (Letzter Besuch: 19. Mai 2023).

sie ein wichtiger Untersuchungsgegenstand (Peverelli et al., 2022; Franzini et al., 2018; Coffee et al., 2013). Wenig beachtete Sprachen dagegen bekommen erst langsam ihren Platz in der Forschung (Sameen et al., 2018; Elagina, 2022; Guillaume et al., 2022).

Historische Sprachstufen stellen eine Herausforderung bei der Erschliessung mittels OCR und der Detektion von Textwiederverwendung dar. Immer mehr Projekte versuchen tolerantere Tools zu entwickeln (Düring & van den Bosch, 2014; Büchler, 2016; Shang & Underwood, 2021; Gladstone & Tharsen, 2022; Belinkov et al., 2019; Manjavacas, 2020; Arnold & Jäschke, 2021).

Im Langzeit-Projekt *Knowledge, Information Technology & the Arabic Book* (KiTAB)[8], bei dem ebenfalls Passim eingesetzt wird, versucht man ein umfassendes Bild der Textüberlieferung und des Sprachwandels im Arabischen zu erstellen. Es werden mindestens zweimal im Jahr Passim-Läufe durchgeführt und die Ergebnisse auf der Webseite des Projekts veröffentlicht. Neuerungen im Korpus werde so Rechnung getragen. Häufigere Aktualisierungen seien nicht möglich, da Vor- und Nachbereitung der Daten sehr zeitintensiv ausfallen[9].

Die bereits zu Beginn dieser Arbeit aufgeworfene Forschungsfrage nach der Vorverarbeitung rückt somit etwas mehr in den Fokus, ergänzt durch die Frage nach dem Aufbereitungsaufwand der Passim-Ausgabe.

Barber (2023) betont den Vorteil von Passim, dass grosse Textsammlungen mit einer sehr guten Trefferquote durchsuchbar sind. Innerhalb des KiTAB-Projektes gelten als Textwiederverwendung „Fälle, in denen ein Buch wörtliche Auszüge mit einem anderen teilt“. Nach seiner Meinung liegt die grosse Schwäche von Passim in der Tatsache, dass aber bei zu starker Verzerrung des einen Textabschnittes keine Alignierung zu einer wiederverwendeten Textpassage möglich ist, weil Passim kein inhaltliches Verständnis zugrunde liegt und für optimale Forschung eine Verbindung zwischen Maschine und Mensch wünschenswert bleibt.

Hiltmann et al. (2021, S. 122) merken kritisch an, dass im Rahmen von Digital-History oft grosse Erwartungen vorherrschen und Forschende eine Rechtfertigung der Anstrengungen mit herausragenden Ergebnissen belegen können sollten. Auf diesbezügliche Kontroversen weisen auch Lynch (2020) und Ehrmann et al. (2019) hin, aber sehen ebenso in der vernünftigen Anwendung von Digital-Humanities-Tools eine enorme Bedeutung für diejenigen, die mit alten Texten arbeiten (Lynch, 2020, S. 17).

Jedoch werfen Ehrmann et al. (2019, S. 17) die Frage auf, inwiefern die gewonnen Erkenntnisse im Anschluss zugänglich gemacht werden können, angesichts der Komplexität der Daten und der Möglichkeiten erforderlicher Schnittstellen. Eine auch in dieser Arbeit sich stellende Frage.

[8] http://kitab-project.org/ (Letzter Besuch: 17. Juni 2023).
[9] http://kitab-project.org/methods/text-reuse (Letzter Besuch: 17. Juni 2023).

Kumpulainen und Late (2022) griffen diese Thematik bereits auf und resümieren: „dass Hindernisse in verschiedenen Bereichen bestehen“, z. B. technische in Zusammenhang mit Schnittstellen und Tools und „ein forschungsbasierter Einblick in Hindernisse, mit denen man konfrontiert ist, dabei helfen kann, herauszufinden, was dafür erforderlich ist, gute Unterstützung bereitzustellen“.

Davor kamen Pfanzelter et al. (2020) zu dem Schluss, dass digitale Tools bevorzugt die „quantitative Makroanalyse (Big-Data-Analyse) und qualitative Mikroanalyse (Lesen)“ kombinieren und beides parallel implementieren sollten.

2.1.3 Stellenanzeigen

Auch in dieser Textsorte lässt sich der stete Sprachwandel verfolgen (Meier, 2018; Bendel, 1999; Ladstätter, 2004). Inwiefern dieser Wandel feste Textbausteine (Bubenhofer, 2017) betrifft, dazu untersucht die vorliegende Arbeit diese alltägliche Textsorte auf formelhafte MWA (Iwatsuki et al., 2020).

Darüber hinaus bilden Stellenanzeigen einen Unterzweig der Werbung, der insgesamt weniger untersucht wird als Zeitungsartikel (Wevers, 2023). Poch et al. (2014) zeigen aber auf wie viel Big-Data darin verborgen liegt. Untersucht werden kann z. B. wie mit Ethnien umgegangen wird (Adams et al., 2022), in dem Zusammenhang auch das Anonymisieren eine Rolle spielt (Jensen et al., 2021), wie Vergleiche der Stellenanzeigen den Job-Finde-Prozess beeinflussen (Elsafty et al., 2018) oder gewisse Fähigkeiten (engl. *skills*) die Aussichten verbessern (Kiener et al., 2022; Zhang, Jensen et al., 2022; Zhang, Jensen & Plank, 2022; Gnehm, Bühlmann, Buchs & Clematide, 2022).

Diese Arbeit wird in Kapitel 5 auf die von Gnehm (2018) und Gnehm, Bühlmann, Buchs und Clematide (2022) vorgestellten, einschlägigen Textzonen zurückgreifen. Im Gegensatz zu dieser Arbeit verwendeten Gnehm und Clematide (2020) mehrsprachige (engl. *multilingual*) Datensätze für Untersuchungen von Textzonen in Deutsch, Französisch und Englisch.

2.2 Plagiatserkennung

Ein Plagiat ist „die ganze oder teilweise Übernahme eines fremden Werks ohne Angabe der Quelle, welche Rückschluss auf den Urheber oder die Urheberin des fremden Werks gibt“[10].

Im akademischen Raum stellen Plagiate eine immer grösser werdende Herausforderung dar und werden vorwiegend in diesem Kontext untersucht (Francopoulo et al., 2016; Moritz et al.,

[10] https://www.disziplinarkommission.uzh.ch/de/disziplinarfehler.html#Plagiate (Letzter Besuch: 19. Juni 2023).

2018; Mariani et al., 2016; Unger et al., 2016; Belyy et al., 2018; Wahle et al., 2022).

Auf Plagiatserkennung im akademischen Umfeld bezieht sich die Mehrheit der Arbeiten zu sprachübergreifender (engl. *crosslingual*) Textwiederverwendung wie z. B. in Kothwal und Varma (2013), Bakhteev et al. (2022), Bakhteev et al. (2019) und Avetisyan et al. (2023).

Bei der sprachübergreifenden Textwiederverwendung wird bereits vorhandener Text in einer Quellsprache übernommen, um einen neuen Text in einer anderen Zielsprache zu erstellen.

Plagiatserkennung wird in dieser Arbeit nicht weiter thematisiert. Formelhafte und spezifische Textbausteine bilden den Untersuchungsgegenstand.

3 Theoretischer Hintergrund

Bussmann (2008, S. 719) beschreibt einen Text ganz allgemein als „formal begrenzte, schriftliche Äusserung, die mehr als einen Satz umfasst“ und linguistisch motiviert als „monologische, im prototypischen Fall schriftlich fixierte, sprachliche Einheit, die insgesamt als sinnvolle, kommunikative Handlung intendiert oder rezipiert wird“.

Der Begriff wird von Dipper et al. (2018) dahingehend ergänzt, dass weder das Trägermedium noch die Anzahl der Sätze weder ein notwendiges noch hinreichendes Kriterium bilden, ja sogar grammatisch unvollständige Sätze eine sprachliche Einheit und damit ein Text sein können. Beispielhaft führen Dipper et al. (2018, S. 146) die Werbung an, in der *Toyota: Nichts ist unmöglich* und *Soo! Muss Technik* ganz selbstverständlich Texte sind. Folglich sind Texte in Schranken gewiesene Wortketten, divers in ihrer Funktionalität sowie varianten- und formenreich.

In diesem Sinne verkörpern auch Stellenanzeigen Texte, ob sie allerdings eine eigene Textsorte bilden, bleibt offen (Meier, 2018, S. 356).

Crochemore et al. (2021) haben eine mehr technische Definition, bei der Text aus Sequenzen von Worten oder Zeichenketten besteht, mit Algorithmen bearbeitbar und eines der wichtigsten unstrukturierten Datenformate in der Informatik ist.

3.1 Sprachwandel

Sprachwandel zeigt sich, in dem etwas anders ist, als man selbst es kennt. Bechmann (2016, S. 70) zeigt zwei mögliche Standpunkte auf, wobei man entweder retrospektiv den Sprachzustand mit dem Jetzt-Zustand vergleicht, dann spricht man von einer *diachronen* Perspektive, oder eine *synchrone* Perspektive einnimmt und zeitgleich auftretende Sprachphänomene betrachtet.

Untersuchbar ist Sprachwandel auf verschiedenen Ebenen bezüglich *kommunikativer Bedürfnisse* von Sprechern und in verschiedenen *linguistischen Dimensionen* (Bechmann, 2016, S. 68 f.). Es zeigen sich wieder zwei Standpunkte. Die Sprache ist für Sprecher ein Mittel zum Zweck und als solche Veränderungen ausgesetzt. Sprache als Entität kann aber so stark gewandelt werden, dass neue Sprachen entstehen können.

Bechmann (2016) kondensiert Sprachwandel in den folgenden Eigenschaften: vom Menschen gemacht, planlos, unstet und dennoch kontinuierlich. Welche Bedingungen zum Sprachwandel führen, also punktuelle Änderungen, die neues Sprachhandeln anhand neuer Handlungsgrundsätze hervorrufen, liegen in vier Determinanten des Sprachwandels begründet (Bechmann, 2016, S. 105 f.).

Soziale Beweggründe für Sprachwandel liegen oft in sozialen Konventionen begründet. Sprache will optimiert sein. Wir versuchen möglichst viel durch möglichst wenig auszudrücken und so kooperativ auf andere einzuwirken und trotzdem unsere intendierten Ziele zu erreichen. Das Miteinander führt immer wieder zu Sprachkontakten, die zu Vermischungen verschiedener Ausdruckweisen und zu ganz neuen Sprachen leitet. Zeitlich begrenzte Sprachentwicklungen in diesem Kontext, auch temporäre Soziolekte genannt, machen den Sprachwandel auch umkehrbar (Bechmann, 2016, S. 109 ff.).

Die *Kreativität* des menschlichen Geistes ist grenzenlos und so die weitreichendste Quelle für Sprachwandel. Neben Bestrebungen Sprache zu pflegen und zu erhalten, gehen u. a. Poesie, Medien, Werbung und Politik oft eigene Wege des Sprachgebrauchs (Bechmann, 2016, S. 121 ff.).

Kognitive Fähigkeiten sind essenziell, um sprechen zu können. Mit ihrer Hilfe finden wir uns im Sprachsystem zurecht, das wir im Gehirn durch unsere Erfahrungen abgelegt haben. Diese Erfahrungen abzurufen ist mitunter fehlerhaft. Daraus können sprachliche Regelumbildungen erfolgen, z. B. zu Neudeutungen oder Bedeutungsübertragungen (Bechmann, 2016, S. 117 ff.).

Die *anatomisch-physiologischen Voraussetzungen* des menschlichen Sprechapparats als mitbestimmender Faktor bringen erneut das Ökonomieprinzip ins Spiel. Es ist einfach artikulatorisch komfortabler gewisse Lautkombinationen auszusprechen als andere, somit werden diese bevorzugt. Dazu gehören i. d. R. auch die Lautkombinationen aus unserer Muttersprache. Andere fremde Sprachlaute lassen sich aber mit Aufwand einüben (Bechmann, 2016, S. 119 ff.).

Das erweiterte 4-Phasen-Modell von Bechmann (2016, S. 75) fasst bildhaft zusammen, dass „die systematischen Fehler von heute (bei hoher Ausbreitungsfrequenz) die neuen Sprachverwendungsregeln von morgen sind“.

Zu Sprachwandel in der Werbesprachenforschung gibt Meier (2018) einen Einblick in die Geschichte der Werbeanzeige im deutschen Sprachraum, auch Stellenanzeigen werden erwähnt. Eine starke Betonung liegt auf der Tatsache, dass die Forschung noch viele offene Fragen unbeantwortet lässt. Mit ein Grund könnte sein, dass Werbung besonders abhängig vom jeweiligen Zeitrahmen ist. Bendel (1999) sieht den Sprachwandel lediglich bedingt als Indiz des sozialen Wandels in der Gesellschaft und in der Stellenanzeigensprache v. a. auch in

den „Worten, die die harten Fakten einkleiden“ (Bendel, 1999, S. 5). Erste Nachweise von der Entstehung eines Stellenmarktes gehen bis ins 18. Jahrhundert zurück und einzelne Stellenanzeigen lassen sich bereits im 17. Jahrhundert finden (Meier, 2018, S. 354 f.).

Das langfristige Ziel des Stellenmarkt-Monitor Schweiz (SMM) (Weiteres dazu siehe Abschnitt 4.1) besteht auch in der Analyse von langfristigen und systematischen Entwicklungen auf dem Stellenmarkt. Aufgrund der Datenauswahl (Weiteres dazu siehe Abschnitt 4.2) der vorliegenden Arbeit ist Sprachwandel schwer als solcher zu untersuchen. Zudem erfordert dieses Phänomen, wie oben angedeutet, eine interdisziplinäre Untersuchung. Im Rahmen dieser Arbeit ist dies so nicht möglich. Die Resultate dagegen können in Zukunft einen Beitrag leisten.

3.2 Formelhafte Sprache

Das Beispiel des Zauberspruchs aus Wray (2013, S. 190 f.) beschreibt eine Ausprägung formelhafter MWA. Jeder kennt aus dem ein oder anderen Grund eine Zauberformel und ihre Wirkung. Was macht nun aber eine Wortverkettung zu einem formelhaften MWA.

In Filatkina (2018, S. 9 ff.) spannt die Begriffsentstehung der *Formelhaftigkeit* oder der *Formulae* im Sprachkontext einen Bogen von Anfang des 20. Jahrhunderts bis zur heutigen Phraseologieforschung und folgert, „dass Phraseologismen als wichtige Komponenten des nominativen Systems der Sprache, des Lexikons und der Kultur verstanden werden müssen“ (Filatkina, 2018, S. 15).

Allgemein ergeben sich nach Wood (2015, S. 3) drei wesentliche Merkmale von formelhafter Sprache. Es handelt sich um einen MWA, d. h. mindestens zwei Wörter sind durch Rekurrenz zu einer mehr oder weniger festen Wortkette verbunden (Stumpf, 2015, S. 19). Dieser MWA besitzt eine Bedeutung oder eine gemeinsame Funktion. Der MWA ist als Ganzes im Sprachgedächtnis abgelegt und wird wie ein Wort abgerufen (Wray, 2013, S. 12 ff.) und Lin (2018, S. 12) ergänzt, dass daraus die „formale Fixiertheit, phonologische Kohärenz und semantische Nicht-Kompositionalität“ von formelhafter Sprache direkt folgt.

Darüber hinaus existieren auch andere Formen von gesellschaftlichen Normen zur mündlichen oder schriftlichen Ausdrucksweise (Filatkina et al., 2018, S. 1 f.), die auf dem Weg eines Festigungsprozesses sind und bereits eine gewisse Formelhaftigkeit besitzen (Stumpf, 2015, S. 19 f.).

Als Untersuchungsgegenstand interessant und für eine ganze Reihe von Publikationen herangezogen (Wood, 2015, S. 1 f.), wird die vorliegende Arbeit nicht vertieft darauf eingehen können. In Abschnitt 3.3 folgen eingehendere Begriffserläuterungen.

3.3 Mehrwortausdrücke

Die Mehrdeutigkeit und unklare Definitionslage in der Literatur bezüglich MWA (engl. *multi word expressions*) zeigt sich einmal mehr bei Bilgin (2022, S. 37). Recht oft definiert sind MWA als mindestens zwei Wörter, die in Texten häufiger zusammen vorkommen, als rein zufällig zu erwarten wäre (Filatkina, 2018, S. 47). Unter den Oberbegriff lassen sich dennoch verschiedene linguistische Konzepte subsummieren (Ramisch & Villavicencio, 2022).

Kookkurrenz ist eine „grundlegende syntaktische Relation, die das Miteinandervorkommen von sprachlichen Elementen verschiedener Klassen in Sätzen bezeichnet“ (Bussmann, 2008, S. 375). Bubenhofer (2017) konkretisiert, dass nicht alle Kookkurrenzen Kollokationen sind, aber beide MWA statistisch signifikant oft zusammen auftretende lexikalische Einheiten bilden.

Kollokationen sind „charakteristische, häufig auftretende Wortverbindungen, deren gemeinsames Vorkommen auf einer Regelhaftigkeit gegenseitiger Erwartbarkeit beruht, also primär semantisch (nicht grammatisch) begründet ist“ (Bussmann, 2008, S. 345), aber nicht oder nur schwach idiomatische Wortverbindungen formen (Hausmann, 2004, S. 312 ff.).

Phrasem beschreibt als „Grundeinheit der Idiomatik“ (Idiom) eine „abstrakte funktionale Einheit“ (Bussmann, 2008, S. 275). Abstrakt bedeutet hier, dass die Bedeutung nicht wörtlich abzuleiten ist (Filatkina, 2018, S. 4). Funktional tragen Phraseme teilweise zur Textausformung bei und dies tritt v. a. in der Anzeigenwerbung deutlich zu Tage (Sabban, 2004, S. 242 ff.).

Die vorliegende Arbeit sucht nach formelhaften Textbausteinen, die sich über die Zeit verändern und entwickeln. Diese Textbausteine müssen aber keine vordefinierten Grenzen aufweisen. Dennoch erschienen Kollokationen und Phraseme gleichermassen interessant (Iwatsuki & Aizawa, 2018, S. 2680).

Phraseme sind vielfältiger in ihrer Form, wobei Kollokationen z. B. in Form von n-Grammen näher spezifizierbar sind. Somit stellt die Textwiederverwendung eine sinnvolle Untersuchungsmethode der Phraseme dar.

Inwieweit Kollokationen mit Passim detektierbar sind, wird sich zeigen müssen. Allgemein gibt es immer noch zu wenig Forschung zur Detektion von langen, idiomatischen MWA (Iwatsuki & Aizawa, 2021; Madabushi et al., 2022).

3.4 Clusteranalyse

Clusteranalyse bezeichnet eine Familie von Methoden für das Gruppieren (engl. *clustering*) von sprachlichen Objekten, in homogene Verbände (engl. *cluster*), die sich gegenseitig aber heterogen verhalten. Die Gruppen bilden sich anhand bedingter Merkmale (engl. *features*) und

ihrer Ausprägung. Aber auch hier divergiert die Begriffsdefinition je nach Forschungsumfeld. Die Ausführungen in diesem Abschnitt lehnen sich an Tan et al. (2020, S. 310 ff.) an.

Clustering lässt sich wie folgt unterteilen:

- flach oder hierarchisch (dazu je agglomerativ oder divisiv),
- exklusiv oder überlappend (engl. *overlapping*) oder weich (engl. *fuzzy*) und
- ganz (engl. *complete*) oder partiell.

Hierarchisches Clustering führt dabei zu einer Baumstruktur mit Unter-Clustern und das *flache* Clustering zu einer reinen Aufteilung der Daten.

Teilt das Clustering jedem Cluster nur ein einzelnes Datenobjekt zu, dann spricht man von *exklusivem* Clustering. *Überlappendes* Clustering ermöglicht einem Datenobjekt in mehreren Clustern gleichzeitig eingeordnet zu werden. Datenobjekte in alle Cluster einzuordnen und dabei jeweils eine Gewichtung der Ähnlichkeit zu den anderen Objekten im Cluster mitzugeben, nennt man *Fuzzy-Clustering*.

Beim *Complete-Clustering* werden alle Datenobjekte auf die entstehenden Cluster verteilt, dagegen mit *partiellen* Clustering unbrauchbare Daten ausgeschlossen.

