Universität Leipzig
Institut für Informatik
Studiengang Informatik, B.Sc.

Einseitige Erkennung von Text-Reuse in Ngram-Datenbanken

Bachelorarbeit

Lange, Ferdinand

1. Gutachter: Prof. Dr. Martin Potthast

Datum der Abgabe: 22. März 2022

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Leipzig, 22. März 2022

. . . . . .
Lange, Ferdinand

Zusammenfassung

Text-Reuse-Erkennung findet in der Regel über einen Vergleich des vorliegenden Dokumentes mit einer Menge an Dokumenten aus einem Referenzkorpus statt. Das Verwenden von Korpora, die keine Volltexte enthalten, ist bisher unüblich, auch wenn z.B. mit dem Google Books Ngram Korpus relevante große Korpora dieser Art existieren. In dieser Arbeit wird ein Verfahren mit zwei verschiedenen Ansätzen vorgestellt, welches es ermöglicht, Text-Reuse über einen Referenzkorpus zu erkennen, der keine vollen Dokumente enthält. Das Korpus liefert nur die Uni- bis 5-Gramme auf Wortebene der Dokumente und deren Häufigkeiten über alle Dokumente hinweg. Untersucht werden dabei die Ergebnisqualität der beiden Ansätze und die Relevanz der fünf Ngramebenen im Kontext zu den beiden Ansätze. Außerdem wird betrachtet, wie sich das Ansetzen eines Schwellwertes für eine Mindesthäufigkeit, die Ngramme im Korpus haben müssen, auswirkt auf die Ergebnisse. Die Idee der Methode ist, aus jedem sprachlichen Satz im Text die Ngramme zu extrahieren, nach deren Häufigkeiten im Korpus zu suchen und die Inverse dieser über die Satzlänge zu mitteln. Dadurch ergibt sich ein Wert, welcher bewertet, wie sehr ein Satz auf Text-Reuse hindeutet oder nicht. Die beiden Methoden erzeugen Evaluationsergebnisse von $F_{1|macro} = 0.44$ und $F_{1|macro} = 0.50$, wobei das Einbeziehen von Uni-, Bi- und Trigrammen am wichtigsten ist, um diese zu erreichen. Der Einfluss des Schwellwertes auf die Ergebnisqualität ist verhältnismäßig gering.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Text-Reuse bezeichnet Textstellen in einem Schriftstück, die deren Autor nicht selbst verfasst hat, sondern die aus einem fremden Dokument übernommen wurden. Jedoch beschränkt sich der Begriff des Text-Reuses nicht auf wortwörtliche Kopien von Textstellen, sondern beschreibt auch übernommene Textstellen, in denen Sätze umgestellt oder Begriffe ausgetauscht, eingefügt bzw. entfernt wurden (Potthast, 2012). Die automatische Erkennung von Text-Reuse in Texten ist ein Feld, das durch methodische Vielfalt geprägt ist (Potthast et al., 2014). Ein großer Teil der Veröffentlichungen entstand im Rahmen der PAN-Konferenzen und deren Shared Tasks bis 2015 (Potthast et al., 2014). Eine konkrete Anwendung der Text-Reuse-Erkennung ist das automatische Untersuchen von Dokumenten dahingehend, ob diese Plagiate enthalten oder nicht. Eine gute automatische Text-Reuse-Erkennung macht das Identifizieren von Plagiaten in Texten sehr viel einfacher. Deshalb können wir davon ausgehen, dass diese einen solchen Diebstahl von geistigem Eigentum unattraktiver macht.
Üblicherweise wird bei der Text-Reuse-Erkennung das vorliegende Dokument mit einem Referenzkorpus an Dokumenten verglichen (Potthast et al., 2014). Alle Passagen, die das vorliegende Dokument aus den Dokumenten des Korpus wiederverwendet hat, sollen gefunden werden. Natürlich können nur die wiederverwendeten Textstellen erkannt werden, deren Original auch aus den Dokumenten des Referenzkorpus stammt. Wie viele und welche Dokumente im Referenzkorpus enthalten sind, ist sehr relevant dafür, wie viel Text-Reuse erkannt werden kann oder nicht. Da aber die meisten Dokumente vom Urheberrecht geschützt sind, lassen sich frei verfügbare Korpora mit einer großen Menge relevanter Dokumente wie etwa Büchern kaum finden.
Der in dieser Arbeit vorgestellte Ansatz ist, Text-Reuse nicht über Volltexte zu finden, sondern nur anhand von Häufigkeiten der Ngramme auf Wortebene. Für das zu nutzende Referenzkorpus werden die zugrundeliegenden Dokumente

in Ngramme auf Wortebene zerkleinert und diese Ngramme werden über alle Dokumente hinweg gezählt. Als wesentliche Orientierung dient die Struktur des Google Books Ngram Korpus. Dieses enthält Uni- bis 5-Gramme (Aiden and Michel, 2011). Daher beschränkt sich das in dieser Arbeit verwendete Korpus auch auf die maximale Ngramgröße von fünf.
Um anhand von einem Korpus dieser Form Text-Reuse zu erkennen, stellt diese Arbeit zwei Methoden vor. Es handelt sich dabei um eine einseitige Erkennung von Text-Reuse. Im Referenzkorpus lassen sich aufgrund von dessen Struktur keine Textstellen identifizieren. Deshalb beschränken sich diese Methoden auf das Finden der wiederverwendeten Textstellen im zu untersuchenden Dokument. Beide entscheiden im Dokument für jeden sprachlichen Satz einzeln, ob dieser Text-Reuse darstellt. Die Entscheidung wird anhand von den Ngrammen des Satzes und deren Häufigkeiten im Referenzkorpus getroffen. Der wesentliche Unterschied beider Methoden ist, dass die erste die Häufigkeiten aller Ngramme aus dem Satz einbezieht, und die zweite eine Vorauswahl an Ngrammen trifft, um mehrfache Repräsentationen von den Textstellen zu vermeiden. Die erste hat ein $F_{1|macro}$-Maß von 0.45. Die zweite schneidet mit $F_{1|macro} = 0.50$ etwas besser ab. Der Einfluss der verschiedenen Ngramgrößen wird bei den beiden Verfahren jeweils durch fünf Parameter bestimmt. Diese werden optimiert, sodass sie zu den bestmöglichen Evaluationsergebnissen führen. Es ist wichtig zu betrachten, welchen Einfluss diese Parameter auf die Evaluationsmaße haben, da auch die Relevanz von den Ngrammen der verschiedenen Ngramgrößen im Kontext der beiden Methoden durch die Höhe dieser Parameter repräsentiert wird.
Im Google Books Ngram Korpus wurde ein Schwellwert von 40 angesetzt. Das bedeutet, es sind nur die Ngramme erfasst, welche in mehr als 40 Büchern vorkommen (Aiden and Michel, 2011). Diese Arbeit untersucht, wie sich ein solcher steigender Schwellwert auf die Qualität der Ergebnisse der beiden Text-Reuse-Erkennungsverfahren auswirkt. Dazu werden vergleichbare Schwellwerte bis zu 100 angesetzt, um zu betrachten, wie sehr die Evaluationsmaße darunter leiden.