In dieser Arbeit bezieht sich nur Abschnitt 4.3 auf das in Passim implementierte Clustering. Deshalb wird nur auf das hierarchisch-agglomerative Clustering (Tan et al., 2020, S. 336 f.) näher eingegangen.

Hierarchisches Clustering kann agglomerativ oder divisiv sein. Häufiger kommt der *agglomerative* Ansatz vor, dabei ist jedes Datenobjekt von Beginn allein in einem Cluster, bei jedem Folgeschritt werden nahe Paare von Clustern zusammengefasst. *Divisiv* startet hingegen mit einem grossen Cluster, der alle Datenobjekte enthält und anhand von vorher definierten Regeln aufgetrennt.

Das hierarchisch-agglomerative Clustering ist eine einfache, aber weitverbreitete Technik. Essenziell ist die Konkretisierung der Clusterannäherung.

Die Cluster können graphisch als räumliche Objekte verstanden werden und haben so eine gewisse Nähe oder Distanz zueinander. In der Nähe verbinden zwei Cluster ihr kleinster Abstand (engl. *single link*), aber in der Distanz ihre entfernteste Ausdehnung (engl. *complete link*) (Tan et al., 2020, S. 337 ff.).

Um diese Clusterannäherung zu quantifizieren, kommen Ähnlichkeits- oder Distanzmasse zum Einsatz. Die Single-Linkage-Methode beurteilt somit die Homogenität zwischen zwei Clustern anhand der zwei sich ähnlichsten Datenobjekte (Aggarwal, 2018, S. 93).

3.5 Textähnlichkeitsmasse

Die in Abschnitt 3.4 erwähnten Ähnlichkeitsmasse sind nicht nur ein wichtiges Kriterium beim Clustering (Zong et al., 2021, S. 125). Sie werden auch oft in der Detektion von Textwiederverwendung eingesetzt (Bär et al., 2012, S. 168). Auch bei den Ähnlichkeitsmassen gibt es eine breite Palette, weil die Wahl des richtigen Masses von der Fragestellung und den vorhandenen Daten abhängt (Wise, 1993; R. Xu & Wunsch, 2005; Cha, 2007).

Dabei scheint es, dass die einen Masse eine Distanz und Andere eine Ähnlichkeit quantifizieren. Zwei Objekte können sich eben ähnlich oder unähnlich sein. Häufig wird Ungleichheit mit Distanz gleichgesetzt, obwohl keine abschliessende Definition existiert (Tan et al., 2020, S. 91 ff.).

Die vorliegende Arbeit verwendet zur Bestimmung der Dokumentenähnlichkeit innerhalb der Passim-Cluster (Weiteres dazu siehe Kapitel 5) den Jaccard-Koeffizienten. Es handelt sich um ein einfaches, gut implementierbares und allgemein übliches Mass für die Untersuchung von Textähnlichkeiten.

Der *Jaccard-Koeffizient* ist definiert als der Betrag der Schnittmenge durch den Betrag der Vereinigungsmenge, wenn eine Menge A und eine Menge B gegeben sind (Kosub, 2016):

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Sind A und B jeweils eine Menge von Wörtern beziehungsweise n-Grammen eines Textes, so berechnet der Jaccard-Koeffizient das Verhältnis zwischen der Anzahl gemeinsamer n-Gramme zweier Texte (Schnitt) zur Gesamtanzahl n-Gramme in mindestens je einem Text (Vereinigung) (Zong et al., 2021, S. 128).

Je näher der Betrag zu 1 strebt, desto gleicher sind sich die Texte. Gibt es also keine gemeinsamen Wörter ist der Betrag 0 und die Texte sind unabhängig, sind alle Wörter gemeinsam ist der Betrag 1 und die Texte absolut gleich (Manning & Schütze, 1999, S. 299). Anhand dieses Masses können somit die im Cluster von Passim jeweils zusammengefassten Dokumente nach der Überlappung ihres Vokabulars beurteilt werden.

4 Daten und Methoden

Das vierte Kapitel beleuchtet einmal die verwendeten Daten bezüglich ihrer Erhebung, Herkunft und Auswahl sowie die eingesetzten Tools. Jedes Tool wird hinsichtlich des zugrundeliegenden Algorithmus, der benötigten Daten-Formate, möglicher Konfigurationen und Limitationen vorgestellt.

4.1 Stellenmarkt-Monitor Schweiz

Das in dieser Arbeit verwendete Stellenanzeigen-Korpus stellt einen Auszug des umfangreichen SMM[11]-Datensatzes dar. Das gesamte Korpus umfasste für den Zeitraum von 1950 bis 2014 ca. 80'000 Stellenanzeigen (Gnehm & Clematide, 2020, S. 84). Diese stammen seit 2001 auch von Firmenwebseiten und seit 2006 von Jobportalen.

Seit 2012 kommen jährlich ca. 1,3 Mio. neue Stellenanzeigen aus dem Web dazu. Firmenbefragungen sind dabei ein Teil des Qualitätsmanagements. Sie sollen sicherstellen, dass 95% der auf Jobportalen veröffentlichten Inserate erfasst werden und repräsentativ für das Firmenprofil der Schweiz sind. Seit 2018 ergaben die Befragungen, dass Erhebungen von Stellenanzeigen in Printmedien keinen Mehrgewinn mehr darstellen und fielen somit weg. Bis dahin erfasste man jährlich 500 bis 650 Inserate aus einem Pool von 90 Schweizer Pressetiteln.

Die Daten liegen in zwei Datensätzen vor, einmal Volltextdaten und einmal angereichert mit hochqualitativen Metainformationen (Buchmann et al., 2022, S. 3 f.). Das Anreichern eines Korpus durch linguistische Informationen stellt gerade bei speziellen Textsorten, wie z. B. Sugisaki et al. (2018) bei Postkarten detailliert aufzeigen, oft einen umfangreichen Vorverarbeitungsschritt dar.

Auch für das SMM-Korpus kommen manuelle und maschinelle Verfahren (Tools z. T. bereitgestellt von *spaCy*[12]) zum Einsatz. Die Stellenanzeigen sind u. a. kodiert mit der jeweiligen Sprache (Deutsch, Französisch, Italienisch und Englisch), dem Jahr der Erhebung und mit Quellenangaben. Für diese Arbeit interessante Metadaten stellen die Korpus-Annotationen zu syntaktischen Basiskategorien (Part-of-Speech (POS)) und Textzonen dar.

[11] https://www.stellenmarktmonitor.uzh.ch/de.html (Letzter Besuch: 23. Mai 2023).
[12] https://spacy.io/ (Letzter Besuch: 12. Juni 2023).

4.1.1 Wortarten

Die Annotation der einzelnen Token (Grundeinheiten eines Textes) mit ihrer jeweiligen Wortart, das POS-Tagging, beinhalten sowohl ein auf dem Stuttgart-Tübingen-Tagset (STTS) basiertes Datenfeld „pos_fine“ als auch ein auf dem Universal-Part-of-Speech (UPOS)-Tagset[13] basierendes Datenfeld „pos_broad“.

Wie Petrov et al. (2011) ausführen, können die häufigsten POS in nahezu allen Sprachen auf wenige, essenzielle Kategorien reduziert werden und so eine sprachübergreifende, maschinelle Verarbeitung vereinfachen. Diese Arbeit verwendet somit nur das Datenfeld „pos_broad“ (Tabelle 4.1). Es können so genügend linguistische Informationen bereitgestellt werden, um die hier untersuchten Fragestellungen zu beantworten und darüber hinaus Erweiterungen auf weitere Sprachen und Anwendungen zu ermöglichen.

Adjektiv	ADJ	Kardinalzahl	NUM
Adposition	ADP	Partikel	PART
Adverb	ADV	Pronomen	PRON
Hilfsverb	AUX	Eigenname	PROPN
Koordinierende Konjunktion	CCONJ	Unterordnende Konjunktion	SCONJ
Artikel & pronominales Adjektiv	DET	Symbol	SYM
Interjektion	INTJ	Verb	VERB
Substantiv	NOUN	Nicht zuordenbar	X

Tabelle 4.1: Die UPOS-Tags im SMM-Korpus

4.1.2 Textzonen

Aggarwal (2018, S. 272) umschreibt Textzonen als unterschiedliche Abschnitte in Dokumenten, die über inhaltliche Abhängigkeiten eine Einheit bilden. Stellenanzeigen gliedern sich in solche differenzierbaren Inhalte. Gerade für die Analyse von Arbeitsmarktentwicklungen stellt die Fähigkeits- und Aufgabenextraktion eine interessante und vieluntersuchte Quelle dar (Zhang, Jensen et al., 2022). Folglich wird das SMM-Korpus seit 2014 vorwiegend manuell mit Informationen zu den Textzonen annotiert (Gnehm & Clematide, 2020, S. 84).

Gnehm (2018) und Gnehm, Bühlmann, Buchs und Clematide (2022) zeigen Möglichkeiten auf, diesen sehr wichtigen, frühen Vorverarbeitungsschritt zu automatisieren und mit wachsender Datenmenge zu beschleunigen. Nicht alle Textzonen sind aber in allen Stellenanzeigen vertreten und oft machen Ambiguitäten und verstreute Zonenverteilungen Probleme (Gnehm,

[13] https://universaldependencies.org/u/pos/index.html (Letzter Besuch: 23. Mai 2023).

Bühlmann, Buchs & Clematide, 2022; Gnehm, Bühlmann & Clematide, 2022).

In Tabelle 4.2 sind die acht Textzonen des SMM-Korpus ersichtlich, die in Gnehm (2018) näher beschrieben werden. Da die Fragestellungen in dieser Arbeit sich auf erforderliche Fähigkeiten, spezifische Eignung sowie Persönlichkeitsmerkmale beziehen, fokussiert sich die Datenaufbereitung auf die Zonen 60, 70 und 80.

Textzone	Definition	Beispiele
10	Unternehmensbeschreibung	„ein erfolgreiches Unternehmen der Baubranche“
20	Grund der Vakanz	„für unsere neu eröffnete Filiale“
30	Administration & Resttext	„Ihre Bewerbung senden Sie an“, „wir suchen“
40	Beschreibung einer Stellenagentur	„Ihr kompetenter Partner für die Vermittlung von Temporär- und Dauerstellen“
50	Materielle Anreize	„den hohen Anforderungen entsprechendes Salär“
60	Stellenbeschreibung	„eine vielseitige Tätigkeit“
70	Erforderliche Ausbildung & fachliche Qualifikation	„Sie haben eine Ausbildung und Berufserfahrung als Sozialarbeiter“
80	Erforderliche Persönlichkeit	„Sie sind belastbar, zuverlässig und diskret“

Tabelle 4.2: Definitionen mit Beispielen der Textzonen des SMM-Korpus (Gnehm, 2018)

4.2 Daten-Auswahl

Die vorliegende Arbeit beschränkt sich auf den Zeitraum von 2014 bis 2022 und auf deutsche Stellenanzeigen, da der überwiegende Anteil der annotierten Daten für Deutsch vorliegt (Gnehm & Clematide, 2020, S. 86). Zu Testzwecken kommen die Jahrgänge 1980 - 1989 als kleineres Datenset dazu. Die Verteilung der insgesamt verwendeten Stellenanzeigen (n = 428,6k) und Token (n = 123,5Mio.) der Jahre 2014 - 2022 zeigt Tabelle 4.3.

	Anzahl Stellenanzeigen	Anzahl Token	Durchschnittl. Anzahl Token pro Stellenanzeige
1980 - 1989	5’274	388’999	73.76
2014	7’695	1’997’540	259.59
2015	14’845	3’937’540	265.24
2016	14’853	3’967’973	267.15
2017	16’196	4’431’524	273.62
2018	15’209	4’253’816	279.69
2019	15’455	4’548’864	294.33
2020	15’853	4’743’400	299.21
2021	137’378	38’263’514	278.53
2022	191’183	57’420’144	300.34
2014 - 2022	428’667	123’564’315	288.25

Tabelle 4.3: Verteilung der Stellenanzeigen und Token im verwendeten SMM-Korpus-Auszug

4.3 Passim

Passim[14] ist ein Open-Source Software-Paket zur Analyse von Textwiederverwendung in sehr grossen Textsammlungen. Entwickelt und vorgestellt von David A. Smith (D. A. Smith et al., 2013; D. A. Smith et al., 2014), wird es kontinuierlich weitergepflegt und eingesetzt (Cordell et al., 2023).

Romanello und Hengchen (2021) vermerken in ihrem Tutorial, dass Passim im Hintergrund *Apache Spark*[15] nutzt, ein in Java geschriebenes Cluster-Computing-Framework. Durch das Einbinden von Apache Spark lassen sich gewisse Prozesse von Passim parallel ausführen. Die Aufteilung in Subprozesse ermöglicht Passim eine effizientere Verarbeitung dieser großen Datenmengen.

Passim ist wie Apache Spark in Scala[16] implementiert, einer rein objektorientierten und funktionalen Programmiersprache, die Datenverarbeitung über verteilte Datensätze optimal unterstützt.

4.3.1 Algorithmus

Der Ablauf folgt grundsätzlich drei Teilschritten und ist in D. A. Smith et al. (2013), D. A. Smith et al. (2014) und D. Smith et al. (2015) eingehender erklärt.

[14] https://github.com/dasmiq/passim (Letzter Besuch: 13. Juni 2023).
[15] https://spark.apache.org/ (Letzter Besuch: 13. Juni 2023).
[16] https://www.scala-lang.org/ (Letzter Besuch: 13. Juni 2023).

Im *ersten Schritt* werden mögliche Dokumentenpaare gesucht, die voraussichtlich wiederverwendete Textpassagen enthalten. Mit dem Ziel eine Vorauswahl zu treffen, um nachfolgende Schritte zu weniger Rechenintensiven abzuändern.

Zuerst kommt das „Shingling“-Verfahren (Broder et al., 1997) zur Anwendung, dabei repräsentiert ein ungeordneter Index von n-Grammen (Subsequenz von n Token, engl. shingles; Henzinger, 2006, S. 285), jeweils ein Dokument.

Anstelle von aufeinanderfolgenden Token-Sequenzen existiert auch ein Verfahren mit „k-Skip-n-Grammen“. Passim adaptiert dieses Verfahren, das mit k Token regelmässig unterbrochenen Token-Sequenzen arbeitet (Guthrie et al., 2006).

Der fertige, invertierte Dokumentenindex beinhaltet für jedes n-Gramm-Merkmal einen Eintrag zusammen mit einem referenzierenden Tupel aus Dokumentenkennziffer und Position innerhalb des Dokuments.

Angesichts der grossen Datenmengen bringt dabei das anschliessende Herausfiltern von nur in einem Dokument vorkommenden und damit uninteressanten n-Grammen nach dem Zwei-Phasen-Ansatz von Huston et al. (2011) eine ressourcenschonende Speicherplatzersparnis von mehr als 50%.

Es bleibt noch jene Dokumentenpaare weiterzuverarbeiten, die eine vordefinierte Mindestanzahl gemeinsamer n-Gramme aufweisen und nicht beide in einer ebenfalls über Parameter vordefinierbaren Serie vorkommen. Auch kann über die Passim-Konfiguration die Filterung mittels Dokumentenfrequenz gesteuert werden. Die Voreinstellung ist ausgerichtet auf eine deutliche Verbesserung der Performanz (D. A. Smith et al., 2013, S. 89), ohne wesentlichen Informationsverlust (Elsayed et al., 2008, S. 267 f.).

Der *zweite Schritt* richtet die ungeordneten, gepaarten n-Gramme zu kompakten Textpassagenpaaren aus, so wie sie in den Dokumenten vorkommen. Angewendet wird eine angepasste Form des Smith-Waterman Algorithmus. T. F. Smith und Waterman (1981, S. 195) formulierten das Optimierungsproblem des paarweisen Sequenzalignments wie folgt: „ein Segmentpaar finden, je verteilt auf zwei lange Sequenzen, so dass kein anderes mögliches Segmentpaar eine grössere Ähnlichkeit aufweist“. Die Anpassung für Passim bewirkt eine weniger strikte Auslegung, da wünschenswert ist, mehr als ein Paar zu finden.

Passim nutzt dazu eine bekannte Technik aus der Bioinformatik, die nach dem Prinzip der dynamischen Programmierung (Gusfield, 1997, S. 215 ff.) ein lokales Dokumenten-Alignment mit affinen Gap-Kosten durchführt.

Lokales Alignment bedeutet hier, dass die passenden Textpassagen nicht über die ganze Sequenzlänge zusammenhängend verlaufen müssen, sondern Auslassungen erlauben. Jede neue Auslassung (engl. gap) wird mit einem hohen negativen Wert, jede erweiternde Auslassung mit einem niedrigen negativen Wert belegt. Dagegen wird jede Übereinstimmung mit einem hohen positiven Wert und jede Nichtübereinstimmung mit einem niedrigen positiven Wert belegt.

So können z. B. OCR-Fehler ausgeglichen werden, sowie die Trefferquote (engl. recall) und

die Genauigkeit (engl. precision) der gefundenen Dokumentenpaare optimiert werden (D. A. Smith et al., 2013, S. 89).

Im letzten *dritten Schritt* müssen passende Cluster aus den erhaltenen alignierten Textpassagenpaaren gebildet werden (D. A. Smith et al., 2013, S. 90). Die nach Textlänge sortierten Passagenpaare durchlaufen ein hierarchisch-agglomeratives Clustering-Verfahren (Weiteres dazu siehe Abschnitt 3.4). Nach dem Bottom-up Prinzip sind zu Beginn alle Listeneinträge eigene Cluster.

Dann fusionieren jeweils Cluster mit Dokumenten, die mindestens eine 80% Überlappung zueinander aufweisen (Single-Linkage-Methode). Sind einmal zwei Cluster zusammengefügt, ist dies unumkehrbar.

Als Ergebnis gibt Passim eine mengensortierte Reihe von wiederum indexierten Clustern zurück. Jeder Cluster enthält die alignierten Textpassagen mit den Rückreferenzen zum Quelltextabschnitt der Übereinstimmung und die Dokumentenkennziffer.

4.3.2 Parameter und Daten-Format

Eine Übersicht[17] der veränderbaren Parameter für Passim gibt Tabelle 4.4. Angesichts der sehr knapp gehaltenen Dokumentation von Passim stellt das Finden einer passenden Konfiguration zur jeweiligen Fragestellung ein nicht-triviales Problem dar.

g	Voreinstellung= 100	Minimale Grösse der Lücke, die Passagen separiert
m	Voreinstellung= 5	Minimale Anzahl n-Gramm-Übereinstimmungen zwischen Dokumenten
M	Voreinstellung= 0.3	Maximale Längendivergenz für eine Zusammenführung von Bereichen
n	Voreinstellung= 5	Index n-Gramm-Funktionen
o	Voreinstellung= 0.8	Minimale relative Überlappung, um Passagen zusammenzuführen
u	Voreinstellung= 100	Obergrenze für die Dokumentenhäufigkeit
w	Voreinstellung= 2	Minimale durchschnittliche Wortlänge für den Abgleich

Tabelle 4.4: Auszug der optional veränderbaren Passim-Parameter

Grundsätzlich besteht das Eingabeformat für Passim aus einem Datensatz von „Dokumenten“, die gegebenenfalls unterteilt sind in „Serien“. Zu beachten ist, dass Textwiederverwendungen innerhalb einer Serie nicht ausgegeben werden.

[17] https://programminghistorian.org/en/lessons/detecting-text-reuse-with-passim#running-passim (Letzter Besuch: 8. Mai 2023).

Ein Dokument muss wenigstens aus einer eindeutigen Dokumentenkennziffer und dem Text als eine Zeichenkette (engl. single string) bestehen. Andere Datenfelder können auch mitgegeben werden. Sie werden von Passim ignoriert.

Das normale Eingabeformat ist eine Datei oder eine Reihe von Dateien im JavaScript-Object-Notation-Lines (JSONL)-Format. Die Ausgabe erfolgt im JavaScript-Object-Notation (JSON)-Format. Passim handhabt auch komprimierte Dateien.

Über die Möglichkeit mit Hilfe von Serien die Dokumente zu gruppieren, kann Passim entweder im „Query“-Modus oder im „All-Pairs“-Modus laufen. Hat man z. B. einen Referenztext kann im Query-Modus dieser mit einer eigenen Serie versehen und gegen alle anderen Texte mit einer gemeinsamen Serie verglichen werden. Im All-Pairs-Modus hingegen werden alle Texte mit allen anderen Texten gleichermassen abgeglichen.

4.3.3 Limitationen

Obwohl Teilprozesse über Apache Spark parallelisierbar sind, ist Passim speicherintensiv, bleibt aber ausreichend performant.

Die Längendivergenz der zu untersuchenden Texte spielt eine zentrale Rolle, da nicht gut steuerbar ist, wie die alignierten Textpassagen schlussendlich aussehen sollen. Die ursprüngliche Annahme, dass Passim direkt Ergebnisse auf Paragraphen-, Satz- oder Phrasenebene ausgibt, ist zu bezweifeln, dadurch bekommen die Vorverarbeitungsschritte und der Umfang der Nachbearbeitung deutlich mehr an Gewicht.

Passim definiert seine n-Gramme auf Tokenebene und ist robust gegen einzelne Zeichenfehler. Damit ist eine Anreicherung der Daten mit linguistischen Annotationen, z. B. POS-Tags gut vorstellbar. Salmi et al. (2021) stellen aber fest, dass ihre sehr schlechte Worterkennungsrate aufgrund einer extrem hohen OCR-Fehlerrate Passim an seine Grenzen brachte und dazu führte, sich gegen eine Verwendung zu entscheiden. Es stellt sich die Frage, in welcher Form die POS-Tags angefügt werden können.

Auch liegt der Single-Linkage-Methode die Schwäche zugrunde, zur Ähnlichkeitsbestimmung nur auf ein einziges Dokumentenpaar zurückzugreifen. Das wiederholte Zusammenfassen von diversen Clustern über nur ein gemeinsames Dokumentenpaar kann schliesslich zum „Chaining“-Effekt und zu grossen, übermässig heterogenen Clustern führen (Aggarwal, 2018, S. 93). Hier muss sich zeigen, wie mit solchen Clustern in der Folge umgegangen werden soll.

4.4 Cutter

Für die Satzsegmentierung kommt *Cutter*[18] zum Einsatz. Cutter ist ein regelbasierter Tokenisierer (Graën et al., 2018). Unter Tokenisierung versteht man das Segmentieren von Fliesstext in Grundeinheiten wie Wörter (Bussmann, 2008, S. 794) beziehungsweise Token, die alle Grundeinheiten (z. B. Satzzeichen) einschliessen. Es handelt sich dabei um einen sprachabhängigen Vorgang.

Wie Graën et al. (2018, S. 79) anmerken, ersetzen heute vorwiegend Machine-Learning-Ansätze ein regelbasiertes Vorgehen. Machine-Learning bedingt die Möglichkeit den Algorithmus mit genügend konsistenten Daten auf eine spezifische Textsorte zu trainieren. Das SMM-Korpus weisst aber einige Besonderheiten auf. Stellenanzeigen bilden nicht nur eine unruhige Textsorte. Auch das Überführen von Webseiteninhalten ins Korpus bereitet noch Schwierigkeiten. Beides resultiert in unregelmässigen Daten.

Erschwerend kommt hinzu, dass mit dem Tokenisierer von spaCy[19] alle Satzzeichen die selben Tags erhalten, das POS-Tag „$.“ und das UPOS-Tag „PUNCT“. Das Datenfeld „dep“ mit „punct“ enthält ebenfalls keine Informationen zur Unterscheidung von satzinterner und satzfinaler Zeichensetzung, um eine Satzsegmentierung direkt vorzunehmen.

Cutter schlägt hier eine Brücke. Diewald et al. (2022) zeigen in ihrer Evaluation auf, dass Cutter im Gegensatz zu spaCy-Tools (Tokenisierer und Satzsegmentierer) ungeeignet für sehr grosse Datenmengen ist. Allerdings tokenisiert und satzsegmentiert Cutter in einem Schritt und erreichte in allen Kategorien eine bessere Genauigkeit.

4.4.1 Algorithmus

Cutter arbeitet rekursiv und nach dem Organisationsprinzip LIFO (Last In First Out) mit sprachspezifischen und sprachunabhängigen Token-identifizierenden Regeln (Graën et al., 2018, S. 75). Für Stellenanzeigen ist dies ein Vorteil. Die Regeln können sehr gezielt auf die typographische und orthographische Vielfalt und die Eigenheiten dieser Textsorte abgestimmt werden. Dieses Vorgehen erweist sich bei häufig vorkommenden, komplexen Token (z. B. E-Mail-Adressen) und Abkürzungen als sehr robust.

Cutter benutzt zwei separate Listen, ein Abkürzungsverzeichnis und eine Liste für untypische Token am Satzbeginn. In dieser Arbeit liegt der Fokus auf den Abkürzungen. Die manuelle Durchsicht der Daten, um solche Listen zu erstellen, ist entsprechend aufwendig.

[18] https://pub.cl.uzh.ch/wiki/public/cutter/start (Letzter Besuch: 23. Mai 2023).
[19] https://spacy.io/api/tokenizer (Letzter Besuch: 16. Juni 2023).

4.4.2 Parameter und Daten-Format

Neben den beiden Listen für Abkürzungen und alternative Satzanfänge, kann eine der integrierten Zielsprachen ausgewählt werden. Als Eingabeformat nimmt Cutter eine Zeichenkette. Die Ausgabe erfolgt in 5-Tupeln. Jedes Tupel enthält das Token selbst, ein cutter-eigenes POS-Tag, Ziffern für die Baumstruktur und je eine Ziffer für die Start- und Endposition in der Eingabe-Zeichenkette. Die Token werden nach den Regeln erkannt und zu Baumstrukturen gruppiert. Die Satzsegmentierung resultiert direkt daraus und durch das satzfinale POS-Tag „+EOS“ markiert.

Zur Verdeutlichung eine Beispielausgabe von Cutter anhand des zu Beginn der Arbeit aufgeführten Stellenanzeigentextes:

Token	**Tag**	**Baum**	**Start**	**Ende**
('Gesucht',	'deRtkA',	2	, 0	, 7)
('ein',	'deRtkA',	3	, 8	, 11)
('Knabe',	'deRtkA',	4	, 12	, 17)
('als',	'deRtkA',	5	, 18	, 21)
('Ausläufer',	'deRtkA',	6	, 22	, 31)
('welcher',	'deRtkA',	7	, 32	, 39)
('mit',	'deRtkA',	8	, 40	, 43)
('Ostern',	'deRtkA',	9	, 44	, 50)
('von',	'deRtkA',	10	, 51	, 54)
('der',	'deRtkA',	11	, 55	, 58)
('Schule',	'deRtkA',	12	, 59	, 65)
('entlassen',	'deRtkA',	13	, 66	, 75)
('wird',	'deRtkB',	14	, 76	, 80)
('.',	'+dot',	1	, 80	, 81)
(",	'+EOS3',	1	, 81	, 81)

4.4.3 Limitationen

Wie bereits erwähnt, ist Cutter wenig performant (Diewald et al., 2022, S. 215 f.). Die Laufzeit hängt sehr stark von der zu verarbeitenden Datenmenge ab und kann unbrauchbar lang werden. Der rekursive Ansatz benötigt zudem viel Speicherplatz. Alle regel-basierten Systeme gehen schlechter mit unvorhergesehenen Phänomenen um als Machine-Learning-basierte, dafür sind sie durch Regelanpassungen besser debugbar.

5 Experimente und Ergebnisse

Der Aufbau der Experimente sah ein stufenweises Setup vor. Zu Beginn sollten die Möglichkeiten von Passim ausgelotet werden. Erstes Ziel dabei stellte die Suche nach einer Einstellung für eine Textsegmentierung und den Erhalt kurzer, sich wiederholender Textpassagen dar. Ebenso von Interesse war die Passim-Konfiguration für eine passende Deduplikation im Vorverarbeitungsprozess. Vorrangig stellte sich die Frage: „Wieviel Vorverarbeitung braucht Passim?“.

5.1 Experimentelle Phase I

5.1.1 Implementierung von Konverter I und Datenaufbereitung

In einem ersten Schritt wurde eine Konvertierung benötigt, um die SMM-Datensätze in das passim-konforme Datenformat zu überführen.

Das Ausgangsformat JSONL konnte übernommen werden. Daraus ergab sich für Passim, dass ein Dokument einer Stellenanzeige entspricht.

Alle Stellenanzeigen haben eine eindeutige Identifikationskennung, die ebenfalls übernommen wurde. Die Einteilung in Serien hatte in diesem Stadium noch keinen Einfluss. Der angewandte Passim-Modus „All-Pairs“ verglich alle Stellenanzeigen miteinander. Der eigentliche Text wurde aus dem Datenfeld „tokens“ extrahiert.

Das folgende Beispiel illustriert die resultierende Datenstruktur einer Stellenanzeige mittels Konverter I:

```
{
”id”=”sjmm_large-8201541791002034”,
”text”=”3045 Meikirch BE Frühzustellung Werktag Auto 05.00 - 06.30 sofort jetzt bewerben”
}
```

Die ersten Passim-Läufe verwendeten die Jahrgänge 1980 - 1989 und 2015 als kleinere Testdaten-Auswahl (siehe Tabelle 4.3). Verschiedene Möglichkeiten (Tabelle 5.1) testeten dabei auch das Einfügen von POS-Tags und ihren Einfluss auf Passim. Getestet wurden sowohl angehängte ganze Tags mit und ohne Unterstrich (TAG/_TAG) als auch einzelne Zeichen mit und ohne Unterstrich (T/_T).

{
”id”=”sjmm_large-8201541791002034”,
”text”=”3045_NUM Meikirch_PROPN BE_PROPN Frühzustellung_NOUN Werktag_NOUN Auto_NOUN 05.00_NUM -_PUNCT 06.30_NUM sofort_ADV jetzt_ADV bewerben_VERB”
}

{
”id”=”sjmm_large-8201541791002034”,
”text”=”3045NUM MeikirchPROPN BEPROPN FrühzustellungNOUN WerktagNOUN AutoNOUN 05.00NUM -PUNCT 06.30NUM sofortADV jetztADV bewerbenVERB”
}

{
”id”=”sjmm_large-8201541791002034”,
”text”=”3045_M Meikirch_N BE_N Frühzustellung_N Werktag_N Auto_N 05.00_M -_T 06.30_M sofort_V jetzt_V bewerben_B”
}

{
”id”=”sjmm_large-8201541791002034”,
”text”=”3045M MeikirchN BEN FrühzustellungN WerktagN AutoN 05.00M -T 06.30M sofortV jetztV bewerbenB”
}

Tabelle 5.1: Getestete Varianten für das Anfügen der POS-Tags

5.1.2 Parameter-Auswahl für Passim

Eine Übersicht der getesteten Passim-Parameter[20] ist in Tabelle 5.2 aufgeführt. Aufgrund der mangelhaften Dokumentation von Passim liefen umfangreiche Testreihen.

Jeder Parameter lief einzeln variiert und in verschiedenen Kombinationen einmal durch Passim. Die Variationen der Parameter bekamen relativ zu den Grundeinstellungen konträre Werte.

[20] https://programminghistorian.org/en/lessons/detecting-text-reuse-with-passim (Letzter Besuch: 07. Juni 2023).

a	Voreinstellung= 20	Minimale Länge der Alignierung
c	Voreinstellung= 0	Größe des Kontexts für alignierte Passagen
g	Voreinstellung= 100	Minimale Grösse der Lücke, die Passagen separiert
m	Voreinstellung= 5	Minimale Anzahl n-Gramm-Übereinstimmungen zwischen Dokumenten
M	Voreinstellung= 0.3	Maximale Längendivergenz für eine Zusammenführung von Bereichen
n	Voreinstellung= 5	Index n-Gramm-Funktionen
o	Voreinstellung= 0.8	Minimale relative Überlappung, um Passagen zusammenzuführen
w	Voreinstellung= 2	Minimale durchschnittliche Wortlänge für den Abgleich

Tabelle 5.2: Getestete Passim-Parameter

5.1.3 Zwischenergebnis I

Die ersten Experimente ergaben, dass eine zielgerichtete Vorverarbeitung nötig ist. Es konnte keine genügende Textsegmentierung für die hier zu untersuchenden, formelhaften und typischen Textbausteine in Stellenanzeigen direkt über Passim erreicht werden. Zudem sind die entstehenden Cluster inhomogen in Textlänge und Vokabular.

Die Annahme musste verworfen werden, dass in Passim direkt auf Paragraphen-, Satz- oder Phrasen-Ebene segmentiert werden kann. Passim ist in diesem Kontext unzureichend konfigurierbar. Dennoch liessen sich geeignete Parameter für eine Deduplikation ableiten (Weiteres dazu siehe Abschnitt 5.2.1).

In Tabellen 5.3 und 5.4 sind die Ergebnisse einiger Testläufe mit der oben erwähnten Datenstruktur anhand von drei Kenngrössen aufgeführt.

Die Anzahl Treffer geben einen Eindruck, wie viele gleiche Muster erkannt werden konnten. Zu den Clustern sind die gesamte Anzahl der trefferreichsten Cluster und die durchschnittliche Anzahl Treffer pro Cluster angegeben. Sie ermöglichen die Verteilung der Treffer über alle Cluster darzustellen. Die Spannweite lässt Aussagen über die resultierende Textsegmentierung zu.

Bereits die Testläufe zu den einzelnen Parametern zeigen in Tabelle 5.3 wie innerhalb der Cluster keine gleichbleibende Textlänge entstand. Die Spannweite liess sich durch das Variieren von Textlücke (g) oder n-Gramm-Index (n) verringern. Alle anderen Parameter hatten jedoch einen marginalen Einfluss.

Die Erkennung von wiederverwendeten Textpassagen wurde quantitativ v. a. durch die minimale Alignierungslänge (a), die Anzahl der n-Gramm-Übereinstimmungen (m) und dem n-Gramm-Index (n) beeinflusst. Das Variieren der Grösse des Kontexts der alignierten Passagen

(c) und der Textlücke (g) hatten allerdings hier keinen Effekt. Alignierungslänge (a), Textlückengrösse (g), Anzahl der n-Gramm-Übereinstimmungen (m) und n-Gramm-Index (n) veränderten die Trefferverteilung über alle Cluster, der Kontext der alignierten Passagen (c) hatte hingegen keinen Einfluss auf diese.

Die maximale Trefferzahl pro Cluster senkte sich am stärksten über die Anzahl der n-Gramm-Übereinstimmungen (m) und dem n-Gramm-Index (n). Im Durchschnitt erreichte die Default-Einstellung von Passim rund sechs Stellenanzeigen pro Cluster. Dieser Wert schwankte v. a. wenn die Anzahl der n-Gramm-Übereinstimmungen (m) und der n-Gramm-Index (n) drastisch vergrössert wurden. Dann allerdings sank der Durchschnitt ab.

Der Einfluss von Alignierungslänge (a) und Kontext der alignierten Passagen (c) erscheinen vernachlässigbar. Anzahl der n-Gramm-Übereinstimmungen (m), der n-Gramm-Index (n) und die Textlückengrösse (g) hingegen wirkten sich auf die Schlüsselwerte aus.

Längendivergenz für eine Zusammenführung von Bereichen (M) und die relative Überlappung, um Passagen zusammenzuführen (o) hatten bezüglich der Default-Einstellung unterschiedliche Effekte.

Während bei der Variation der Längendivergenz (M) geringe Effekte auftraten, erreichte dies bei der relativen Überlappung (o) deutlichere Abweichungen.

Die Treffer- und Clusteranzahl wurden kleiner, aber die maximale und durchschnittliche Clustergrösse sowie die Spannweite vergrösserten sich.

Bei einer Steigerung der durchschnittlichen Wortlänge für den Abgleich (w) verringerte sich die Trefferzahl auf null. Bei einem Wert von 20 n-Grammen für *w* gab Passim bereits keine Treffer mehr zurück.

Die Testläufe mit Parameter-Kombinationen aus Tabelle 5.4 ergänzen das Bild. Die Spannweite blieb weiter ein Problem. Grösstmöglichen Einfluss war über die Textlückengrösse (g) zu erreichen. In Kombination mit der Anzahl der n-Gramm-Übereinstimmungen (m), dem n-Gramm-Index (n), der Längendivergenz (M) und der relativen Überlappung (o) lassen sich ebenfalls Trefferzahl und Clusterausprägung kontrollieren. Die Alignierungslänge (a) führte zur Erhöhung der Trefferanzahl, jedoch auf Kosten steigender Clustergrösse.

D. A. Smith et al. (2013, S. 88 f.) konfigurierten das Prinzip der k-Skip-n-Gramme (Weiteres dazu siehe Abschnitt 4.3.1) in Passim mit *n* (Anzahl eingeschlossener Token), *g* (Mindestabstand zwischen Token) und *w* (Höchste Anzahl Token, die die k-Skip-n-Gramme überspannen sollen).

Die Angaben zu *w* aus Abschnitt 5.1.2 sind im Vergleich dazu aber uneindeutig. Allerdings ergab die angegebene Passim-Konfiguration (n=5, g=1, w=5) für eine sehr kleine Spannweite

					Cluster				**Spannweite**	
	Treffer		Anzahl		Max. Grösse		∅ Grösse		∅ [Token]	
Lauf	1980 - 1989	2015	1980 - 1989	2015	1980 - 1989	2015	1980 - 1989	2015	1980 - 1989	2015
Default	841	34’673	335	5’795	76	167	2.51	5.98	8.94	36.60
a: 5	1709	48’964	626	8’228	76	733	2.73	5.94	7.14	28.26
a: 100	124	13’395	42	2’942	33	65	2.95	4.55	7.41	35.76
c: 5	841	34’673	335	5’795	76	167	2.51	5.98	8.94	36.60
c: 20	841	34’673	335	5’795	76	167	2.51	5.98	8.94	36.60
g: 5	663	34’966	263	8’018	54	70	2.52	4.35	1.60	6.79
g: 20	744	35’508	286	6’846	71	126	2.60	5.17	3.26	14.28
m: 12	688	32’745	281	5’478	75	101	2.44	5.97	5.79	36.31
m: 20	577	30’703	220	5’191	72	75	2.62	5.91	3.95	35.95
m: 100	62	11’529	29	2’900	4	51	2.14	3.98	1.04	23.35
M: 0.8	823	32’329	326	4’993	76	168	2.52	6.47	8.87	36.11
n: 3	1790	44’493	681	7’994	78	238	2.63	5.56	20.70	44.20
n: 12	572	27’134	225	5’086	73	74	2.54	5.33	3.22	24.74
n: 20	526	25’372	199	5’046	52	73	2.64	5.03	2.82	20.84
n: 100	41	5’029	20	1’824	3	45	2.05	2.76	0.55	7.67
o: 0.3	814	20’326	320	2’787	78	716	2.54	6.83	9.83	62.43
w: 5	723	32’244	291	5’369	74	151	2.48	6	7.40	34.72

Tabelle 5.3: Ergebnisse aus den Testläufen: Variation eines Parameters

dennoch eine grosse Anzahl Cluster und Treffer.

Der Jaccard-Koeffizient (Weiteres dazu siehe Abschnitt 3.5) zeigt, wie sich die Ähnlichkeit innerhalb der Cluster mit den unterschiedlichen Passim-Konfigurationen ändert. Als additives Kriterium einbezogen, hilft dies die Homogenität innerhalb der Cluster zu verdeutlichen und wie in Wise (1993, S. 2) dient der kürzeste Text im Cluster als Referenzpunkt.

Abbildung 5.1 zeigt, wie steigende Werte für die Textlückengrösse (g) die clusterinterne Homogenität verschlechterten, während n-Gramm-Übereinstimmung (m) und n-Gramm-Index (n) bei Werterhöhung die Homogenität verbesserten. Die relative Überlappung (o) erhöhte die Ähnlichkeit der gefundenen Texte beträchtlich. Eine geringe, positive Änderung erzielte die Erhöhung der durchschnittlichen Wortlänge für den Abgleich (w).