Kapitel 2

Related Work

2.1 Externe Text-Reuse-Erkennung

Bei den Verfahren zur Text-Reuse-Erkennung ist es populär, ein Referenzkorpus zu nutzen, mit dem das zu untersuchende Dokument abgeglichen wird, um Ähnlichkeiten zu finden und damit Text-Reuse zu klassifizieren. Verwendet ein Verfahren solch ein externes Referenzkorpus, dann wird es als externes Text-Reuse-Erkennungsverfahren bezeichnet. Letzten Endes ist der gewünschte Output des Verfahrens die Menge aller wiederverwendeten Passagen aus dem zu untersuchenden Dokument sowie jeweils die Stelle, von die Passage der wiederverwendet wurde. Diese wird in den Dokumenten des Referenzkorpus gefunden. Damit dies möglich ist, müssen im Referenzkorpus die Dokumente in Form von Volltexten vorliegen.

Text-Alignment. Das Text-Alignment ist dabei der Kern der externen Text-Reuse-Erkennung. Dessen Ziel ist es, im paarweisen Vergleich von zwei Dokumenten allen Text-Reuse zu identifizieren. Dieser wird in Form von Paaren von Textstellen angegeben, eine aus jedem Dokument (Potthast et al., 2014).
Seit 2009 findet jährlich die PAN statt, welche jedes Jahr unter anderem eine Reihe von Shared Tasks umfasst. Bis einschließlich 2015 wurden im Rahmen von PAN mehrere Shared Tasks zu Text-Reuse-Erkennung bzw. Plagiatserkennung gestellt, unter anderem auch zu Text-Alignment. Wie von Potthast et al. (2014) beschreiben, folgen viele der Einreichungen zum Text-Alignment den drei Schritten: Seeding, Extension und Filtering. Beim Seeding werden über eine Heuristik Ähnlichkeiten zwischen den beiden Texten gefunden. Viele Algorithmen suchen nach übereinstimmenden oder sich ähnlichen Ngrammen. Dies geschieht oft auf Wortebene aber auch andere Ähnlichkeitsmaße, wie die Cosinusähnlichkeit im Bezug zu Term-Frequency-Vektoren, werden verwendet. Diese gefundenen Seeds werden bei der Extension auf ganze Textpassagen

Tabelle 2.1: Einreichungen mit höchstem *plagdet*-Score zu Text-Alignment bis einschließlich 2014 bei PAN (Potthast et al., 2014)

Einreichung	**Maße**			
	prec	*rec*	F_1	*plagdet*
Sanchez-Perez et al. (2014)	0.88	0.88	0.88	0.88
Oberreuter and Eiselt (2014)	0.89	0.86	0.87	0.87
Palkovskii and Belov (2014)	0.92	0.83	0.87	0.87
Glinos (2014)	0.96	0.79	0.86	0.86

ausgeweitet, indem Seeds, die nahe im Text beieinanderliegen werden zusammengeclustert werden. Zuletzt werden beim Filtern Passagen aussortiert, die bestimmten Kriterien nicht entsprechen. Zum Beispiel werden zu kurze Passagen entfernt. Alle Paare an Textstellen, die übrig bleiben, klassifiziert der Algorithmus als Text-Reuse. Neben den ngram-basierten Verfahren sind besonders die Methoden erfolgreich, die eine Form des Vektorraummodells nutzen. Bei diesem ist es auch möglich, Part-of-Speech-Tags einzubeziehen (Vani and Gupta, 2016).

Mögliche Ansätze. Die besten Ergebnisse im Rahmen von PAN erreichen Sanchez-Perez et al. (2014). Die Seeds findet deren Methode über ein tf-idf-Vektorraummodell, wobei anstatt von Dokumenten in einer Kollektion Sätze in einem Dokument als Vektoren repräsentiert werden. Ein Paar von Sätzen der beiden betrachteten Dokumente wird zum Seed, wenn die Cosinusähnlichkeit und der Dice-Koeffizient von deren Vektoren über entsprechenden Schwellwerten liegen. Bei der Extension clustert deren Methode Seeds, die nahe beieinanderliegen, rekursiv zusammen, wobei die entstehenden Textfragmente eine Cosinusähnlichkeit haben müssen, die über einem weiteren Schwellwert liegen. Beim Filtering werden Text-Reuse-Fälle, die mit anderen überlappen, sowie Text-Reuse-Fälle deren Größe unter einem spezifischen Wert liegen, entfernt. Ein Vektorraummodell zu nutzen, ist jedoch nur ein Ansatz. Glinos (2014) nutzt im Gegensatz dazu den Smith-Waterman-Algorithmus, um wortwörtlichen Text-Reuse zu identifizieren. Für das Finden von Text-Reuse, welcher Textstellen oder ganze Dokumente zusammenfasst und umschreibt, verwendet Glinos (2014) ngram-basierte Verfahren. Diese beruhen darauf, die häufigsten Wörter des Dokumentes, von dem potenziell Text wiederverwendet wurde, zu extrahieren und zu untersuchen, ob Ngramme aus dem Text, die diese häufigen Wörter enthalten, auch in Textstellen des anderen Dokumentes enthalten sind.