			Cluster						Spannweite	
	Treffer		Anzahl		Max. Grösse		∅ Grösse		∅ [Token]	
Lauf	1980 - 1989	2015	1980 - 1989	2015	1980 - 1989	2015	1980 - 1989	2015	1980 - 1989	2015
Default	841	34’673	335	5’795	76	167	2.51	5.98	8.94	36.60
g5a5	2657	116’670	985	23’135	75	700	2.70	5	1.64	4.39
g5m20	632	34’830	248	7’990	54	70	2.55	4.35	1.67	6.79
g5m20w5	512	32’384	208	7’806	54	66	2.46	4.14	1.06	5.18
g5o0.3M0.8	646	28’083	257	6’263	54	82	2.51	4.39	1.76	16.38
g5o0.3M0.8a5	2435	86’555	919	17’459	75	1’192	2.64	4.73	2.19	8.51
g5o0.3M0.8w5	560	28’069	230	6’639	54	73	2.43	4.18	1.09	10.83
g1n5w5	419	24’447	166	6’902	54	48	2.52	3.54	0.52	1.66
g5n3m20	641	35’251	245	7’883	70	70	2.62	4.47	1.86	7.10
g5n3m20w5	533	31’214	201	7’477	68	70	2.65	4.17	1.39	5.12
g5n3m100	76	22’279	35	6’368	4	55	2.17	3.50	0.80	5.52
g5n3m100w5	42	17’751	21	5’794	2	63	2.00	3.06	0.57	3.20
g5n12m12	635	35’337	249	8’006	54	112	2.55	4.40	1.66	6.99
g5n12m12w5	617	35’038	242	8’159	54	112	2.55	4.28	1.50	6.14
g5n100m20	17	4’348	8	1’702	3	27	2.12	2.55	0.25	3.53
g5n100m20w5	17	4’336	8	1’696	3	27	2.12	2.56	0.25	3.56
g5n3m20o0.3M0.8	629	28’134	239	6’037	70	137	2.63	4.52	2.98	17.20
g5n3m100o0.3M0.8	72	18’063	33	5’110	4	81	2.15	3.45	3.49	14.84
g5n3m100o0.3M0.8w5	42	16’028	21	5’211	2	75	2.00	3.04	0.57	6.86
g5n12m12o0.3M0.8	618	28’033	243	6’177	54	113	2.54	4.42	1.84	17.03
g5n12m12o0.3M0.8w5	601	29’034	236	6’602	54	113	2.55	4.32	1.87	13.73
g5n100m20o0.3M0.8	17	3’885	8	1’506	3	27	2.12	2.57	0.25	12.91
n3m100	62	11’271	29	2’779	4	46	2.14	4.06	1.04	28.56
n3m100w5	20	6’806	10	2’034	2	45	2.00	3.35	1.10	21.46
n12m12	529	26’315	206	4’951	61	74	2.57	5.31	3.40	24.45
n12m12w5	496	25’252	191	4’789	61	73	2.60	5.27	3.45	23.98
n100m20	17	4’151	8	1’606	3	27	2.12	2.58	0.50	6.43
n3m100o0.3M0.8	62	8’120	29	2’049	4	63	2.14	3.88	1.03	40.10
n3m100o0.3M0.8w5	20	5’615	10	1’722	2	58	2.00	3.22	1.10	26.80
n12m12o0.3M0.8	506	16’990	203	2’616	62	113	2.48	6.06	3.55	55.72
n12m12o0.3M0.8w5	473	16’322	188	2’534	62	113	2.52	6.02	3.61	55.50
n100m20o0.3M0.8	17	3’836	8	1’480	3	27	2.12	2.59	0.50	12.32
o0.3M0.8	798	18’888	311	2’296	77	886	2.56	7.65	10.90	67.03
o0.3M0.8w5	700	18’296	280	2’324	75	499	2.50	7.43	8.14	66.08

Tabelle 5.4: Ergebnisse aus den Testläufen: Variation Parameter-Kombinationen

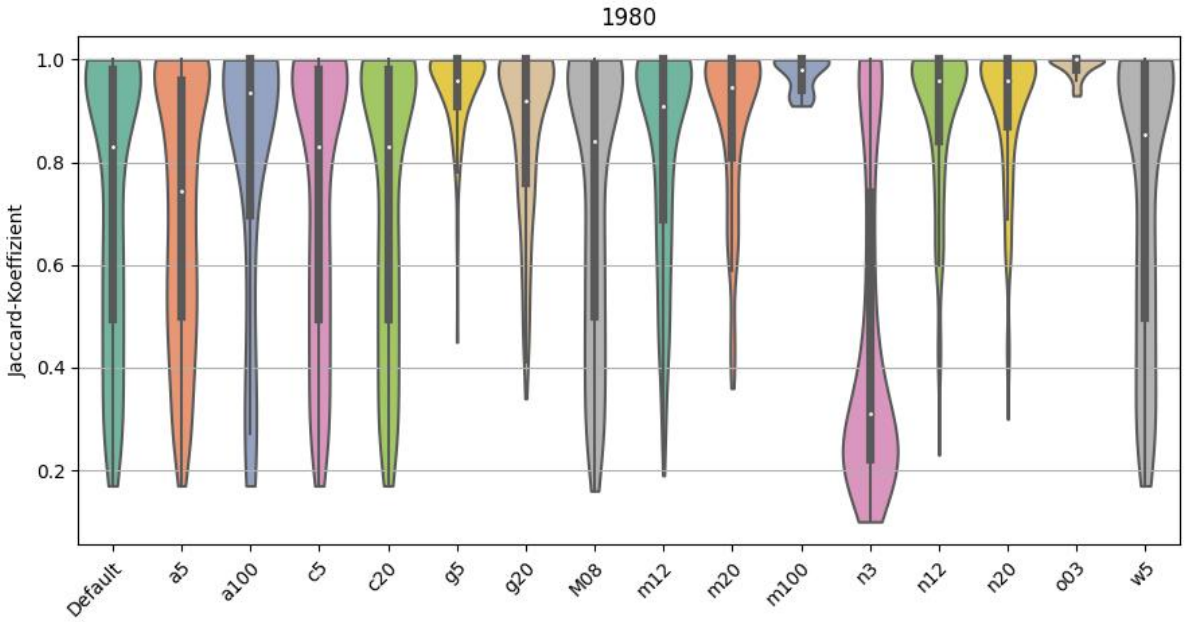

Abbildung 5.1: Jaccard-Koeffizient zu Einzel-Parameter Läufen (1980 - 1989)

Bei deutlich mehr Datenmaterial ergaben sich für den Jaccard-Koeffizienten ähnliche Tendenzen, aber mit einer deutlichen Verschiebung der mittleren Ähnlichkeit in Richtung mehr Heterogenität, wie in Abbildung 5.2 erkennbar ist.

Die veränderten Geigenplots in den Abbildungen 5.3 und 5.4 illustrieren die Notwendigkeit die Parameter in Kombination zu verändern, da so eine bessere Homogenität erreichbar war.

Während Abbildung 5.3 den positiven Einfluss von Textlückengrösse (g) und durchschnittlicher Wortlänge für den Abgleich (w) sowie die Kombination aus n-Gramm-Index (n) und n-Gramm-Übereinstimmung (m) demonstriert, so zeigt Abbildung 5.4, dass mehr Daten auch mehr Heterogenität in den einzelnen Clustern hervorbrachte.

Wenn zusätzlich die relative Überlappung (o) und die Längendivergenz (M) dazukamen, veränderte sich die Homogenität in den Clustern nur marginal (Abbildungen 5.5 und 5.6). Dabei ergaben wenig Daten mit kleinem n-Gramm-Index (n) und wachsender n-Gramm-Übereinstimmung (m) eine Verbesserung der Textähnlichkeit (Abbildung 5.5).

Mehr Daten wiederum (Abbildung 5.6) liessen zusammen mit konstanter Textlückengrösse (g), aber sich veränderndem n-Gramm-Index (n) und verändernder n-Gramm-Übereinstimmung (m) die Heterogenität in einzelnen Clustern grösser werden.

Abbildungen 5.1 bis 5.6 veranschaulichen, dass die Textlückengrösse (g), die n-Gramm-

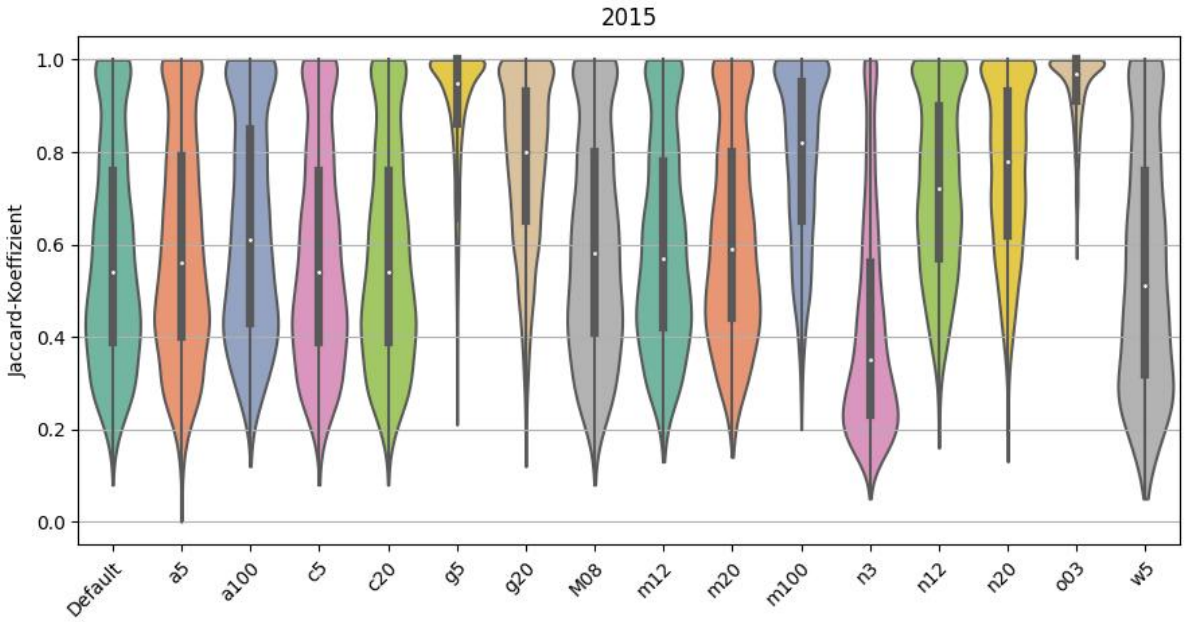

Abbildung 5.2: Jaccard-Koeffizient zu Einzel-Parameter Läufen (2015)

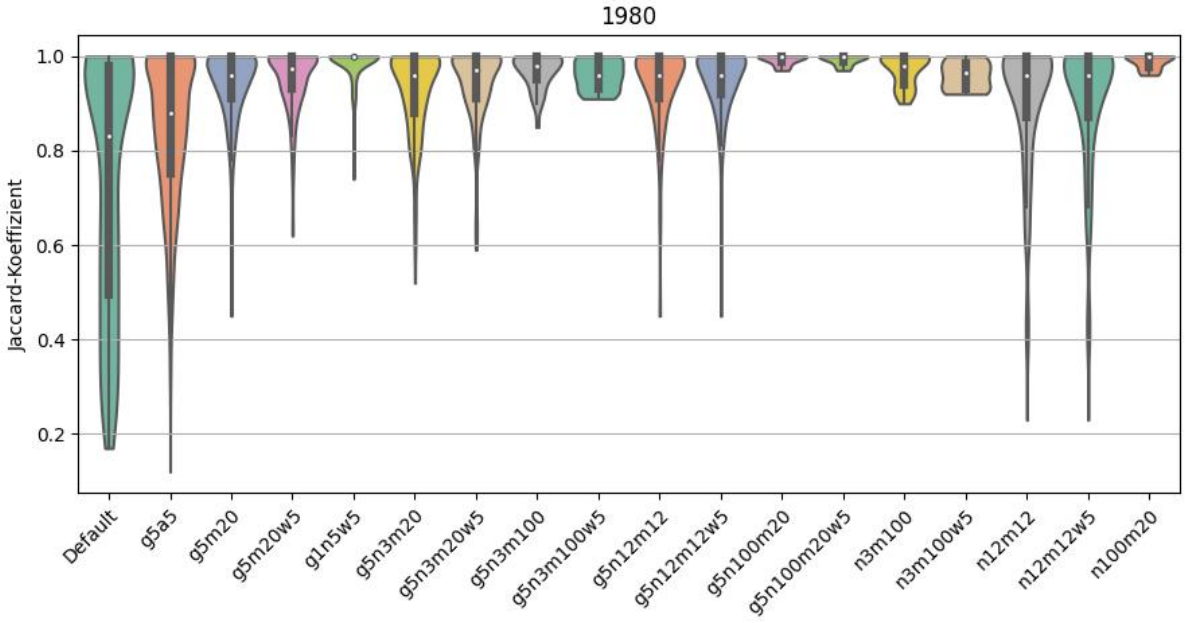

Abbildung 5.3: Jaccard-Koeffizient zu Multi-Parameter Läufen 1 (1980 - 1989)

Übereinstimmung (m) und der n-Gramm-Index (n) für die Textwiederverwendung geeignet sind. Mit Textlückengrösse (g) und der durchschnittlichen Wortlänge für den Abgleich (w) konnte eine

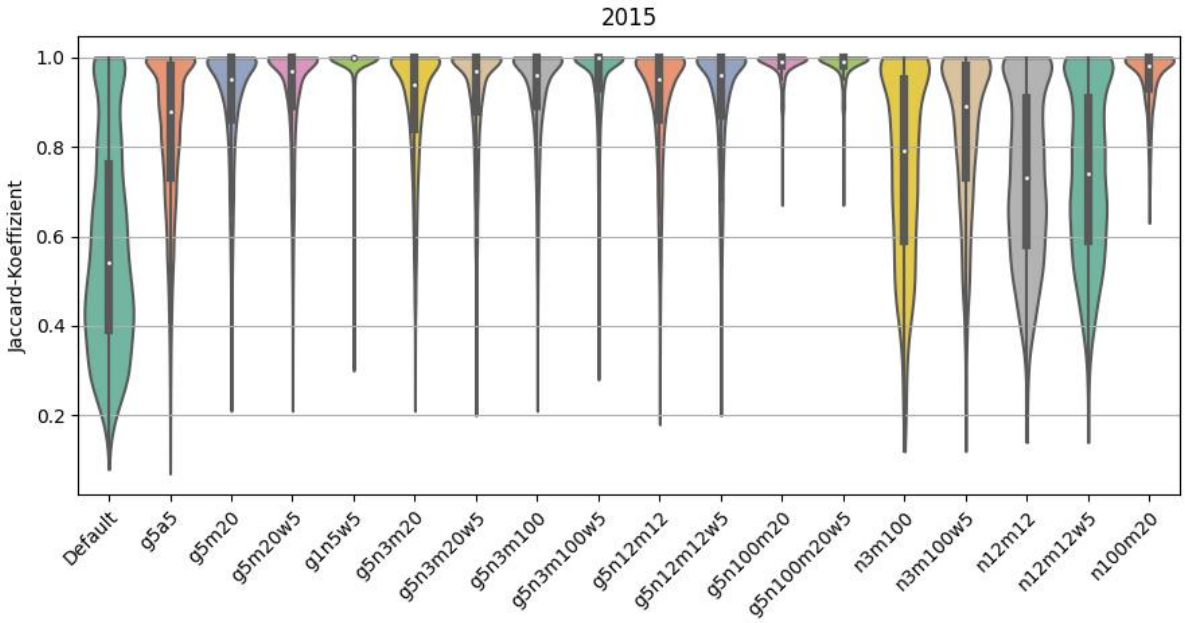

Abbildung 5.4: Jaccard-Koeffizient zu Multi-Parameter Läufen 1 (2015)

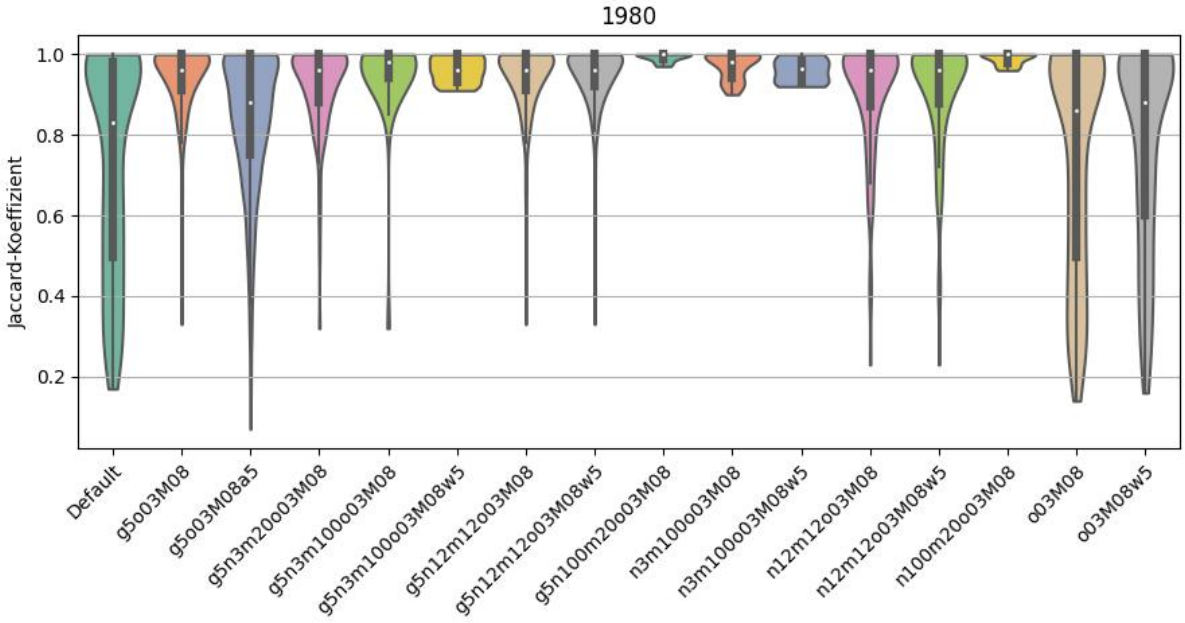

Abbildung 5.5: Jaccard-Koeffizient zu Multi-Parameter Läufen 2 (1980 - 1989)

begrenzte Textsegmentierung erreicht werden.

Die relative Überlappung (o) und die Längendivergenz (M) sollten nur bei der Deduplikation

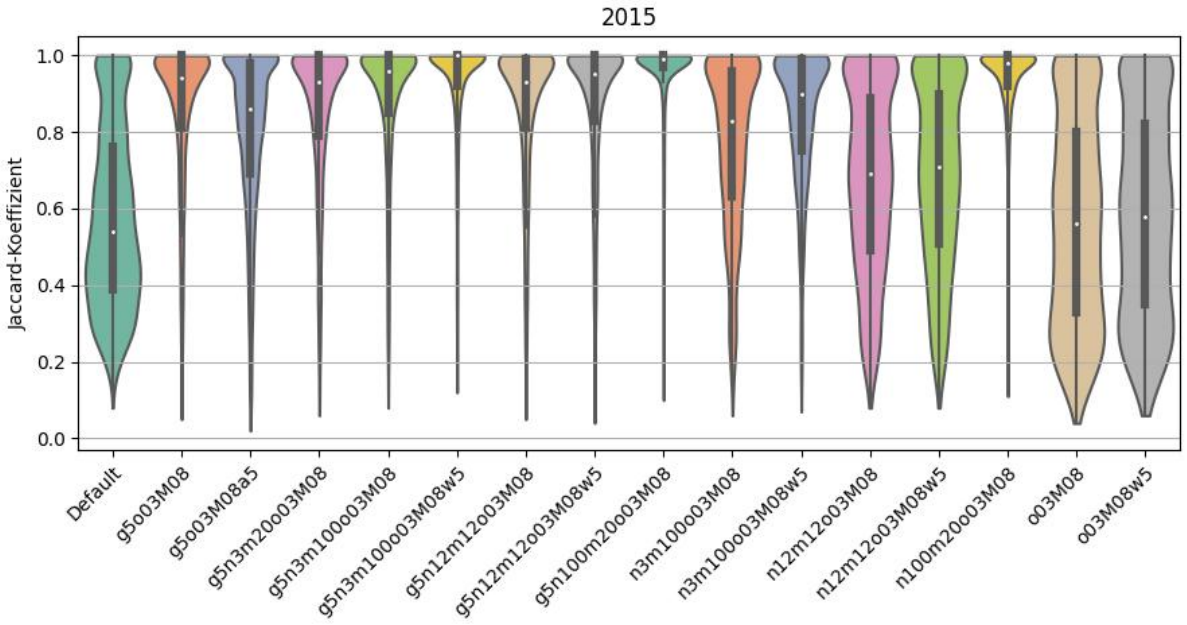

Abbildung 5.6: Jaccard-Koeffizient zu Multi-Parameter Läufen 2 (2015)

zusammen mit n-Gramm-Index (n) und n-Gramm-Übereinstimmung (m) verwendet werden.