Source Retrieval. Ist bei der klassischen externen Text-Reuse-Erkennung das Referenzkorpus relativ groß, wäre es im Anbetracht der benötigten Ressourcen zu aufwändig, zehntausende oder hunderttausende paarweise Vergleiche im Sinne des Text-Alignments zu machen. Daher ist es nach Potthast et al. (2012) sinnvoll, vor dem Text-Alignment, die Menge der Dokumente mithilfe eines Source Retrievals einzuschränken. Dessen Aufgabe ist es, über eine Suchmaschine alle Dokumente aus dem Referenzkorpus zu extrahieren, von denen das zu untersuchende Dokument Text wiederverwendet. Diese gefundenen Dokumente werden vom Text-Alignment genauer auf Text-Reuse untersucht. Wie zum Text-Alignment wurden auch zum Source Retrieval bei der PAN Shared Tasks gestellt. Durch die Struktur von unserem Referenzkorpus ist Source Retrieval im Rahmen von diesem kaum sinnvoll. Es lassen sich schließlich keine vollen Dokumente extrahieren.

2.2 Intrinsische Text-Reuse-Erkennung

Im Gegensatz zur externen Text-Reuse-Erkennung steht die intrinsische Text-Reuse-Erkennung. Diese verzichtet gänzlich auf ein Referenzkorpus und soll Text-Reuse alleine anhand des vorliegenden Dokumentes identifizieren. Intrinsische Text-Reuse-Erkennung ist einseitig. Bei der externen Text-Reuse-Erkennung sollen sowohl die Textstellen, die wiederverwendet wurden, als auch die Textstellen die diese wiederverwenden, gefunden werden. Die intrinsische Text-Reuse-Erkennung soll und kann nur letzteres finden. Vorrangig geschieht dies durch das Erkennen von Unterschieden im Sprachstil innerhalb des Textes. Merkmale, die signifikant für den Sprachstil sind, müssen quantifiziert werden. Dazu gehören Text-Statistiken auf Buchstabenebene, syntaktische Merkmale auf Satzebene, Part-of-Speech-Merkmale, das Zählen von spezifischen Wörtern sowie die Struktur des Textes (zu Eissen and Stein, 2006). Im Rahmen von PAN wurden Shared Tasks auch zur intrinsischen Text-Reuse-Erkennung gestellt. Zu den Einreichungen in 2012 schreiben Potthast et al. (2011a), dass die Teilnehmer als erstes das Dokument in Abschnitte spezifischer Größe unterteilt hatten. Danach werden in diesen Abschnitten syntaktische Merkmale quantifiziert erhoben. Die Abschnitte werden danach hinsichtlich dieser syntaktischen Merkmale miteinander oder mit dem Rest des Textes verglichen, wodurch entschieden wird, ob die Unterschiede signifikant genug sind, um den Abschnitt als Text-Reuse zu klassifizieren.

Mögliche Ansätze. Oberreuter et al. (2011) unterteilen den Text in Segmente von 400 Wörtern und erzeugen tf-gewichtete Vektoren von allen Wörtern im Abschnitt, sowie einen Vektor für das gesamte Dokument. Zu jedem

Tabelle 2.2: Einreichungen mit höchstem *plagdet*-Score zu intrinsischer Text-Reuse-Erkennung bei PAN 2011 (Potthast et al., 2011a)

Einreichung	**Maße**			
	prec	*rec*	F_1	*plagdet*
Oberreuter et al. (2011)	0.34	0.31	0.32	0.33
Kestemont et al. (2011)	0.43	0.11	0.18	0.17
Akiva (2011)	0.07	0.13	0.09	0.08
Rao et al. (2011)	0.08	0.11	0.09	0.07

Segment wird der Vektor mit dem des gesamten Textes verglichen dahingehend, ob die Wörter aus dem betrachteten Segment im restlichen Dokument sehr viel seltener sind als im Segment selbst. Ist dies der Fall, wird das Segment als Text-Reuse klassifiziert. Kestemont et al. (2011) unterteilen den Text in Segmente von 1000 Wörtern (Potthast et al., 2011a) und zählen in jedem Segment die Häufigkeiten aller darin enthaltenen Trigramme auf Buchstabenebene. Anders als bei der Methode von Oberreuter et al. (2011) wird hier Segment mit Segment verglichen, was ein Ähnlichkeitsmaß für jedes Paar an Segmenten liefert. Letzten Endes nutzt die Methode die Mahalanobisdistanz, um die Ausreißer und damit den potenziellen Text-Reuse zu erkennen. Vergleicht man die Ergebnisse der intrinsischen und externen Text-Reuse-Erkennung bei den Shared Tasks, fällt schnell auf, dass die besten externen Verfahren besser performen als die besten intrinsischen.

2.3 Evaluationsmaße

Potthast et al. (2010) beschreiben Maße, um bei Text-Alignment sinnvoll evaluieren zu können. Diese führen wir im Detail aus, da sich die Evaluationsmaße, die in dieser Arbeit verwentet werden, stark an diesen orientieren. Ein Fall von Text-Reuse wird dort mit $s = \langle s_{plg}, d_{plg}, s_{src}, d_{src} \rangle$ beschrieben. Sei d_{plg} das zu untersuchende Dokument und s_{plg} die in d_{plg} wiederverwendete Passage. s_{plg} verwendet die Passage s_{src} des Dokumentes d_{src} wieder. $\mathbf{s}$ repräsentiert dabei s als Menge von Referenzen auf Buchstaben aus d_{plg} und d_{src}. Wird eine Passage aus d_{plg} durch das zu prüfende Verfahren als Text-Reuse klassifiziert, lässt sich diese als $r = \langle r_{plg}, d_{plg}, r_{src}, d'_{src} \rangle$ definieren. Äquivalent zu $\mathbf{s}$ repräsentiert $\mathbf{r}$ r als Menge von Referenzen auf Buchstaben aus d_{plg} und d'_{src}. r entdeckt s, wenn $\mathbf{s}_{src} \cap \mathbf{r}_{src} = \emptyset$, $\mathbf{s}_{plg} \cap \mathbf{r}_{plg} = \emptyset$ und $d_{src} = d'_{src}$. S ist die Menge aller existierenden Text-Reuse-Fälle und R die Menge aller Passagen, die als Text-Reuse klassifiziert wurden. Die meisten Maße basieren auf den zwei Kennwerten Pre-

cision und Recall. Beide Maße liefern Werte zwischen null und eins. Recall gibt auf der einen Seite an, wie viele der gesuchten Elemente gefunden wurden im Verhältnis zu der Anzahl der Elemente, die hätten gefunden werden müssen. Precision gibt auf der anderen Seite das Verhältnis zwischen allen korrekt gefundenen Elementen zu allen gefundenen Elementen an, also auch den fälschlicherweise gefundenen Elementen. Auf Buchstabenebene definieren Potthast et al. (2010) diese Maße wie folgt:

$$prec_{micro}(S,R) = \frac{|\bigcup_{(s,r)\in(S\times R)}(\mathbf{r}\sqcap\mathbf{s})|}{|\bigcup_{s\in S}\mathbf{s}|}$$

$$rec_{micro}(S,R) = \frac{|\bigcup_{(s,r)\in(S\times R)}(\mathbf{r}\sqcap\mathbf{s})|}{|\bigcup_{r\in R}\mathbf{r}|}$$

wobei $\mathbf{r}\sqcap\mathbf{s} = \begin{cases} \mathbf{r}\cap\mathbf{s} & \text{wenn } r \text{ entdeckt } s, \\ \emptyset & \text{sonst} \end{cases}$.

Das F_1-Maß bezieht sowohl Precision als auch Recall ein, indem es das Harmonische Mittel beider Werte bildet:

$$F_{1|micro} = \frac{2\cdot prec_{micro}(S,R)\cdot rec_{micro}(S,R)}{prec_{micro}(S,R)+rec_{micro}(S,R)}$$

Im Gegensatz zu den Micro-Maßen mitteln die Macro-Maße auf den Text-Reuse-Fällen, sodass jeder Fall unabhängig von seiner Länge gleich viel Einfluss auf das entsprechende Maß hat.

$$prec_{macro}(S,R) = \frac{1}{|R|}\cdot\sum_{r\in R}\frac{|\bigcup_{s\in S}(\mathbf{r}\sqcap\mathbf{s})|}{|\mathbf{r}|}$$

$$rec_{macro}(S,R) = \frac{1}{|S|}\cdot\sum_{s\in S}\frac{|\bigcup_{r\in R}(\mathbf{r}\sqcap\mathbf{s})|}{|\mathbf{s}|}$$

$F_{1|macro}$ wird analog zu $F_{1|micro}$ gebildet. Bei der Erkennung von Text-Reuse ist es nicht erwünscht, dass Fälle s von vielen verschiedenen Funden r getrennt entdeckt werden. (Potthast et al., 2010) Geschieht dies, wirkt es sich jedoch nicht zwingend negativ auf die Precision-, Recall- bzw. F_1-Maße aus. Deswegen beschreiben Potthast et al. (2010) die Granularity und das damit zusammenhängende Plagdet-Maß. Sei $S_R = \{s|s\in S \text{ und } \exists r\in R : r \text{ entdeckt } s\}$ und $\{r|r\in R \text{ und } r \text{ entdeckt } s\}$.

$$gran(S,R) = \frac{1}{|S_R|}\cdot\sum_{s\in S_R}|R_s|$$

Die Granularity besagt somit, wie oft jedes s im Durchschnitt von r entdeckt wurde, solange es mindestens ein r gibt, welches s entdeckt. Im Plagdetmaß werden sowohl die Granularity als auch das entsprechende F_1-Maß berücksichtigt:

$$plagdet(S, R) = \frac{F_1}{log_2(1 + gran(S, R))}$$

Bei PAN stellen Potthast et al. (2014) zusätzlich Maße vor, welche nicht wie die Micro- und Macro-Maße auf Buchstabenebene evaluieren, sondern auf Fallebene oder Dokumentebene. Auf Fallebene wird jeder Fall entweder als korrekt erkannt oder als gar nicht erkannt gewertet. Diese Maße generalisieren somit auf dieser Fallebene. Dafür muss bestimmt werden, ab welchem Grad an Überlappung ein Fall als korrekt gefunden gilt.

$$\begin{aligned} S' = \{s | s \in S \text{ und } rec_{macro}(\{s\}, R) > \tau_1 \text{ und } \exists r \in R : \\ r \text{ entdeckt } s \text{ und } prec_{macro}(S, \{r\}) > \tau_2\} \end{aligned}$$

$$\begin{aligned} R' = \{r | r \in R \text{ und } prec_{macro}(S, \{r\}) > \tau_2 \text{ und } \exists s \in S : \\ r \text{ entdeckt } s \text{ und } rec_{macro}(\{s\}, R) > \tau_1\} \end{aligned}$$

τ_1 ist eine Konstante, welche bestimmt, wie hoch der Recall mindestens sein muss, damit eine Klassifikation von Text-Reuse als korrekt angenommen wird. τ_2 bestimmt die Mindesthöhe der Precision. Bei der Evaluierung von externen Text-Reuse-Erkennungsverfahren im Rahmen der PAN sind $\tau_1 = 0,5$ und $\tau_2 = 0.5$ gewählt. Precision und Recall auf Fallebene sind wie folgt definiert:

$$prec_{case}(S, R) = \frac{R'}{R}$$

$$rec_{case}(S, R) = \frac{S'}{S}$$

Auf Dokumentebene werden die Paare von Dokumenten mit betrachtet $D_{pairs} = D_{plg} \times D_{src}$. Außerdem:

$D_{pairs|S} = \{(d_{plg}, d_{src}) | (d_{plg}, d_{src}) \in D_{pairs} \text{ und } \exists s \in S : d_{plg} \in s \text{ und } d_{src} \in s\}$
$D_{pairs|D} = \{(d_{plg}, d_{src}) | (d_{plg}, d_{src}) \in D_{pairs} \text{ und } \exists r \in R : d_{plg} \in r \text{ und } d_{src} \in r\}$
$D_{pairs|D'} = \{(d_{plg}, d_{src}) | (d_{plg}, d_{src}) \in D_{pairs} \text{ und } \exists r \in R' : d_{plg} \in r \text{ und } d_{src} \in r\}$

Die Evaluationsmaße auf Dokumentebene werden wie folgt definiert:

$$prec_{doc}(S, R) = \frac{|D_{pairs|S} \cap |D_{pairs|R'}|}{|D_{pairs|R}|}$$

$$rec_{doc}(S, R) = \frac{|D_{pairs|S} \cap |D_{pairs|R'}|}{|D_{pairs|S}|}$$

$F_{1|case}$ und $F_{1|doc}$ werden analog zu $F_{1|micro}$ gebildet.