Tabelle 5.5 zeigt die Testläufe zu den Jahrgängen 1980 bis 1989 mit POS-Tags in den vier Varianten. Auch hier erfasst die Übersichtstabelle die Trefferanzahl, daneben aber nur die Anzahl Cluster und Token, da dabei lediglich interessierte, wie die Treffer und ihre Verteilung zum jeweiligen Referenzwert variieren.

Der geklammerte Referenzwert in Tabelle 5.5 gibt den Wert zum jeweiligen Passim-Lauf ohne POS-Tag und für die gleiche Datenstruktur an. Werte aus Passim-Läufen mit POS-Tag sollten dem Referenzwert möglichst nahekommen, ansonsten musste von einer übermässigen Verzerrung der Ergebnisse ausgegangen werden.

Treffer- und Clusteranzahl waren mit POS-Tag ohne Unterstrich immer näher am Referenzwert und die Tokenzahl um ein Vielfaches erhöht. Der Unterstrich wurde somit in den Folgeexperimenten weggelassen. Die Mitnahme nur eines Einzelzeichens am Ende des Tokens reduzierte die entstehende Verzerrung nochmals. Passim zeigte hier wiederum seine Stärke bezüglich einer moderaten Fehlerrate.

Wir benötigen dementsprechend zwei Vorverarbeitungsschritte. Einmal eine vorgegebene Textsegmentierung und eine separate Deduplikation, weil Passim ungenügend auf eine gewünschte Ausgabe-Textlänge konfigurierbar ist.

Abhängig davon wie die Informationen zur Wortart im Text einfügt wurden, veränderte sich

	Treffer [10^3]				**Cluster** [10^3]				**Token** [10^3]			
Lauf	_TAG	TAG	_T	T	_TAG	TAG	_T	T	_TAG	TAG	_T	T
a: 5	14.6	2.6	18.9	2.6	3.9	0.9	5.3	0.9	534.4	123.6	505.5	100.7
		(1.7)				(0.6)				(82.3)		
c: 5	8.9	1.0	9.5	1.0	2.5	0.4	2.8	0.4	486.3	75.9	424.4	71.1
		(0.8)				(0.3)				(64.0)		
g: 5	1.4	0.7	1.4	0.7	0.6	0.3	0.6	0.3	43.3	34.0	42.1	33.8
		(0.7)				(0.3)				(33.8)		
m: 20	1.1	0.6	1.1	0.7	0.5	0.3	0.4	0.3	77.5	50.0	72.0	50.6
		(0.6)				(0.2)				(46.8)		
M: 0.8	6.6	1.0	6.7	1.0	1.4	0.4	1.6	0.4	378.5	74.5	320.0	69.7
		(0.8)				(0.3)				(62.8)		
n: 3	14.7	3.2	17.1	3.2	4.3	1.0	5.2	1.1	837.1	228.7	829.3	208.2
		(1.8)				(0.7)				(130.7)		
o: 0.3	5.8	1.0	6.5	1.0	1.2	0.4	1.6	0.4	340.3	74.2	312.3	69.5
		(0.8)				(0.3)				(62.9)		
w: 5	2.6	1.0	0.7	0.8	1.0	0.4	0.3	0.3	150.3	75.4	50.5	62.0
		(0.7)				(0.3)				(57.3)		

Tabelle 5.5: Ergebnisse aus den POS-Testläufen (1980 - 1989)

das Verhalten von Passim. Sie wurden bei der Deduplikation somit weggelassen. Daraus ergab sich aber auch die Notwendigkeit eines eigenen Konverters für die Vorverarbeitung.

Die Experimente wurden auf Satzebene wiederholt und in diesem Schritt zusätzlich noch ein Zonenfilter und die POS-Tags als Einzelzeichen am Tokenende hinzugefügt.

5.2 Experimentelle Phase II

Die beiden Vorverarbeitungsschritte der Deduplikation und Textsegmentierung auf Satzebene wurden in Teilschritten implementiert. Zur Satzsegmentierung kam der Tokenizer „Cutter“ (Weiteres dazu siehe Abschnitt 4.4) zum Einsatz.

Bei der Deduplikation dürfen nur identische Stellenanzeigen herausgefiltert werden. Einige wenige Duplikate sind tolerierbar (Schofield et al., 2017), aber zu viele Duplikate können auch bei klassischen Systemen eine Überanpassung (engl. *overfitting*) bewirken (Fröbe et al., 2020).

5.2.1 Deduplikation

Die Deduplikation musste die Schwierigkeit der unkontrollierbaren Textsegmentierung von Passim berücksichtigen. Da der Einfluss der Textlänge erheblich zum Clusterprofil beigetragen hatte, war die Frage, ob über das ganze Korpus oder über einzelne Jahrgänge zu deduplizieren ist.

5.2.1.1 Parameter-Auswahl

Aus den ersten Experimenten liessen sich die Passim-Parameter in Tabelle 5.6 ableiten. Zur Deduplikation eigneten sich die aufbereiteten Daten aus Experiment I, aber ohne die POS-Tags. Datenformat und Aufbau blieben gleich.

m	Voreinstellung= 5	Minimale Anzahl n-Gramm-Übereinstimmungen zwischen Dokumenten
M	Voreinstellung= 0.3	Maximale Längendivergenz für eine Zusammenführung von Bereichen
n	Voreinstellung= 5	Index n-Gramm-Funktionen
o	Voreinstellung= 0.8	Minimale relative Überlappung, um Passagen zusammenzuführen

Tabelle 5.6: Auswahl der Passim-Parameter zur Deduplikation

Als erste Möglichkeit wurde die Deduplikation über die ganze Zeitspanne getestet. Wie aus den vorangegangenen Experimenten ersichtlich, führte eine wachsende relative Überlappung (o) zusammen mit grossem n-Gramm-Index (n) und grosser n-Gramm-Übereinstimmung (m) zu einer höheren Homogenität in allen Clustern.

Die relative Überlappung (o) wurde mit dem Wert 0.999 gesetzt und trotz kleiner Effekte in den ersten Experimenten, mit dem Wert 0.1 die Längendivergenz (M). Dabei wurden für den n-Gramm-Index (n) und die n-Gramm-Übereinstimmung (m) bei der Konfiguration von Passim verschiedene Werte gewählt.

Eine Auswahl der Passim-Läufe ist in Tabelle 5.6 aufgeführt. Nach jedem Passim-Lauf geschah eine manuelle Datensichtung. Nur ganze und identische Texte sollten von Passim gefunden werden. Zudem war entscheidend, die Clustergrösse und Spannweite möglichst klein zu halten.

Abbildung 5.7 zeigt, dass sich die Spannweite nicht genügend über alle Daten glätten liess. Auch der jeweilige Blick in die Datensätze verdeutlichte die Problematik der stark unterschiedlichen Textlängen. Keine Passim-Konfiguration ermöglichte ausschliesslich ganze Texte zu finden.

		Cluster			Spannweite
Lauf	**Treffer**	Anzahl	Max. Grösse	∅ Grösse	∅ [Token]
o0999n200m200M01	19’515	7’207	72	2.71	4.59
o0999n200m600M01	390	169	12	2.31	3.98
o0999n200m700M01	144	68	4	2.12	4.09
o0999n250m250M01	7’395	2’808	45	2.63	4.15
o0999n300m300M01	3’120	1’190	45	2.62	4.01
o0999n400m400M01	438	193	12	2.27	3.65

Tabelle 5.7: Passim-Läufe für Deduplikation aller Jahrgänge

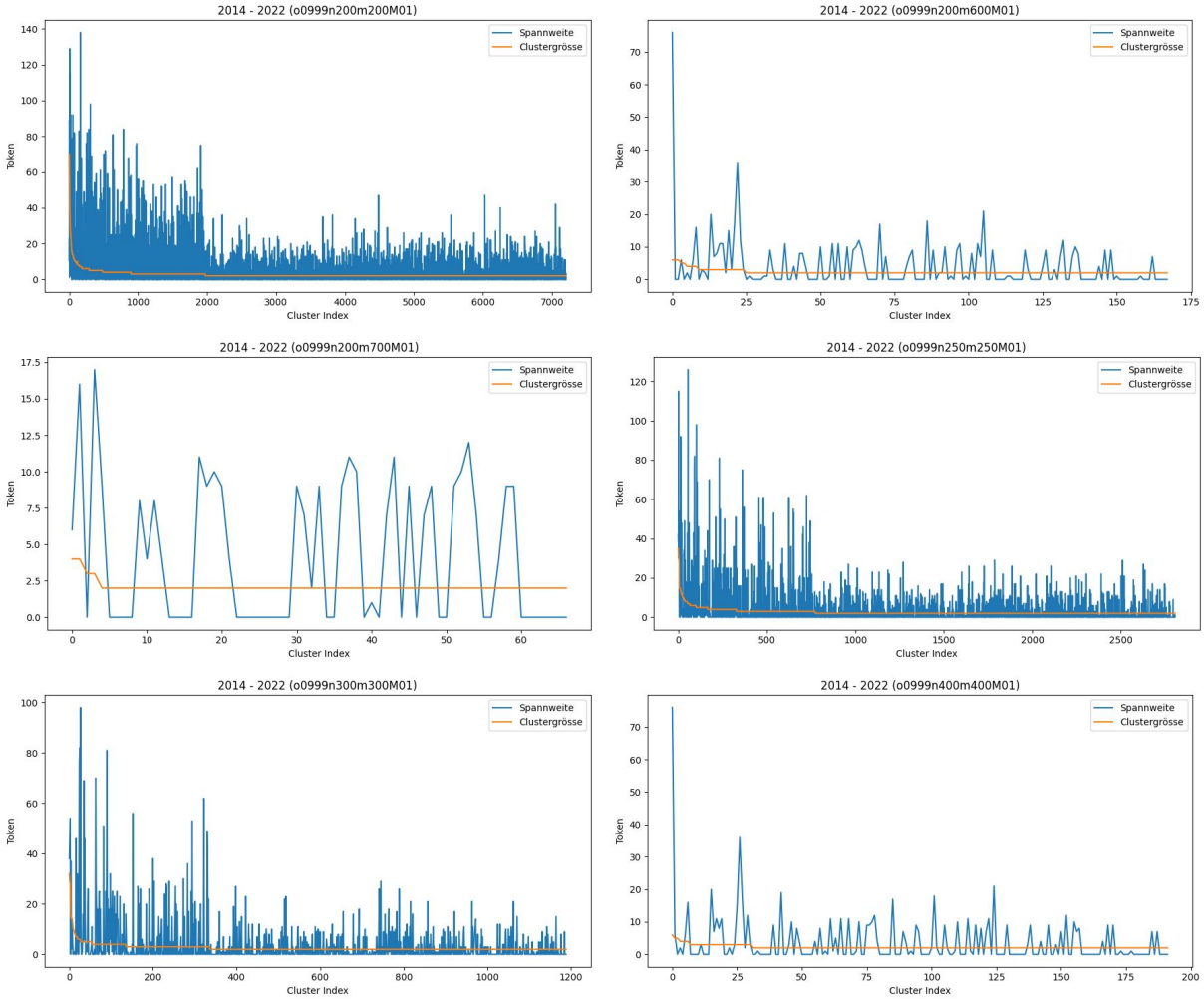

Abbildung 5.7: Passim-Läufe für Deduplikation aller Jahrgänge: Spannweite & Clustergrösse

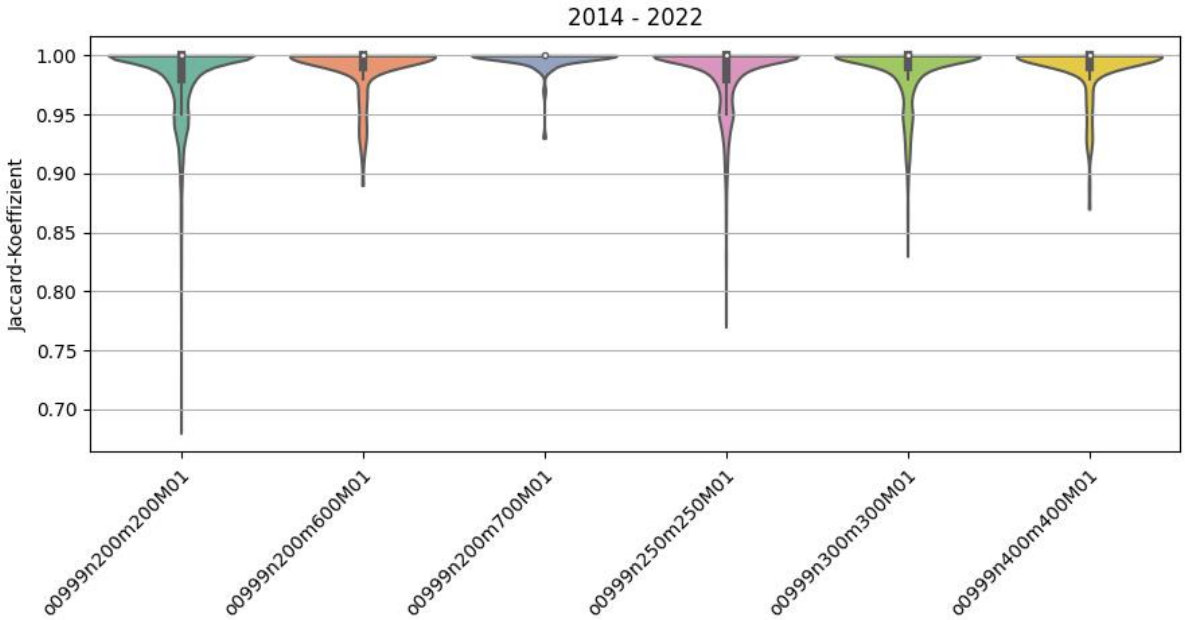

Abbildung 5.8: Jaccard-Koeffizient zu Passim-Läufen für Deduplikation aller Jahrgänge

Der Jaccard-Koeffizient war dennoch hoch, was eine dementsprechend hohe Homogenität in den Clustern zeigt (Abbildung 5.8). Die gewählten Parameter ergaben in dieser Kombination somit viele, ähnliche Textfragmente.

Folglich musste die jahrgangsweise Deduplikation durchgeführt werden. So konnte auch die Veröffentlichung von identischen Stellenanzeigen über grössere Zeitabstände in die Bewertung der Textwiederverwendung einfliessen.

Die jahrgangsweise Deduplikation geschah anhand derselben Parameter-Konfigurationen von Passim, mit einer relativen Überlappung (o) mit dem Wert 0.999 und einer Längendivergenz (M) mit dem Wert von 0.1. Bei n-Gramm-Index (n) und n-Gramm-Übereinstimmung (m) variierte die Konfiguration von Passim wiederum in Fünfziger Schritten. Auch hier wurde nach jedem Passim-Lauf eine manuelle Datensichtung durchgeführt.

Die Auswahl der Passim-Läufe zur jahrgangsweisen Deduplikation ist in Tabelle 5.8 aufgeführt. Die Clustergrösse und die Spannweite konnten in den ersten Jahrgängen klein gehalten werden (Abbildung 5.10).

Mit steigender Datenmenge wurde es zunehmend schwieriger, falsch-positive Treffer auszuschliessen, dies analog zur Problematik bei einer Deduplikation über alle Jahrgänge. Die Textlängen schwankten erneut zu stark. Im Jahr 2018 musste die n-Gramm-Übereinstimmung (m) beachtlich hoch gewählt werden. Ab 2019 konnte wieder auf eine reduzierte n-Gramm-Übereinstimmung (m) eingestellt werden. Erst nach 2021 wuchsen die Datenmengen sprung-

haft an und eine erneute Erhöhung der n-Gramm-Übereinstimmung (m) von 350 Token wurde beibehalten. Abbildung 5.10 zeigt auch die zunehmend starke Schwankung der Spannweite.

		Cluster			Spannweite
Lauf	**Treffer**	Anzahl	Max. Grösse	∅ Grösse	∅ [Token]
o0999n200m200M01 (2014)	33	14	7	2.36	1.43
o0999n200m200M01 (2015)	115	46	7	2.50	3.95
o0999n200m200M01 (2016)	110	48	5	2.29	4.88
o0999n200m200M01 (2017)	128	48	11	2.67	3.85
o0999n200m300M01 (2018)	19	9	3	2.11	1.56
o0999n200m250M01 (2019)	91	40	6	2.27	4.13
o0999n200m200M01 (2020)	218	98	8	2.22	3.53
o0999n200m350M01 (2021)	1’092	455	15	2.4	4.01
o0999n200m350M01 (2022)	2’675	1’006	40	2.66	3.97

Tabelle 5.8: Passim-Läufe für jahrgangsweise Deduplikation

Der sehr hohe Jaccard-Koeffizient für das Jahr 2018 illustriert Abbildung 5.9. Allerdings hatte 2018 auch die kleinste Trefferzahl. Die steigende Heterogenität ab 2021 wird auch ersichtlich. Mit Werten über oder in der Nähe von 0.9 der restlichen Jahrgänge konnte von einer genügend hohen Homogenität ausgegangen werden.

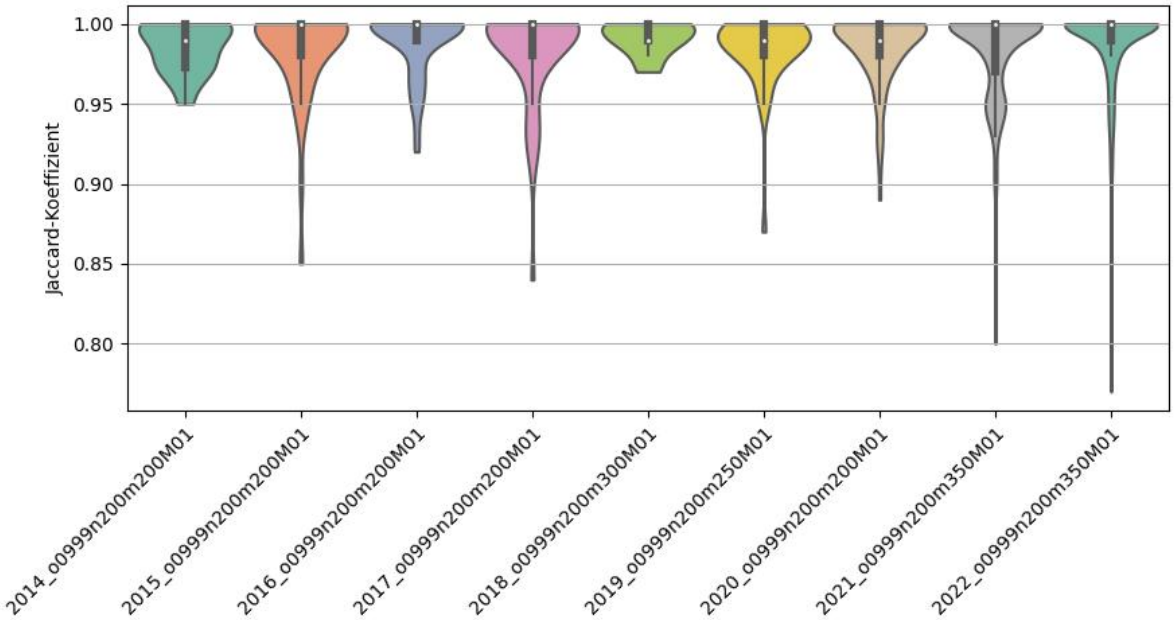

Abbildung 5.9: Jaccard-Koeffizient zu Passim-Läufen für jahrgangsweise Deduplikation

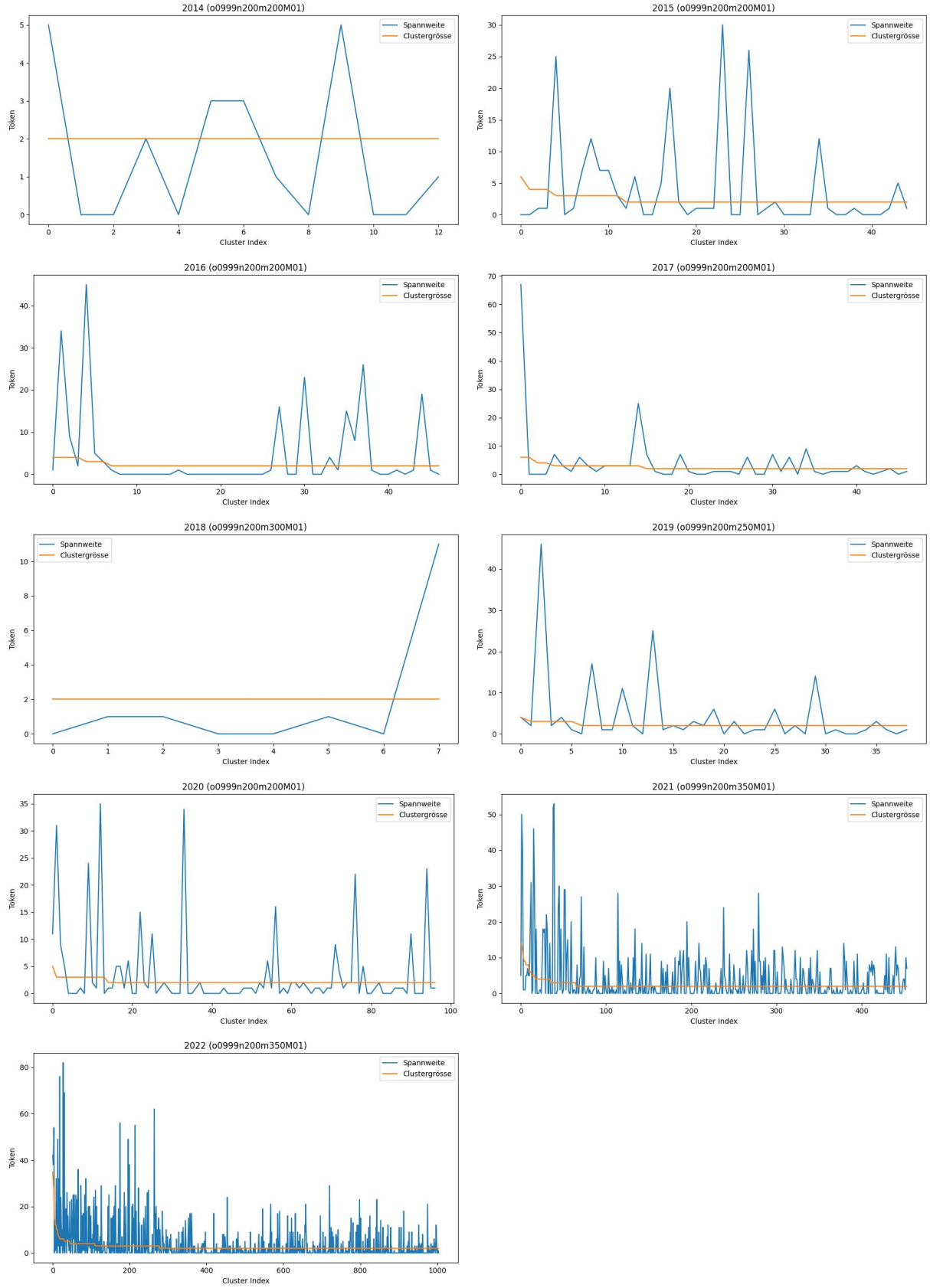

Abbildung 5.10: Passim-Läufe für jahrgangsweise Deduplikation: Spannweite & Clustergrösse

5.2.1.2 Implementierung der Deduplikation

Die Deduplikation benötigt pro Jahrgang eine Referenzdatei. Dazu werden nur die Kennungen von Clustern und enthaltenen Stellenanzeigen aus den dazugehörenden Passim-Läufen in die Referenzdatei übernommen. Unter Beibehaltung einer Stellenanzeige pro Cluster filtert der Programmablauf Duplikate heraus.

5.2.2 Datenaufbereitung für Textwiederverwendung

Die deduplizierten Datensätze gehen weiter in den Konvertierungsprozess. Ein Teilschritt besteht aus dem Anfügen der Part-of-Speech-Tags und der Textzonen-Information an jedes Token. Danach folgt die Textsegmentierung mit Hilfe von Cutter. Im letzten Teilschritt erfolgt die Filterung anhand der wesentlichen Textzonen.

5.2.2.1 Implementierung von Part-of-Speech-Tagging und Zonenfilter

Wie die ersten Experimente gezeigt haben, können in Passim die POS-Tags nur als Einzelzeichen am Ende jedes Tokens angefügt werden. Tabelle 5.9 gibt eine Übersicht der verwendeten Abkürzungen, damit jedem UPOS-Tag ein eindeutiges Einzelzeichen zugeordnet werden konnte. Eigennamen und Substantive erhalten als Ausnahme ein gemeinsames Zeichen und alle Token mit dem UPOS-Tag „PUNCT“ kein Zeichen.

Adjektiv	ADJ	A	Kardinalzahlen	NUM	M
Adposition	ADP	P	Partikel	PART	R
Adverb	ADV	V	Pronomen	PRON	O
Hilfsverb	AUX	U	Eigennamen	PROPN	N
Koordinierende Konjunktion	CCONJ	C	Unterordnende Konjunktion	SCONJ	S
Artikel & pronominales Adjektiv	DET	D	Symbole	SYM	Y
Interjektion	INTJ	I	Verben	VERB	B
Substantiv	NOUN	N	Nicht zuordenbar	X	X

Tabelle 5.9: Abkürzungen der UPOS-Tags

Für den Zonenfilter werden die in Abschnitt 4.1.2 eingeführten Textzonen verwendet. Tabelle 4.2 gibt eine nähere Beschreibung der gefilterten Textzonen 60, 70 und 80. Der Filter sortiert alle Dokumente aus, in denen nicht mindestens ein Token einer der drei genannten Zonen vorkommt.

5.2.2.2 Textsegmentierung mit Cutter

Die vorliegende Arbeit gab Cutter als Zielsprache Deutsch vor. Cutter bezog wichtige Zusatzinformationen aus einer manuell erstellten und sprachabhängigen Abkürzungsliste. Die inkonsistenten Datensätze erforderten eine manuelle Durchsicht. Mitberücksichtigte Quellen dienten dazu, gebräuchliche Abkürzungen geographischer und administrativer Art zu vervollständigen.[21] [22] [23]

Anhand des satzfinalen POS-Tags „+EOS“ in den von Cutter erstellten 5-Tupeln sind die Texte in Sätze segmentierbar. Die resultierende Tokensequenz bildet ein neues Dokument. Jedes Dokument erhält eine neue, eindeutige Kennung mit Satznummer und enthaltenen Textzonen. Die Stellenanzeigen-Kennung gruppiert alle beibehaltenen Sätze als jeweilige *Serie* für Passim.

5.2.3 Zwischenergebnis II

Aus dieser zweiten Experimentphase leitete sich ab, dass eine Deduplikation über das ganze Korpus ohne weitere vorverarbeitende Schritte keinen Sinn macht. Da keine generalisierbare Passim-Konfiguration gefunden werden konnte, ergab sich eine jahrgangsweise Deduplikation.

Die verwendeten Passim-Konfigurationen mussten sich auf eine gewisse Textlänge beschränken. In Folge konnten nur sehr lange Stellenanzeigen-Texte dedupliziert werden, auch um möglichst wenig falsch-positive Dokumente zu erhalten.

Die maschinelle Verarbeitung von Stellenanzeigen gestaltete sich schwierig, da die vorhandenen Daten des SMM noch nicht in genügendem Masse vorverarbeitet vorlagen. Passim liess sich nicht so konfigurieren, dass eine variable Textsegmentierung möglich war. Daraus leitete sich auch die Entscheidung ab, jede Stellenanzeige in einem Vorverarbeitungsschritt auf Satzebene zu segmentieren und Serien einzuführen.

Die bereits mehrfach erwähnten verrauschten Textdaten ergeben dabei eine schwer zu beeinflussende Fehlerrate. Zudem bieten die für Cutter manuell erstellten Abkürzungsverzeichnisse keine abschliessende Sammlung.

[21] https://www.cadastre.ch/de/manual-av/publication/recommendation.detail.document.html/cadastre-internet/de/documents/av-empfehlungen/Empfehlung-Schreibweise-Gemeinde-Ortschaftsnamen-de.pdf.html (Letzter Besuch: 23. März 2023).

[22] https://www.idiotikon.ch/woerterbuch/abk/geogr-liste (Letzter Besuch: 23. März 2023).

[23] https://www.bk.admin.ch/bk/de/home/dokumentation/sprachen/hilfsmittel-textredaktion/schreibweisungen.html (Letzter Besuch: 23. März 2023).

5.3 Experimentelle Phase III

Die folgenden Experimente wendeten sieben Parameter-Kombinationen auf alle vorverarbeiteten Datensätze der Jahrgänge 2014 bis 2022 an. Über das eingeführte Datenfeld „Serien“, wobei jede Serie eine Stellenanzeige repräsentiert, suchte Passim weiter nur zwischen den Stellenanzeigen. Ziel dieses Vorgehens war, dass im Anschluss nach vordefinierten Kriterien aus allen Läufen entsprechende Cluster in die Datenstruktur zur Textwiederverwendung übernommen werden.

5.3.1 Parameter-Auswahl

In den vorangegangenen Experimenten hatten sich v. a. die Textlückengrösse (g) und die durchschnittliche Wortlänge für den Abgleich (w) sowie die Kombination aus n-Gramm-Index (n) und n-Gramm-Übereinstimmung (m) als einflussreich auf die angestrebte Homogenität in den Clustern und die Spannweite beziehungsweise die Textlänge gezeigt (Tabelle 5.10).

g	Voreinstellung= 100	Minimale Grösse der Lücke, die Passagen separiert
m	Voreinstellung= 5	Minimale Anzahl n-Gramm-Übereinstimmungen zwischen Dokumenten
n	Voreinstellung= 5	Index n-Gramm-Funktionen
w	Voreinstellung= 2	Minimale durchschnittliche Wortlänge für den Abgleich

Tabelle 5.10: Auswahl der Passim-Parameter für Textwiederverwendung

Tabelle 5.11 bietet eine Übersicht der Passim-Läufe in diesen Experimenten. Dabei schwankte sowohl die Treffer- als auch die Clusteranzahl sehr stark. Die durchschnittliche Clustergrösse war in allen sieben Läufen dagegen weniger variierend. Die mittlere Spannweite blieb mehrheitlich unter vier Token.

		Cluster			**Spannweite**
Lauf	**Treffer**	Anzahl	Max. Grösse	∅ Grösse	∅ [Token]
g5 (A)	816’330	214’278	137	3.81	3.80
g5m20 (B)	704’626	187’063	127	3.76	4.14
g5m20w5 (C)	649’269	173’722	127	3.73	4.07
g5n3m20 (D)	560’364	153’160	140	3.66	3.68
g5n3m100 (E)	93’647	32’808	54	2.85	2.26
g5n3m100w5 (F)	79’272	28’094	52	2.82	1.95
g1n5w5 (G)	608’347	174’159	100	3.49	1.98

Tabelle 5.11: Passim-Läufe für Textwiederverwendung

5.3.1.1 Passim-Konfiguration A

Passim-Konfiguration A verkleinerte nur die Textlückengrösse (g) von 100 Token in der Voreinstellung auf fünf Token. In den vorangegangenen Experimenten konnte so eine bessere Textsegmentierung erreicht werden. Abbildung 5.11 verdeutlicht wie die Homogenität gemessen durch den Jaccard-Koeffizienten zunimmt, je kleiner die Cluster werden. Unter Beibehaltung langer Textpassagen in allen Clustern verringerte sich die Spannweite mit Abnahme der Clustergrösse. Zudem verzeichnete dieser Lauf die meisten Treffer sowie Cluster (Tabelle 5.11).

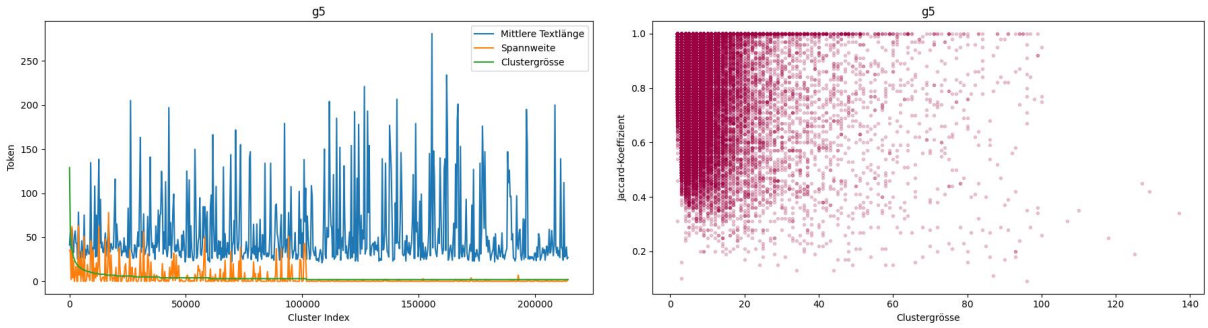

Abbildung 5.11: Clusterprofil zu Passim-Konfiguration A

5.3.1.2 Passim-Konfiguration B

Passim-Konfiguration B übernahm die verkleinerte Textlückengrösse (g) aus Konfiguration A und erhöhte die n-Gramm-Übereinstimmung (m) von fünf n-Grammen auf 20. Mit einer Steigerung der n-Gramm-Übereinstimmung (m) ging eine verbesserte Textähnlichkeit in den Clustern einher. Die Textlänge und die Clustergrösse veränderten sich im Mittel kaum über alle Cluster. Die Spannweite und der Jaccard-Koeffizient zeigten ebenfalls eine Tendenz zur Verbesserung der Homogenität (Abbildung 5.12).

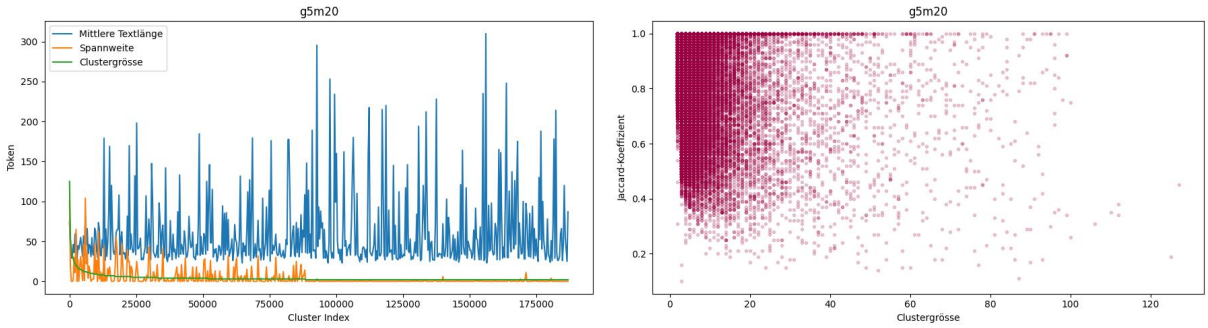

Abbildung 5.12: Clusterprofil zu Passim-Konfiguration B

5.3.1.3 Passim-Konfiguration C

Passim-Konfiguration C erweiterte neben der Textlückengrösse (g) und der n-Gramm-Übereinstimmung (m) die Konfiguration erneut um eine Erhöhung der durchschnittlichen Wortlänge für den Abgleich (w) von zwei n-Grammen auf fünf n-Gramme. Abbildung 5.13 illustriert den kleinen Einfluss auf Spannweite und Jaccard-Koeffizient. Treffer- und Clusterzahl sanken weiter (Tabelle 5.11).

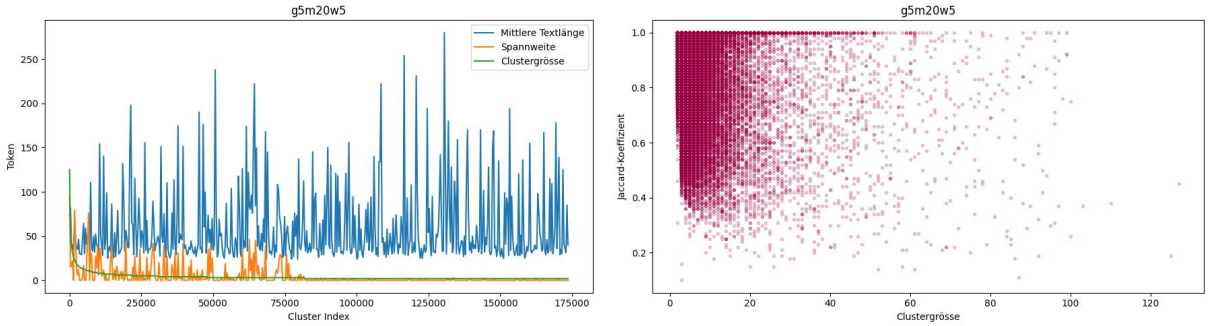

Abbildung 5.13: Clusterprofil zu Passim-Konfiguration C

5.3.1.4 Passim-Konfiguration D

Bei Passim-Konfiguration D erweiterte der n-Gramm-Index (n) die Konfiguration B, um 3-Gramme anstelle der 5-Gramme aus der Voreinstellung. Aus Tabelle 5.11 lässt sich eine weiter gesunkene Trefferanzahl ablesen, wohingegen keine bemerkenswerten Veränderungen in Abbildung 5.14 festzustellen sind.

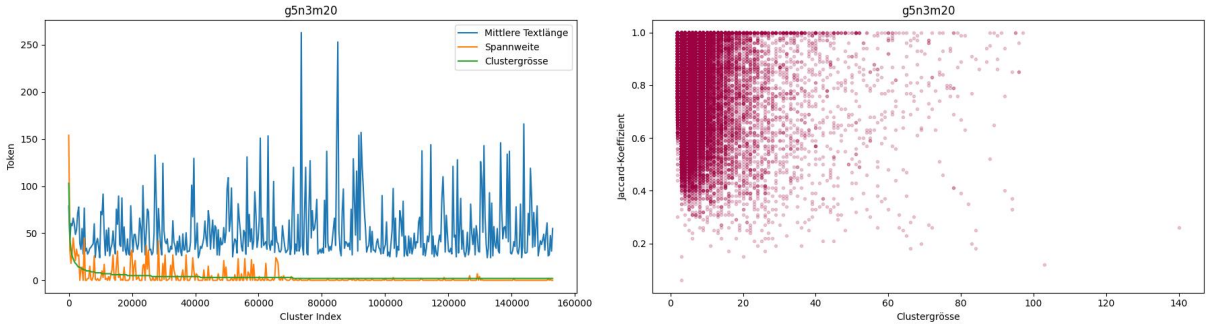

Abbildung 5.14: Clusterprofil zu Passim-Konfiguration D

5.3.1.5 Passim-Konfiguration E

Passim-Konfiguration E beinhaltete eine Erhöhung der n-Gramm-Übereinstimmung (m) von 20 n-Grammen auf 100. Wie Tabelle 5.11 aufzeigt, ergab sich ein deutlicher Abfall der Trefferan-

zahl. Der Jaccard-Koeffizient zeigt die verbesserte Homogenität und die kleine Spannweite (Abbildung 5.15). Die Textlänge war immer noch schwankend.

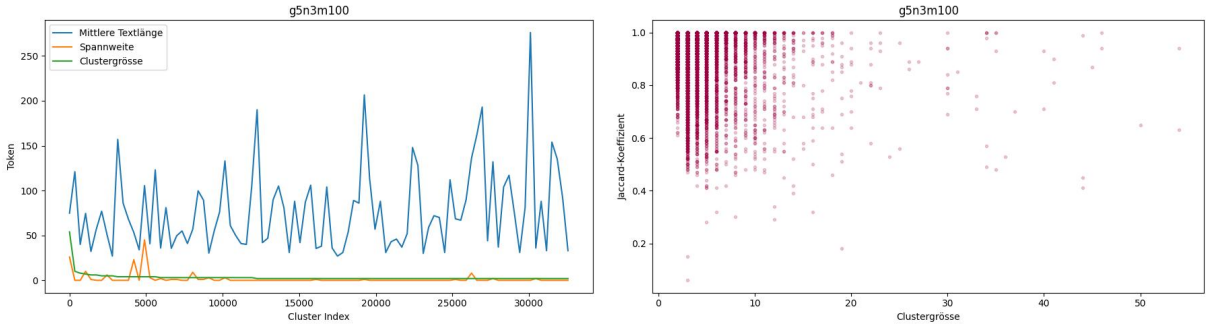

Abbildung 5.15: Clusterprofil zu Passim-Konfiguration E

5.3.1.6 Passim-Konfiguration F

Passim-Konfiguration F nahm zur Konfiguration E die durchschnittliche Wortlänge für den Abgleich (w) dazu. Wieder mit dem gleichen Wert von fünf n-Grammen wie in Konfiguration C. Abbildung 5.16 ergibt nur marginale Veränderungen gegenüber Konfiguration C. Die mittlere Spannweite war gesunken.