2.4 Evaluationskorpora

Um die Einreichungen für die Shared Tasks zur Text-Reuse-Erkennung zu evaluieren, stellt PAN in jedem Jahr ein Korpus zur Verfügung, auf den die zuvor beschriebenen Evaluationsmaße anwendbar sind. Das Evaluationskorpus, das 2011 veröffentlicht wurde, ist das PAN-PC-11-Korpus. Dieses war die Grundlage für die Evaluation der Shared Tasks im Rahmen von PAN 2011 (Potthast et al., 2011a). Das PAN-PC-11 enthält eine Menge an zu untersuchenden Dokumenten, den Suspicious-Documents, und eine Menge an Referenzdokumenten, den Source-Documents. Diese umfassen jeweils 11093 Texte. (Potthast et al., 2011b) Alle Dokumente stammen aus dem Projekt-Gutenberg-Korpus und sind großteils in englischer Sprache verfasst. Projekt-Gutenberg stellt eine große Menge an Büchern und andere Dokumente bereit, die meist aufgrund von deren Alter nicht mehr durch das Urheberrecht geschützt sind (Projekt Gutenberg, 2022a). In die Source-Documents und in die Suspicious-Documents sind Text-Reuse-Fälle eingefügt, so dass jeder Text-Reuse-Fall in einem der Suspicious-Documents ein Gegenstück in den Source-Documents hat. Zu jedem der Dokumente ist eine XML-Datei beigelegt, in der notiert ist, an welchen Stellen im Text des Dokuments der Text-Reuse eingefügt ist und wo deren Gegenstücke zu finden sind. Die zu testenden Verfahren untersuchen die Suspicious-Documents einzeln auf Text-Reuse, wobei alle Source-Documents als Referenzkorpus genutzt werden. Über die XML-Dateien wird danach mit den bereits beschriebenen Evaluationsmaßen die Qualität der Ergebnisse des Verfahrens ermittelt. Somit bietet das PAN-PC-11-Korpus eine gute Grundlage, um Verfahren zu Text-Reuse-Erkennung zu evaluieren.

2.5 Referenzkorpora

Auch wenn die intrinsische Text-Reuse-Erkennung einen wichtigen Ansatz verfolgt, erzeugen die State-Of-The-Art-Verfahren zur externen Text-Reuse-Erkennung weit bessere Ergebnisse als die guten intrinsischen Methoden. In der Praxis verwenden daher die meisten Implementierungen Referenzkorpora. Die Größe dieses zugrundeliegenden Referenzkorpus ist dabei essentiell für die Sinnhaftigkeit der darauf angewendeten Text-Reuse-Erkennungsverfahren. Schließlich kann nur Text-Reuse erkannt werden, wenn das Dokument, aus dem Text wiederverwendet wurde, auch im Referenzkorpus enthalten ist. Wollen wir beispielsweise testen, ob ein Dokument Text von Internetseiten wiederverwendet hat, dann existiert mit den Clueweb-Korpora eine angemesse Grundlage. Clueweb09 beispielsweise enthält etwa eine Milliarde Webseiten und bildet damit einen großen Teil des Internets ab (The Lemur Project, 2022). Außerdem ist Clueweb09 frei verfügbar. Beim Testen auf Text-Reuse, der aus Büchern

stammt, lässt sich kaum ein Korpus vergleichbarer Größe finden, welcher im Gegensatz zu den Clueweb-Korpora Bücher enthält. Der wesentliche Grund ist natürlich, dass die meisten Bücher durch das Urheberrecht geschützt sind. Somit sind die Korpora, die Bücher enthalten, entweder nicht öffentlich zugänglich oder es werden nur Bücher erfasst, die nicht durch das Urheberrecht geschützt sind. Letzteres ist beim Projekt-Gutenberg-Korpus der Fall. Dieses enthält über 60.000 Bücher und andere Dokumente (Projekt Gutenberg, 2022b). Als universelles Referenzkorpus, um Text-Reuse in Büchern zu bestimmen, ist es jedoch unzureichend. Text-Reuse beschränkt sich schließlich keineswegs auf die nicht urheberrechtlich geschützten Bücher.

Google Books Ngram Korpus Eine wesentliche Motivation für diese Arbeit ist das Google Books Ngram Korpus. Dieses ist erstmals 2009 von Google veröffentlicht worden und repräsentierte 5.195.769 Bücher (Aiden and Michel, 2011). Er wurde bereits mehrmals erweitert. Außerdem ist das Korpus frei verfügbar. Es enthält keine Volltexte, sondern Ngramme auf Wortebene. Aiden and Michel (2011) schreiben, dass das Google Books Ngram Korpus die folgende Struktur hat. Es sind alle Ngramme bis zu einer Wortanzahl von 5 erfasst, die in mindestens 40 von allen erfassten Büchern vorkommen. Zu dem Ngram selbst sind die Jahre aufgelistet, in dem die Bücher erschienen sind, welche das Ngram im Text enthalten. Dazu ist jeweils notiert, in wie vielen Büchern, die in dem Jahr erschienen sind, das Ngram vorkommt, sowie die Anzahl der Vorkommen in allen Büchern aus dem entsprechenden Jahr. Als Vorkommen werden nur Folgen von Wörtern gezählt, die in exakt der Reihenfolge unverändert unmittelbar hintereinander im Text existieren. Seit 2012 werden innerhalb eines angegebenen Ngramms des Korpus nie Grenzen von sprachlichen Sätzen überschritten. Zuvor wurden stattdessen Ngramme ignoriert, welche Seitengrenzen überschreiten (Lin et al., 2012). Lin et al. (2012) fügen 2012 zu den Ngrammen dieser klassischen Form auch Ngramme hinzu, welche Part-of-Speech-Tags (POS-Tags) enthalten. Diese POS-Tags geben die Wortarten von den Wörtern des Ngrams an. Im Korpus von Google werden an diese Wörter teilweise POS-Tags als Annotationen angefügt oder das Wort wird durch einen POS-Tag als Platzhalter ersetzt. Zusätzlich existieren Platzhalter, die den Start oder das Ende eines Satzes repräsentieren. Es existieren auch Platzhalter, die ein beliebiges Wort unabhängig vom POS-Tag repräsentieren. Bei Ngrammen mit bis zu maximal drei Wörtern sind alle Kombinationen von den rohen Wörtern, den Annotierten und den Platzhaltern erfasst. Bei 4-Grammen und 5-Grammen sind keine Kombinationen von annotierten und rohen Wörtern mehr zulässig. Dies betrifft jedoch nicht die Platzhalter. Unabhängig von Ngrammen enthält das Google Books Ngram Korpus auch Darstellungen von Abhängigkeiten im Parsetree. Da diese im Rahmen dieser