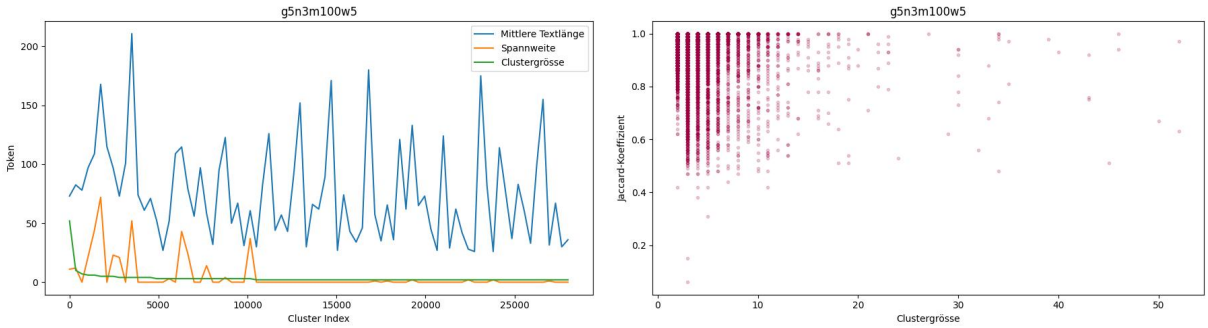

Abbildung 5.16: Clusterprofil zu Passim-Konfiguration F

5.3.1.7 Passim-Konfiguration G

Passim-Konfiguration G entsprach der Konfiguration aus D. A. Smith et al. (2013, S. 88) mit einem n-Gramm-Index (n) von fünf, einer Textlückengrösse (g) von einem 5-Gramm und durchschnittlicher Wortlänge für den Abgleich (w) von fünf 5-Grammen (n=5, g=1, w=5). Die Trefferanzahl nahm wieder deutlich zu, aber die mittlere Spannweite blieb klein (Tabelle 5.11). Der Jaccard-Koeffizient in Abbildung 5.16 zeigte eine bessere Homogenität über alle Cluster verteilt an.

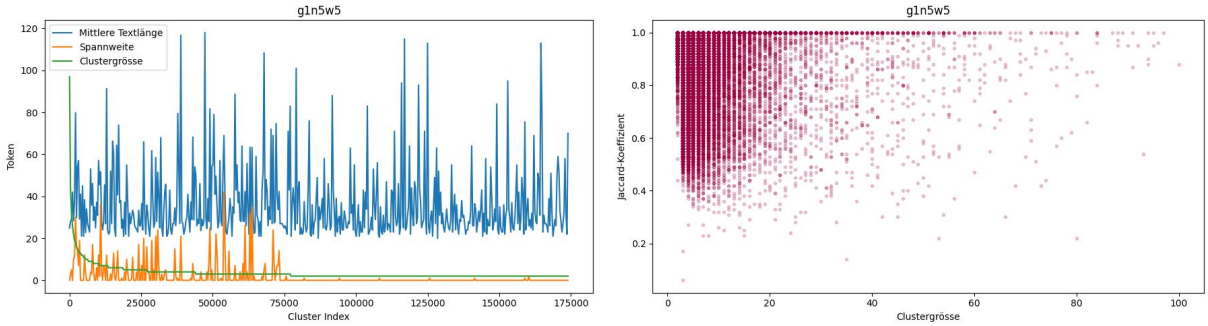

Abbildung 5.17: Clusterprofil zu Passim-Konfiguration G

5.3.2 Zwischenergebnis III

Alle sieben Passim-Konfigurationen hatten ihre Vor- und Nachteile. Die Auswirkungen der variierten Parameter machten sich in allen erfassten Kennzahlen bemerkbar. Somit wurden auch Cluster aus allen sieben Läufen in die Datenstruktur aufgenommen, um möglichst viele Phrasenmuster zu akkumulieren.

5.4 Aufbau der Datenstruktur für Textwiederverwendung

Die Datenstruktur ist eine Reihe von JSONL-Dateien. Jede Datei entspricht einem Passim-Lauf und erhält eine Kennung aus den Angaben zur jeweiligen Konfiguration. In jeder Datei ist jeder übernommene Cluster ein JSONL-Eintrag und bekommt eine Kennung mit einem Datenfeld des Passim-Laufs und einem Datenfeld mit der von Passim selbst vergebenen Clusterkennziffer. Alle Cluster werden über ihren Jaccard-Koeffizienten gefiltert ($J \geq 0.67$). Die geeigneten Kenngrössen werden als Datenfelder ebenfalls hinzufügen.

5.4.1 Auswahl geeigneter Kennzahlen als Metadaten

Die mitgegebenen Kennzahlen sind

- die Clustergrösse,
- die mittlere Textlänge in Token,
- der Jaccard-Koeffizient jeweils zum kürzesten Satz im Cluster berechnet (Mass für die Textähnlichkeit),
- die Spannweite der Texte in Token,
- alle im Cluster einmal vorkommenden Token und

- alle Phrasen des Clusters mit der dazugehörenden Stellenanzeigen-Kennung (id) und der von Passim vergebenen Treffer-Kennung (uid).

5.4.2 Schema

Es ergibt sich folgendes Schema für jeden JSONL-Eintrag in der Datenstruktur:

{
„cid“ : Passim-Lauf_Clusterkennziffer,
„run“ : Passim-Lauf,
„cluster“ : Clusterkennziffer,
„size“ : Clustergrösse,
„average textlength“ : mittlere Textlänge im Cluster,
„average jaccard“ : mittlere Jaccard-Koeffizient jeweils zum kürzesten Satz im Cluster,
„span“ : Spannweite,
„jobads“ : Stellenanzeigen-Identifikationskennung,
„vocabulary“ : alle einmal vorkommenden Token im Cluster,
„phrases“ : alle Phrasen mit Stellenanzeigen-Kennung (id) und Passim-Kennung (uid)
}

6 Schlussbetrachtung und Ausblick

Die vorliegende Arbeit hatte zum Ziel, typische, formelhafte Textbausteine in den Stellenanzeigen des SMM-Korpus im Sinne einer Textwiederverwendung zu erfassen.

Dabei interessierten v. a. Textzonen, in denen Stellen beschrieben oder erforderliche Persönlichkeitsmerkmale aufgelistet werden sowie Ausbildung und Qualifikation im Fokus stehen, weil sich dort Veränderungen im Hinblick auf Sprachwandel-Phänomene mittels formelhafter Phrasen deutlicher zeigen. Wie sich Unternehmen präsentieren oder stetig wiederholende administrative Textpassagen stellen diesbezüglich weniger aufschlussreiche Bereiche dar.

Mit Hilfe von Passim als Tool für die Detektion von Textwiederverwendung konnte ein Schritt in die richtige Richtung gegangen werden. Passim hat sich als mächtiges Tool erwiesen, wobei die Anwendung viel Vor- und Nachbearbeitung der Daten erfordert. Auch konnten keine kurzen, formelhaften Wendungen direkt mit Passim erzeugt werden, da die Segmentierung der Texte nicht über die Parameter kontrollierbar ist. Dieser Umstand erschwerte es ebenfalls, eine zufriedenstellende Deduplikation durchzuführen.

Eine denkbare Ergänzung, um kleine, formelhafte Phrasen besser zu erschliessen, ist die Kollokationsanalyse, beispielsweise mit der Gensim-Bibliothek. Beide Ansätze zusammen bilden einen möglichen Ausgangspunkt für die Entwicklung einer benutzerfreundlichen Webanwendung für Suchabfragen.

Im Bereich der Deduplikation ist eine Optimierung der Passim-Einstellungen wünschenswert. Zunehmende Datenmengen gestalteten diesen Prozess merklich schwieriger. Hier wäre die Implementierung eines Algorithmus vorstellbar, der anhand der Textlänge die Daten gruppiert und jeweils die beste Passim-Einstellung wählt.

In der im Rahmen dieser Arbeit entstandenen Datenstruktur werden die gefundenen, wiederverwendeten Textpassagen u. a. mit Hilfe der Textähnlichkeit charakterisiert. Der dazu verwendete Jaccard-Koeffizient liess sich gut implementieren und ermöglichte, die Passim-Konfigurationen systematisch zu analysieren. Eine Reihe von Metadaten kann nun für weitere Untersuchungen herangezogen werden. Umfangreiche Sprachwandelforschung könnte sich aufgrund der Datenmangellage im 20. Jahrhundert jedoch schwierig gestalten.

Literatur

Adams, J., Poelmans, K., Hendrickx, I., & Larson, M. (2022). Doing not Being: Concrete Language as a Bridge from Language Technology to Ethnically Inclusive Job Ads. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion* (S. 19–25). Association for Computational Linguistics. https://aclanthology.org/2022.ltedi-1.3

Aggarwal, C. C. (2018). *Machine Learning for Text* (1st). Springer.

Alshomary, M., Völske, M., Licht, T., Wachsmuth, H., Stein, B., Hagen, M., & Potthast, M. (2018). *Wikipedia Text Reuse: Within and Without*. arXiv: 1812.09221 [cs.IR].

Arnold, F., & Jäschke, R. (2021). Lotte and Annette: A Framework for Finding and Exploring Key Passages in Literary Works. In *Proceedings of the Workshop on Natural Language Processing for Digital Humanities* (S. 55–63). NLP Association of India (NLPAI). https://aclanthology.org/2021.nlp4dh-1.7

Avetisyan, K., Malajyan, A., Ghukasyan, T., & Avetisyan, A. (2023). *A Simple and Effective Method of Cross-Lingual Plagiarism Detection*. arXiv: 2304.01352 [cs.CL].

Bakhteev, O., Chekhovich, Y., Grabovoy, A., Gorbachev, G., Gorlenko, T., Grashchenkov, K., Ivakhnenko, A., Kildyakov, A., Khazov, A., Komarnitsky, V., Nikitov, A., Ogaltsov, A., & Sakharova, A. (2022). Cross-Language Plagiarism Detection: A Case Study of European Languages Academic Works. In *Academic Integrity: Broadening Practices, Technologies, and the Role of Students: Proceedings from the European Conference on Academic Integrity and Plagiarism 2021* (S. 143–161). Springer International Publishing.

Bakhteev, O., Ogaltsov, A., Khazov, A., Safin, K., & Kuznetsova, R. (2019). CrossLang: the System of Cross-lingual Plagiarism Detection. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Workshop on Document Intelligence*. https://openreview.net/forum?id=BkxiG6qqIr

Ball, R. (2022). *Graph - Text Reuse in Rare Books*. ETH Library. Verfügbar 21. Mai 2023 unter https://graph-rare-books.ethz.ch/#/home

Bär, D., Zesch, T., & Gurevych, I. (2012). Text Reuse Detection Using a Composition of Text Similarity Measures. In *Proceedings of COLING 2012* (S. 167–184). The COLING 2012 Organizing Committee. https://aclanthology.org/C12-1011

Barber, M. (2023). *Adventures in Alignments: Training an Algorithm to Recognise Text Reuse*. Aga Khan University International, UK. Verfügbar 17. Juni 2023 unter http://kitab-

project.org/Adventures-in-Alignments-Training-an-Algorithm-to-Recognise-Text-Reuse/

Bechmann, S. (2016). *Sprachwandel – Bedeutungswandel: Eine Einführung*. A. Francke Verlag.

Belinkov, Y., Magidow, A., Barrón-Cedeño, A., Shmidman, A., & Romanov, M. (2019). Studying the History of the Arabic Language: Language Technology and a Large-Scale Historical Corpus. *Language Resources and Evaluation*, *53*. https://arxiv.org/abs/1809.03891

Belyy, A., Dubova, M., & Nekrasov, D. (2018). Improved Evaluation Framework for Complex Plagiarism Detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (S. 157–162). Association for Computational Linguistics. https://aclanthology.org/P18-2026

Bendel, S. (1999). *Von der Stellenausschreibung zur Personalwerbung. Sprachliche Veränderungen in den Stelleninseraten und ihre Bedeutung* [unpublished]. Verfügbar 22. Mai 2023 unter https://www.sbendel.ch/publikationen/unpubliziertes/

Bilgin, O. (2022). A Matrix-Based Heuristic Algorithm for Extracting Multiword Expressions from a Corpus. In *Proceedings of the 18th Workshop on Multiword Expressions @LREC2022* (S. 37–48). European Language Resources Association. https://aclanthology.org/2022.mwe-1.7

Blankenship, A. (2021, 24. Februar). *A Dataset of Nineteenth-Century American Recipes*. The Viral Texts Project. Verfügbar 9. Mai 2023 unter https://viraltexts.org/2021/02/24/c19-recipes/

Broder, A. Z., Glassman, S. C., Manasse, M. S., & Zweig, G. (1997). Syntactic Clustering of the Web [Papers from the Sixth International World Wide Web Conference]. *Computer Networks and ISDN Systems*, *29*(8), 1157–1166. https://www.sciencedirect.com/science/article/pii/S0169755297000317

Bubenhofer, N. (2017). Kollokationen, n-Gramme, Mehrworteinheiten. In *Handbuch Sprache in Politik und Gesellschaft* (S. 69–93, Bd. 19). De Gruyter.

Büchler, M. (2016). *TRACER: A Text Reuse Detection Machine*. eTRAP (Electronic Text Reuse Acquisition Project). Verfügbar 21. Mai 2023 unter https://www.etrap.eu/research/tracer/

Büchler, M., Franzini, E., & Franzini, G. (2016). *Historical Text Reuse: What Is It?* eTRAP (electronic Text Reuse Acquisition Project). Verfügbar 20. Juni 2023 unter http://www.etrap.eu/historical-text-re-use/

Büchler, M., Franzini, G., Franzini, E., & Moritz, M. (2014). Scaling Historical Text Re-use. In *2014 IEEE International Conference on Big Data (Big Data)* (S. 23–31). Institute of Electrical; Electronics Engineers (I triple E). https://ieeexplore.ieee.org/document/7004449

Buchmann, M., Buchs, H., Busch, F., Clematide, S., Gnehm, A.-S., & Müller, J. (2022). Swiss Job Market Monitor: A Rich Source of Demand-Side Micro Data of the Labour Market. *European Sociological Review*, *38*(6), 1001–1014. https://doi.org/10.1093/esr/jcac002

Bussmann, H. (2008). *Lexikon der Sprachwissenschaften* (4. Aufl.). Kröner Verlag.

Cha, S.-H. (2007). Comprehensive Survey on Distance/Similarity Measures Between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, *1*(4), x300–307. https://www.naun.org/main/NAUN/ijmmas/mmmas-49.pdf

Citron, D. T., & Ginsparg, P. (2015). Patterns of Text Reuse in a Scientific Corpus. *Proceedings of the National Academy of Sciences of the United States of America*, *112*(1), 25–30. https://europepmc.org/articles/PMC4291616

Clough, P., Gaizauskas, R., & Piao, S. L. (2002). Building and Annotating a Corpus for the Study of Journalistic Text Reuse. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)* (S. 1678–1685). European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2002/pdf/218.pdf

Clough, P., Gaizauskas, R., Piao, S. S., & Wilks, Y. (2002). METER: MEasuring TExt Reuse. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (S. 152–159). Association for Computational Linguistics. https://aclanthology.org/P02-1020

Coffee, N., Koenig, J.-P., Shakti, P., Ossewaarde, R., Forstall, C., & Jacobson, S. (2013). The Tesserae Project: Intertextual analysis of Latin poetry. *Literary and Linguistic Computing*, 221–228.

Cordell, R., Smith, D., & Mullen, A. (2015). Reprinting, Circulation, and the Network Author in Antebellum Newspapers. *American Literary History*, *27*(3), 417–445.

Cordell, R., Smith, D. A., Mullen, A., Fitzgerald, J., & Kinias, T. (2023). *Going the Rounds: Virality in Nineteenth-Century American Newspapers*. Northeastern University, United States. Verfügbar 17. Mai 2023 unter https://viraltexts.org/rounds/

Cordella, B., Greco, F., Meoli, P., Palermo, V., & Grasso, M. (2020). Educational Culture and Job Market: A Text Mining Approach. In *Text Analytics: Advances and Challenges* (S. 287–297). Springer.

Crochemore, M., Lecroq, T., & Rytter, W. (2021). *125 Problems in Text Algorithms: With Solutions*. Cambridge University Press.

D'hondt, E., Grouin, C., Névéol, A., Stamatatos, E., & Zweigenbaum, P. (2016). Detection of Text Reuse in French Medical Corpora. In *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)* (S. 108–114). The COLING 2016 Organizing Committee. https://aclanthology.org/W16-5112

Diewald, N., Kupietz, M., & Lüngen, H. (2022). Tokenizing on Scale. Preprocessing Large Text Corpora on the Lexical and Sentence Level. In *Dictionaries and Society. Proceedings of the XX EURALEX International Congress* (S. 208–221). IDS-Verlag. https://github.com/KorAP/Tokenizer-Evaluation

Dipper, S., Klabunde, R., & Mihatsch, W. (2018). *Linguistik: Eine Einführung (nicht nur) für Germanisten, Romanisten und Anglisten*. Springer.

Düring, M., & van den Bosch, A. (2014). Multi-perspective Event Detection in Texts Documenting the 1944 Battle of Arnhem. In *Text Mining: From Ontology Learning to Automated Text Processing Applications* (S. 201–219). Springer.

Ehrmann, M., Bunout, E., & Düring, M. (2019). Historical Newspaper User Interfaces: A Review [Session Digital Humanities Digital Scholarship Special Interest Group]. In *85th World Library and Information Congress of IFLA*. https://doi.org/10.5281/zenodo.3404155

Ehrmann, M., Romanello, M., Clematide, S., Ströbel, P. B., & Barman, R. (2020). Language Resources for Historical Newspapers: the Impresso Collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (S. 958–968). European Language Resources Association. https://aclanthology.org/2020.lrec-1.121

Elagina, D. (2022). Encoding of Text Reuse in the Project Beta Masaheft. *Journal of the Text Encoding Initiative*. https://doi.org/10.4000/jtei.3763

Elsafty, A., Riedl, M., & Biemann, C. (2018). Document-based Recommender System for Job Postings using Dense Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)* (S. 216–224). Association for Computational Linguistics. https://aclanthology.org/N18-3027

Elsayed, T., Lin, J., & Oard, D. (2008). Pairwise Document Similarity in Large Collections with MapReduce. In *Proceedings of ACL-08: HLT, Short Papers* (S. 265–268). Association for Computational Linguistics. https://aclanthology.org/P08-2067

Filatkina, N. (2018). *Historische formelhafte Sprache: Theoretische Grundlagen und methodische Herausforderungen*. De Gruyter.

Filatkina, N., Steyer, K., & Stumpf, S. (2018). *Formelhafte Sprache in Text und Diskurs*. De Gruyter.

Francopoulo, G., Mariani, J., & Paroubek, P. (2016). A Study of Reuse and Plagiarism in LREC Papers. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (S. 1890–1897). European Language Resources Association (ELRA). https://aclanthology.org/L16-1298

Franzini, G., Passarotti, M., Moritz, M., & Büchler, M. (2018). Using and Evaluating TRACER for an Index fontium computatus of the Summa contra Gentiles of Thomas Aquinas. In *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)* (S. 199–205). aAccademia University Press. https://ceur-ws.org/Vol-2253/paper22.pdf

Fröbe, M., Bevendorff, J., Reimer, J. H., Potthast, M., & Hagen, M. (2020). Sampling Bias Due to Near-Duplicates in Learning to Rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (S. 1997–2000). Association for Computing Machinery. https://doi.org/10.1145/3397271.3401212

Gienapp, L., Kircheis, W., Sievers, B., Stein, B., & Potthast, M. (2022). *STEREO: Scientific Text Reuse in Open Access Publications*. arXiv: 2112.11800 [cs.DL].