Arbeit keine Rolle spielen, wird hier nicht weiter darauf eingegangen. Auch POS-Tags sind kein Teil der Arbeit. Das Evaluationskorpus, das in der Arbeit verwendet wird, enthält nur rohe Ngramme. Das Erfassen von POS-Tags, wie es das Google Books Ngram Korpus tut, würde das Evaluationskorpus sehr aufblähen. Auf der anderen Seite ist auch nicht klar, wie relevant diese POS-Tags im Kontext von der Text-Reuse-Erkennung sind. Die in dieser Arbeit beschriebenen Methoden beschränken sich auf das Betrachten von rohen Ngrammen. Der Grund für die Kreation des Google Books Ngram Korpus hängt schließlich nicht mit Text-Reuse-Erkennung oder Ähnlichem zusammen. Das Korpus soll eine Grundlage schaffen, um kulturelle oder sprachliche Veränderungen quantitativ analysieren zu können (Aiden and Michel, 2011).
Im Rahmen der Text-Reuse-Erkennung wäre es jedoch interessant zu untersuchen, ob es möglich ist, das Google Books Ngram Korpus oder andere Korpora dieses Formats als Referenzkorpus für Text-Reuse-Erkennung zu nutzen. Es existiert schließlich kein von der Größe her vergleichbares Korpus, das Bücher als Volltexte enthält und frei verfügbar ist. Ein solches Ngramkorpus bricht kein Urheberrecht, weshalb dieses auch problemfrei publiziert werden kann. Somit ist es relevant, auch Verfahren zu entwickeln, die anhand von solchen Ngramkorpora Text-Reuse erkennen können. Die Arbeit stellt zwei Verfahren vor und untersucht auch, wie sehr deren Ergebnisqualität darunter leidet, wenn ein Schwellwert an Minimalvorkommen der Ngramme im Korpus angesetzt wird. Dies ist vergleichbar mit dem Schwellwert von 40 im Google Books Ngram Korpus. Ein solcher Schwellwert kann die Größe des Korpus signifikant verringern. Auch kann dadurch sichergestellt werden, dass die Ursprungstexte nicht mithilfe des Korpus rekonstruierbar sind. So können solche Korpora auch durch das Urheberrecht geschützte Texte verwenden, wobei deren Veröffentlichung dieses Urheberrecht nicht verletzt.
Als Evaluationskorpus ist das Korpus von Google kaum verwendbar. Die massive Größe macht ihn sehr unpraktikabel. Es ist nicht klar, welche Bücher im Korpus enthalten sind und welche nicht, daher ist es auch kaum möglich, entsprechend Text-Reuse-Fälle zu identifizieren oder zu kreieren, um letzten Endes eine Evaluation darüber durchführen zu können. Dementsprechend ist das PAN-PC-11-Korpus die bessere Grundlage zur Evaluation.

Kapitel 3

Methodischer Ansatz

Dieses Kapitel stellt die Methode zur Identifikation von Text-Reuse vor und differenziert dabei in zwei verschiedene Ansätze. Der erste Abschnitt geht auf den Hintergrund ein und beschreibt die Annahmen, auf der die Methode fußt. Im zweiten Abschnitt wird die theoretische Grundlage geschaffen, die wesentlich für die Beschreibung der Methode ist. Der dritte Abschnitt beschreibt diese ausführlich. Darauf folgt die Beschreibung der beiden verschiedenen Ansätze mitsamt deren Eigenschaften.

3.1 Hintergrund

Unseren Methoden liegen mehrere Annahmen zugrunde. Die erste ist, dass Text-Reuse meist volle sprachliche Sätze umfasst. Dies bedeutet nicht, dass Text-Reuse-Fälle nicht mehr als einen Satz umfassen können. Vielmehr halten wir Text-Reuse, der nur einen Teil eines Satzes umfasst, für ungewöhnlich. Deshalb betrachten wir jeden Satz in unserem zu untersuchenden Dokument als Einheit, für die entschieden wird, ob sie Text-Reuse darstellt oder nicht. Dies ist ein wesentlicher Unterschied zum klassischen Text-Alignment, in dem es üblich ist, abhängig von gefundenen Ähnlichkeiten zu clustern.
Im Text-Alignment ist es das Ziel, Paare von Textstellen zu finden (Potthast et al., 2014). Bei diesen Paaren können wir nicht davon ausgehen, dass ein Satz so wiederverwendet wird, sodass genau ein neuer entsteht. Sätze können verändert, zusammengefügt oder in mehrere kleinere Sätze aufgeteilt werden. Beim Text-Ailgnment nur Paare von einzelnen Sätzen zu betrachten, wäre naiv. Im Fall von einseitiger Text-Reuse-Erkennung ist das anders. Es werden keine Paare gesucht, sondern nur die wiederverwendeten Textstellen im zu untersuchenden Dokument. Ohne zweites Dokument halten wir flexibles Clustern, wie es beim Text-Alignment üblich ist, nicht für zielführend. Wie in Kapitel 2 beschrieben, ist ein sinnvoller Ansatz im Text-Alignment, Ähn-

lichkeiten zu suchen, welche auch Seeds genannt werden, und wenn mehrere Seeds innerhalb der beider Texte annehmbar nah aneinander liegen, werden sie in beiden Texten zusammengeclustert. Aber auf welcher Grundlage sollten wir bei der Struktur unserer Datenbank clustern? Dementsprechend halten wir das statische Betrachten von vollen Sätzen in diesem Kontext für einen sinnvollen Ansatz. Ein Vektorraummodell vergleichbar zu dem, wie es im Text-Alignment üblich ist, ist im Kontext der Arbeit kaum anwendbar. Es setzt den Vergleich von zwei Dokumenten in Volltextform vorraus. Dieser ist durch die Struktur des Korpus nicht möglich.
Eine weitere Annahme ist, dass ein Satz eher Text-Reuse sei, wenn die Ngramme, welche in diesem Satz enthalten sind, auch im Korpus gefunden werden können. Des Weiteren nehmen wir auch an, dass ein solches gefundenes Ngram, dessen Häufigkeit im Korpus sehr selten ist, eher für Text-Reuse spricht als gefundene Ngramme mit sehr großen Häufigkeiten. Kommt beispielsweise ein 5-Gram viele tausende Male in den Texten vor, ist dies ein recht unspezifischer Textbaustein, an dessen Hand kaum eine Aussage über Text-Reuse getroffen werden kann. Mehrere seltene Funde würden viel mehr auf Text-Reuse hindeuten.

3.2 Theoretische Grundlage

Zuerst muss eine theoretische Grundlage geschaffen werden. Sei dazu S_d die Menge an Sätzen s in einem Dokument d. $s = (w_1, ..., w_t)$ mit $t \in \mathbb{N} \setminus \{0\}$ stellt den Satz als Folge von Wörtern w_i mit $1 \leqslant i \leqslant t$ dar, wobei $t = |s|$ die Anzahl der Wörter im Satz darstellt. Jedes Wort w_i eines Satzes ist eine Menge an Referenzen zu den entsprechenden Buchstaben im Dokument d. Ein Ngram in einem Satz s lässt somit als Vereinigung von n aufeinanderfolgende Wörter repräsentieren: $g_n = \bigcup_{i=x}^{x+n-1} w_i$ mit $1 \leqslant x \leqslant |s| - n + 1$. Sei $t = |s|$, dann sind alle Ngramme der Größe n, welche im Satz s enthalten sind, in $G_{n,s}$ mit

$$G_{n,s} = \begin{cases} \bigcup_{i=1}^{t+n-1} \bigcup_{j=i}^{i+n-1} w_j & \text{wenn } t \geqslant n, \\ \emptyset & \text{sonst.} \end{cases}$$

Sei nun $s = (w_1, ..., w_t)$ ein Satz mit $t = |s|$, $|s| \leqslant n$, $n < 1$ und $1 \leqslant x \leqslant t-n+1$, $n, x \in \mathbb{N} \setminus \{0\}$. Dieser enthält die Ngramme: $g_n = \bigcup_{i=x}^{x+n-1} w_i$ und $g_{n-1}^1 = \bigcup_{i=x}^{x+n-2} w_i$ sowie $g_{n-1}^2 = \bigcup_{i=x+1}^{x+n-1} w_i$, dann gilt $g_{n-1}^1 \subset g_n$ sowie $g_{n-1}^2 \subset g_n$. g_{n-1}^1 und g_{n-1}^2 sind somit enthalten in g_n. Ist in dieser Arbeit davon die Rede, dass Ngramme einander enthalten, dann meint dies diese Teilmengenrelation.
Angenommen ein Satz eines Textes enthält die Textstelle "Annas favourite food is pizza" und dessen Buchstaben wären als 5-Gram referenziert. Die darin enthaltenden Texte "Annas favourite food is" und "favourite food is pizza" würden

dann ebenfalls als 4-Gramme referenziert sein. Da diese beiden 4-Gramme nur die Buchstaben im Text referenzieren, die auch das zuvor beschriebene 5-Gram referenziert, sind sie in diesem 5-Gram enthalten. Natürlich ist diese Relation transitiv. Das 5-Gram enthält nicht nur diese beiden 4-Gramme, sondern auch die Trigramme, die in mindestens einem der beiden 4-Gramme enthalten sind, wie z.B. "Annas favourite food". Angenommen $n > 2$, dann sind die Ngramme $g^3_{n-2} = \bigcup_{i=x}^{x+n-3} w_i$ und $g^4_{n-2} = \bigcup_{i=x+1}^{x+n-1} w_i$ in g^1_{n-1} enthalten und damit auch in g_n.

3.3 Methodik

Für ein zu untersuchendes Dokument d mit der Menge an sprachlichen Sätzen S_d wird jeder Satz $s \in S_d$ auf Text-Reuse untersucht. Für $n, k \in \mathbb{N}$ mit $1 \leqslant n \leqslant k$ ist $G'_{n,s} \subseteq G_{n,s}$ eine gewählte Menge von Ngrammen g_n der Länge n aus dem Satz s. Wir wählen $k = 5$, da das Korpus nur maximal 5-Gramme enthält. $\alpha_1, ..., \alpha_n \in \mathbb{R}$ sind Konstanten. Die Häufigkeit eines Ngrams g_n im Korpus sei mit $h(g_n)$ definiert. Die Funktion f liefert einen Wert, nach dem entschieden wird, ob der Satz s als Text-Reuse klassifiziert wird oder nicht. Je höher $f(s, k)$ desto eher wird s als Text-Reuse ausgewertet.

$$f(s,k) = \frac{\sum_{n=1}^{k} \sum_{g_n \in G'_{n,s}} \frac{\alpha_n}{h(g_n)}}{|s|}$$

f summiert für alle gewählten Ngramme mit $n \leqslant k$ die Inverse der gefundenen Häufigkeit auf, wobei zu jedem n α_n als festes Gewicht fungiert. Eine Annahme war, dass das Vorkommen von Ngrammen mit wenigen Vorkommen im Korpus für Text-Reuse sprechen. Durch das Betrachten der Inverse der Häufigkeiten führen niedrige Häufigkeiten zu größerer Veränderung in f als höhere Häufigkeiten. Des Weiteren soll die Länge eines Satzes kein wesentliches Argument für Text-Reuse darstellen. Da für jeden Satz s mit $|s| \geqslant k$ gilt für alle $n \in \mathbb{N}$ mit $1 \leqslant n \leqslant k : |G_{n,s}| = |s| + n - 1$. Mit sinnvoll gewählten Konstanten $\alpha_1, ..., \alpha_n$ ist f invariant gegenüber $|s|$. Die Entscheidung, für jedes n eine Kontante α_n zu wählen, hat mehrere Gründe. Es ist nicht klar, ob die Ngramme unterschiedlicher n gleich relevant für die Erkennung von Text-Reuse sind. Eine Vermutung wäre, dass Ngramme mit größeren n relevanter sind, da diese, wenn sie gefunden werden, für recht spezifische Übereinstimmung sprechen. Wird jedoch bei Text-Reuse die Textstelle stark verändert, kann es auch sein, dass es keine oder wenige übereinstimmende Ngramme größerer n gibt. Es ist daher nicht intuitiv klar, wie relevant die Häufigkeiten für Ngramme unterschiedlicher n sind. Dazu hängen die Häufigkeiten der Ngramme auch von n ab. Für geringere n werden weniger verschiedene Ngramme gefunden als bei größeren n. Das ist

z.B. auch in Tabelle 4.1 zu erkennen. Dementsprechend sind die Häufigkeiten der Ngramme kleinerer n im Durchschnitt höher als die der größeren n. Weiterführend gilt es, diese hinsichtlich der Ergebnisqualität zu optimieren und zu untersuchen. Sei C_d die Menge aller als Text-Reuse klassifizierten Sätze für ein Dokument d.