Gladstone, C., & Tharsen, J. (2022). *TextPAIR - Pairwise Alignment for Intertextual Relations*. The ARTFL Project. Verfügbar 21. Mai 2023 unter https://artfl.blogspot.com/2018/12/textpair-new-high-performance-sequence.html

Gnehm, A.-S. (2018). *Text Zoning for Job Advertisements with Bidirectional LSTMs* [Bachelor's Thesis]. University of Zurich: Institute for Computational Linguistics. https://www.stellenmarktmonitor.uzh.ch/dam/jcr:70806570-8dac-4b72-a6a7-77e6c5d99166/TextZoning_Gnehm2018.pdf

Gnehm, A.-S., Bühlmann, E., Buchs, H., & Clematide, S. (2022). Fine-Grained Extraction and Classification of Skill Requirements in German-Speaking Job Ads. In *Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)* (S. 14–24). Association for Computational Linguistics. https://aclanthology.org/2022.nlpcss-1.2

Gnehm, A.-S., Bühlmann, E., & Clematide, S. (2022). Evaluation of Transfer Learning and Domain Adaptation for Analyzing German-Speaking Job Advertisements. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (S. 3892–3901). European Language Resources Association. https://aclanthology.org/2022.lrec-1.414

Gnehm, A.-S., & Clematide, S. (2020). Text Zoning and Classification for Job Advertisements in German, French and English. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science* (S. 83–93). Association for Computational Linguistics. https://aclanthology.org/2020.nlpcss-1.10

Graën, J., Bertamini, M., & Volk, M. (2018). Cutter – a Universal Multilingual Tokenizer. In M. Cieliebak, D. Tuggener & F. Benites (Hrsg.), *Swiss Text Analytics Conference* (S. 75–81). CEUR-WS. https://doi.org/10.5167/uzh-157243

Guillaume, S., Wisniewski, G., Macaire, C., Jacques, G., Michaud, A., Galliot, B., Coavoux, M., Rossato, S., Nguyên, M.-C., & Fily, M. (2022). Fine-tuning pre-trained models for Automatic Speech Recognition, experiments on a fieldwork corpus of Japhug (Trans-Himalayan family). In *Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages* (S. 170–178). Association for Computational Linguistics. https://aclanthology.org/2022.computel-1.21

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press.

Guthrie, D., Allison, B., Liu, W., Guthrie, L., & Wilks, Y. (2006). A Closer Look at Skip-gram Modelling. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)* (S. 1222–1225). European Language Resources Association. http://www.lrec-conf.org/proceedings/lrec2006/pdf/357_pdf.pdf

Hausmann, F. J. (2004). Was sind eigentlich Kollokationen? In *Wortverbindungen - mehr oder weniger fest* (S. 309–334, Bd. 2003). De Gruyter.

Henzinger, M. (2006). Finding Near-Duplicate Web Pages: A Large-scale Evaluation of Algorithms. In *Proceedings of the 29th Annual International ACM SIGIR Conference on*

Research and Development in Information Retrieval (S. 284–291). https://dl.acm.org/doi/10.1145/1148170.1148222

Hiltmann, T., Keupp, J., Althage, M., & Schneider, P. (2021). Digital Methods in Practice: The Epistemological Implications of Applying Text Re-Use Analysis to the Bloody Accounts of the Conquest of Jerusalem (1099). *Geschichte und Gesellschaft*, *47*(1), 122–156. https://doi.org/10.13109/gege.2021.47.1.122

Huston, S., Moffat, A., & Croft, W. B. (2011). Efficient Indexing of Repeated N-Grams. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining, WSDM 2011* (S. 127–136). Association for Computing Machinery. https://doi.org/10.1145/1935826.1935857

Iwatsuki, K., & Aizawa, A. (2018). Using Formulaic Expressions in Writing Assistance Systems. In *Proceedings of the 27th International Conference on Computational Linguistics* (S. 2678–2689). Association for Computational Linguistics. https://aclanthology.org/C18-1227

Iwatsuki, K., & Aizawa, A. (2021). Communicative-Function-Based Sentence Classification for Construction of an Academic Formulaic Expression Database. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (S. 3476–3497). Association for Computational Linguistics. https://aclanthology.org/2021.eacl-main.304

Iwatsuki, K., Boudin, F., & Aizawa, A. (2020). An Evaluation Dataset for Identifying Communicative Functions of Sentences in English Scholarly Papers. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (S. 1712–1720). European Language Resources Association. https://aclanthology.org/2020.lrec-1.212

Jensen, K. N., Zhang, M., & Plank, B. (2021). De-identification of Privacy-related Entities in Job Postings. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)* (S. 210–221). Linköping University Electronic Press. https://aclanthology.org/2021.nodalida-main.21

Kiener, F., Gnehm, A.-S., Clematide, S., & Backes-Gellner, U. (2022). IT skills in vocational training curricula and labour market outcomes. *Journal of Education and Work*, *35*(6-7), 614–640. https://doi.org/10.1080/13639080.2022.2126968

Koolen, M., & Hoekstra, R. (2022). Detecting Formulaic Language Use in Historical Administrative Corpora. In *Proceedings of the Computational Humanities Research Conference 2022 (CHR 2022)* (S. 127–151, Bd. 3290). CEUR-WS. https://ceur-ws.org/Vol-3290/long_paper5740.pdf

Kosub, S. (2016). *A Note on the Triangle Inequality for the Jaccard Distance*. arXiv: 1612.02696 [cs.DM].

Kothwal, R., & Varma, V. (2013). Cross Lingual Text Reuse Detection Based on Keyphrase Extraction and Similarity Measures. In *Multilingual Information Access in South Asian Languages: Second International Workshop, FIRE 2010, Gandhinagar, India, February*

19-21, 2010 and Third International Workshop, FIRE 2011, Bombay, India, December 2-4, 2011, Revised Selected Papers (S. 71–78, Bd. 7536).

Kumpulainen, S., & Late, E. (2022). Struggling with Digitized Historical Newspapers: Contextual Barriers to Information Interaction in History Research Activities. *Journal of the Association for Information Science and Technology*, *73*(7), 1012–1024. https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.24608

Ladstätter, F. (2004). Die ünsichtbare Handïn der Sprache. Eine kritische Betrachtung von Kellers Sprachwandeltheorie. *Linguistik Online*, *18*(1). https://bop.unibe.ch/linguistik-online/article/view/767

Liebl, B., & Burghardt, M. (2020). The Vectorian - Eine parameterisierbare Suchmaschine für intertextuelle Referenzen. In *Spielräume - Digital Humanities zwischen Modellierung und Interpretation* (S. 232–235). Digital Humanities im deutschsprachigen Raum. https://www.researchgate.net/publication/345626510_The_Vectorian_-Eine_parametrisierbare_Suchmaschine_fur_intertextuelle_Referenzen

Lin, P. (2018). *The Prosody of Formulaic Sequences: A Corpus and Discourse Approach*. Bloomsbury Publishing.

Loose, F., Becker, S., Potthast, M., & Stein, B. (2008). Retrieval-Technologien für die Plagiaterkennung in Programmen. In J. Baumeister & M. Atzmüller (Hrsg.), *LWA 2008 – Workshop-Woche: Lernen, Wissen & Adaptivität, Würzburg, 6.-8. Oktober 2008, Proceedings* (S. 5–12, Bd. 448). Department of Computer Science, University of Würzburg, Germany. https://webis.de/downloads/publications/papers/loose_2008.pdf

Lynch, R. J. (2020). Self-Revision and the Arabic Historical Tradition: Identifying Textual Reuse and Reorganization in the Works of al-Baladhuri. In *Recreating the Medieval Globe: Acts of Recycling, Revision and Relocation*. Amsterdam University Press.

Madabushi, H. T., Gow-Smith, E., Garcia, M., Scarton, C., Idiart, M., & Villavicencio, A. (2022). SemEval-2022 Task 2: Multilingual Idiomaticity Detection and Sentence Embedding. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)* (S. 107–121). Association for Computational Linguistics. https://aclanthology.org/2022.semeval-1.13

Manjavacas, E. (2020). *RETRIEVE: A Text Reuse Software Package*. MIT License (MIT). Verfügbar 21. Mai 2023 unter https://github.com/emanjavacas/retrieve

Manning, C., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Mariani, J., Francopoulo, G., & Paroubek, P. (2016). A Study of Reuse and Plagiarism in Speech and Natural Language Processing Papers. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)* (S. 72–83). https://aclanthology.org/W16-1509

McGinnis, S., & Madden, T. L. (2004). BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Research*, *32*(suppl_2), W20–W25. Verfügbar 20. Mai 2023 unter https://academic.oup.com/nar/article/32/suppl_2/W20/1040657

Meier, J. (2018). Sprachwandel in der Anzeigenwerbung vom 18. bis zum 20. Jahrhundert. In *Sprachwandel im Deutschen* (S. 349–360, Bd. 19). De Gruyter.

Meinecke, C., & Jänicke, S. (2020). Detecting Text Reuse and Similarities between Artists in Rap Music through Visualization. In *LEVIA'20: Leipzig Symposium on Visualization in Applications 2020*. https://osf.io/j4cn8/

Meinecke, C., Schebera, J., Eschrich, J., & Wiegreffe, D. (2022). Visualizing Similarities between American Rap-Artists. In *LEVIA'22: Leipzig Symposium on Visualization in Applications 2022*. https://doi.org/10.2312/evp.20221129

Moritz, M., Hellrich, J., & Büchel, S. (2018). A Method for Human-Interpretable Paraphrasticality Prediction. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature* (S. 113–118). Association for Computational Linguistics. https://aclanthology.org/W18-4513

Olsen, M., Horton, R., & Roe, G. (2011). Something Borrowed: Sequence Alignment and the Identification of Similar Passages in Large Text Collections. *Digital Studies/le Champ Numérique*, *2*(1). https://www.digitalstudies.org/article/id/7224/

O'Neill, H., Welsh, A., Smith, D. A., Roe, G., & Terras, M. (2021). Text mining Mill: Computationally Detecting Influence in the Writings of John Stuart Mill from Library Records. *Digital Scholarship in the Humanities*, *36*(4), 1013–1029. https://academic.oup.com/dsh/article/36/4/1013/6153976

Paju, P., Rantala, H., & Salmi, H. (2023). Towards an Ontology and Epistemology of Text Reuse. In *Digitised Newspapers - A New Eldorado for Historians?: Reflections on Tools, Methods and Epistemology* (S. 253–274, Bd. 3). De Gruyter Oldenbourg.

Petrov, S., Das, D., & McDonald, R. (2011). *A Universal Part-of-Speech Tagset*. arXiv: 1104.2086 [cs.CL].

Peverelli, A., van Erp, M., & Bloemendal, J. (2022). Tracking Textual Similarities in Neo-Latin Drama Networks. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (S. 5295–5303). European Language Resources Association. https://aclanthology.org/2022.lrec-1.567

Pfanzelter, E., Oberbichler, S., Marjanen, J., Langlais, P.-C., & Hechl, S. (2020). *Digital Interfaces of Historical Newspapers: Opportunities, Restrictions and Recommendations*. arXiv: 2006.02679 [cs.DL].

Poch, M., Bel, N., Espeja, S., & Navío, F. (2014). Ranking Job Offers for Candidates: learning hidden knowledge from Big Data. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (S. 2076–2082). European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2014/pdf/791_Paper.pdf

Potthast, M., Völske, M., & Stein, B. (2013). Crowdsourcing Interaction Logs to Understand Text Reuse from the Web. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (S. 1212–1221). Association for Computational Linguistics. https://aclanthology.org/P13-1119

Radford, W., Hachey, B., Curran, J. R., & Milosavljevic, M. (2009). Tracking Information Flow in Financial Text. In *Proceedings of the Australasian Language Technology Association Workshop 2009* (S. 11–19). https://aclanthology.org/U09-1003

Ramisch, C., & Villavicencio, A. (2022). Computational Treatment of Multiword Expressions. In *The Oxford Handbook of Computational Linguistics* (S. 649–678). Oxford University Press.

Romanello, M., & Hengchen, S. (2021). *Detecting Text Reuse with Passim*. Programming Historian 10. Verfügbar 19. Mai 2023 unter https://doi.org/10.46430/phen0092

Roth, K. S., Wengeler, M., & Ziem, A. (2017). *Handbuch Sprache in Politik und Gesellschaft* (Bd. 19). De Gruyter.

Sabban, A. (2004). Zur Rolle der Phraseme für die Konstitution und Funktion des Textes. Ein Beitrag zum Konzept der textbildenden Potenzen. In *Wortverbindungen - mehr oder weniger fest* (S. 238–261, Bd. 2003). De Gruyter.

Salmi, H., Paju, P., Rantala, H., Nivala, A., Vesanto, A., & Ginter, F. (2021). The Reuse of Texts in Finnish Newspapers and Journals, 1771-1920: A Digital Humanities Perspective. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, *54*(1), 14–28. https://doi.org/10.1080/01615440.2020.1803166

Salmi, H., Rantala, H., Vesanto, A., & Ginter, F. (2019). The Long-Term Reuse of Text in the Finnish Press, 1771-1920. In *Proceedings of the Digital Humanities in the Nordic Countries 4th Conference (DHN 2019)* (S. 394–404, Bd. 2364). CEUR-WS. https://ceur-ws.org/Vol-2364/

Sameen, S., Sharjeel, M., Nawab, R. M. A., Rayson, P., & Muneer, I. (2018). Measuring Short Text Reuse for the Urdu Language. *IEEE Access*, *6*, 7412–7421. https://ieeexplore.ieee.org/document/8118088

Schofield, A., Thompson, L., & Mimno, D. (2017). Quantifying the Effects of Text Duplication on Semantic Models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (S. 2737–2747). Association for Computational Linguistics. https://aclanthology.org/D17-1290

Shang, W., & Underwood, T. (2021). Improving Measures of Text Reuse in English Poetry: A TF–IDF Based Method. In *Diversity, Divergence, Dialogue: 16th International Conference, IConference 2021, Proceedings, Part I* (S. 469–477). Springer-Verlag. https://doi.org/10.1007/978-3-030-71292-1_36

Smith, D. (2019). *Improving Optical Character Recognition and Tracking Reader Annotations in Printed Books by Collating and Transcribing Multiple Exemplars* [National Endowment for the Humanities (NEH), Grant number: HAA-263837-19]. Northeastern University, United States. Verfügbar 17. Mai 2023 unter https://app.dimensions.ai/details/grant/grant.8385506

Smith, D., Cordell, R., & Mullen, A. (2015). Computational Methods for Uncovering Reprinted Texts in Antebellum Newspapers. *American Literary History*, *27*(3), E1–E15.

Smith, D. A., Cordell, R., & Dillon, E. M. (2013). Infectious Texts: Modeling Text Reuse in Nineteenth-Century Newspapers. In *2013 IEEE International Conference on Big Data* (S. 86–94). Institute of Electrical; Electronics Engineers (I triple E). https://ieeexplore.ieee.org/document/6691675

Smith, D. A., Cordell, R., Dillon, E. M., Stramp, N., & Wilkerson, J. (2014). Detecting and Modeling Local Text Reuse. In *IEEE/ACM Joint Conference on Digital Libraries* (S. 183–192). Institute of Electrical; Electronics Engineers (I triple E). https://doi.org/10.1109/JCDL.2014.6970166

Smith, T. F., & Waterman, M. S. (1981). Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, *147*(1), 195–197. https://dornsife.usc.edu/assets/sites/516/docs/papers/msw_papers/msw-042.pdf

Stäcker, T. (2022). Literaturwissenschaft und Bibliothek – Eine Beziehung im digitalen Wandel. In *Digitale Literaturwissenschaft: DFG-Symposion 2017* (S. 679–707). J.B. Metzler.

Stumpf, S. (2015). *Formelhafte (Ir-)Regularitäten: Korpuslinguistische Befunde und sprachtheoretische Überlegungen* [http://library.oapen.org/handle/20.500.12657/27258]. Peter Lang GmbH.

Sugisaki, K., Wiedmer, N., & Hausendorf, H. (2018). Building a Corpus from Handwritten Picture Postcards: Transcription, Annotation and Part-of-Speech Tagging. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (S. 255–259). European Language Resources Association (ELRA). https://aclanthology.org/L18-1038

Tan, P.-N., Steinbach, M., Kumar, V., & Karpatne, A. (2020). *Introduction to Data Mining: Global Edition* (2nd). Pearson Education.

Tiepmar, J., Teichmann, C., Heyer, G., Berti, M., & Crane, G. (2014). A New Implementation for Canonical Text Services. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)* (S. 1–8). Association for Computational Linguistics. https://aclanthology.org/W14-0601

Torget, A. (2023). Mapping Texts: Examining the Effects of OCR Noise on Historical Newspaper Collections. In *Digitised Newspapers - A New Eldorado for Historians?: Reflections on Tools, Methods and Epistemology* (S. 47–66, Bd. 3). De Gruyter Oldenbourg.

Unger, N., Thandra, S., & Goldberg, I. (2016). Elxa: Scalable Privacy-Preserving Plagiarism Detection. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society* (S. 153–164). Association for Computing Machinery. https://doi.org/10.1145/2994620.2994633

Vesanto, A. (2018). *Detecting and Analyzing Text Reuse with BLAST* [Master's Thesis]. University of Turku. https://www.utupub.fi/bitstream/handle/10024/146706/Vesanto_Aleksi_opinnayte.pdf?sequence=1

Vesanto, A., Ginter, F., Salmi, H., Nivala, A., & Salakoski, T. (2017). A System for Identifying and Exploring Text Repetition in Large Historical Document Corpora. In *Proceedings of*

the 21st Nordic Conference on Computational Linguistics (S. 330–333). Association for Computational Linguistics. https://aclanthology.org/W17-0249

Vesanto, A., Nivala, A., Rantala, H., Salakoski, T., Salmi, H., & Ginter, F. (2017). Applying BLAST to Text Reuse Detection in Finnish Newspapers and Journals, 1771-1910. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language* (S. 54–58). Linköping University Electronic Press. https://aclanthology.org/W17-0510

Wahle, J. P., Ruas, T., Kirstein, F., & Gipp, B. (2022). How Large Language Models are Transforming Machine-Paraphrase Plagiarism. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (S. 952–963). Association for Computational Linguistics. https://aclanthology.org/2022.emnlp-main.62

Wender, H., & Peter, R. (1999). Probleme der Wiederverwendung elektronisch gespeicherter Texte. Zwei Fallstudien. In *Computergestützte Text-Edition* (S. 47–60, Bd. 12). Max Niemeyer Verlag.

Wevers, M. (2023). Mining Historical Advertisements in Digitised Newspapers. In *Digitised Newspapers - A New Eldorado for Historians?: Reflections on Tools, Methods and Epistemology* (S. 227–252, Bd. 3). De Gruyter Oldenbourg.

Wilkerson, J., Smith, D., & Stramp, N. (2015). Tracing the Flow of Policy Ideas in Legislatures: A Text Reuse Approach. *American Journal of Political Science*, *59*(4), 943–956. https://onlinelibrary.wiley.com/doi/abs/10.1111/ajps.12175

Wise, M. (1993). String Similarity via Greedy String Tiling and Running Karp-Rabin Matching. *Unpublished Basser Department of Computer Science Report*. Verfügbar 8. Mai 2023 unter https://www.researchgate.net/profile/Michael_Wise/publication/262763983_String_Similarity_via_Greedy_String_Tiling_and_Running_Karp-Rabin_Matching/links/59f03226aca272a2500141f4/String-Similarity-via-Greedy-String-Tiling-and-Running-Karp-Rabin-Matching.pdf

Wood, D. (2015). *Fundamentals of Formulaic Language: An Introduction*. Bloomsbury Academic.

Wray, A. (2013). Formulaic language. *Language Teaching*, *46*(3), 316–334. https://doi.org/10.1017/S0261444813000013

Xu, R., & Wunsch, D. (2005). Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks*, *16*(3), 645–678. https://ieeexplore.ieee.org/document/1427769

Xu, S., Smith, D., Mullen, A., & Cordell, R. (2014). Detecting and Evaluating Local Text Reuse in Social Networks. In *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media* (S. 50–57). Association for Computational Linguistics. https://aclanthology.org/W14-2707

Xu, X., Zheng, Q., Yan, Z., Fan, M., Jia, A., & Liu, T. (2021). *Interpretation-enabled Software Reuse Detection Based on a Multi-Level Birthmark Model*. arXiv: 2103.10126 [cs.SE].

Zhang, M., Jensen, K., Sonniks, S., & Plank, B. (2022). SkillSpan: Hard and Soft Skill Extraction from English Job Postings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Langua-*

ge Technologies (S. 4962–4984). Association for Computational Linguistics. https://aclanthology.org/2022.naacl-main.366

Zhang, M., Jensen, K. N., & Plank, B. (2022). Kompetencer: Fine-grained Skill Classification in Danish Job Postings via Distant Supervision and Transfer Learning. In *Proceedings of the Language Resources and Evaluation Conference (LREC)* (S. 436–447). European Language Resources Association. https://aclanthology.org/2022.lrec-1.46

Zong, C., Xia, R., & Zhang, J. (2021). *Text Data Mining*. Springer.