$$C_d = \{s | s \in S_d \text{ und } f(s,5) \geqslant 1\}$$

Jeder Satz s wird somit genau dann als Text-Reuse klassifiziert, wenn f mindestens 1 ergibt. Wir beschränken uns auf Uni- bis 5-Gramme und wählen damit $k = 5$. Für die Auswahl der Ngramme g_n aus $G_{n,s}$, welche $G'_{n,s}$ enthält und damit für f relevant sind, stellen wir zwei Ansätze vor.

3.4 Klassischer Ansatz

Der klassische Ansatz bezieht alle Ngramme ein, die im Referenzkorpus gefunden werden. Für einen Satz s mit $n \in \mathbb{N}$ und $1 \leqslant n$ wählt der klassische Ansatz $G'_{n,s}$ wie folgt:

$$G'_{klassisch,n,s} = \{g_n | g_n \in G_{n,s} \text{ und } h(g_n) > 0\}$$

Aufgrund der Teilmengeneigenschaften der Ngramme, welche zu Beginn des Kapitels beschrieben wurden, gilt für alle $n, n' \in \mathbb{N}$ mit $1 \leqslant n < n'$ bei einem Satz s: $\forall g_{n'} \in G'_{klassisch,n',s} \exists g_n \in G'_{klassisch,n,s} : (g_n \subseteq g_{n'})$. Es sind also alle Ngramme aus $G'_{klassisch,n',s}$ auch durch $G'_{klassisch,n,s}$ repräsentiert. Wählen wir $G'_{n,s} = G'_{klassisch,n',s}$ für f, dann nehmen neben $\alpha_{n'}$ auch alle α_n indirekt Einfluss. Die selbe Textstelle wird auf den verschiedenen Ngramebenen in f einbezogen.

3.5 Erweiterter Ansatz

Der erweiterte Ansatz versucht diese Doppelung an Informationen zu vermeiden und damit eine striktere Trennung der $\alpha_1, ..., \alpha_n$ bzw. deren Einflussbereichs zu erreichen. Für einen Satz s mit $n \in \mathbb{N}$ und $1 \leqslant n$ wählt der klassische Ansatz $G'_{n,s}$ wie folgt:

$$\begin{aligned} G'_{erweitert,n,s} = &\{g_n | g_n \in G_{n,s} \text{ und } h(g_n) > 0 \text{ und } \nexists g_{n'} : \\ &(n' > n \text{ und } g_{n'} \in G_{n',s} \text{ und } h(g_{n'}) > 0 \text{ und } g_n \subset g_{n'})\} \end{aligned}$$

$G'_{erweitert,n,s}$ enthält also nur ein Ngram des Satzes, wenn es im Korpus vorkommt und es in keinem anderen im Korpus gefundenen Ngram enthalten ist. Jedes Wort wird so nur auf maximal einer Ngramebene repräsentiert,

und zwar der höchstmöglichen. In f wirkt sich auf jedes Wort nur eine der Konstanten $\alpha_1, ..., \alpha_n$ aus. Eine Ausnahme wären jedoch Überlappungen über Ebenen hinweg. Ist beispielweise ein 5-Gram gefunden worden, das die Wörter w_1, w_2, w_3, w_4 und w_5 repräsentiert, und ein 4-Gram mit den Wörtern w_3, w_4, w_5 und w_6, dann werden drei der Wörter auf beiden Ebenen berücksichtigt, da die beiden Ngramme einander nicht enthalten. Jedoch ist dies nicht als redundante Information anzusehen.
Für die beiden Ansätze werden die Konstanten getrennt optimiert und gesetzt, um so die bestmögliche Qualität der Ergebnisse zu erreichen. Der Nachteil des erweiterten Ansatzes gegenüber dem klassischen ist, das der erweiterte viele Informationen ignoriert. Die Häufigkeiten der ignorierten Ngramme lässt dieser Ansatz gänzlich außer Acht. Wenn ein 5-Gram gefunden wurde, spielt es beim erweiterten Ansatz keine Rolle, ob die darin enthaltenen Uni- und Bigramme ein oder tausende Male im Korpus zu finden sind. Beim klassischen Ansatz ist dies nicht der Fall.

Kapitel 4

Daten und experimentelles Setup

Um die beiden Ansätze umzusetzen, müssen jeweils die Konstanten $\alpha_1, ..., \alpha_n$ sinnvoll gesetzt werden. Die Implementierung lässt uns diese Konstanten ermitteln, die zu den bestmöglichen Ergebnissen führen. Des Weiteren gilt es zu untersuchen, welchen Einfluss die verschiedenen Konstanten bei den verschiedenen Ansätzen auf die Ergebnisqualität haben. Wie sich ein Schwellwert für die Mindestanzahl der Vorkommen im Korpus auswirkt, betrachtet diese Arbeit auch. Beim Testen von Methoden zur Text-Reuse-Erkennung anhand eines Korpus sind unabhängig von den Methoden drei Aufgaben zu bewerkstelligen: der Aufbau des Referenzkorpus, die Durchführung der beschriebenen Methoden und die Evaluation der Ergebnisse. Dies geschieht über vier getrennte Javaanwendungen. Die erste baut das Korpus auf und macht es über eine Datenbank zugänglich. Die zweite und dritte führen die beschriebenen Methoden zur Erkennung von Text-Reuse bei einer beliebigen Menge an Dokumenten durch, wobei die dritte Anwendung einen Parameterdurchlauf über die Konstanten ermöglicht. So wird nach der Evaluation die Optimierung der Konstanten hinsichtlich der Ergebnisqualität möglich. Die letzte Anwendung evaluiert die zuvor kreierten Ergebnisse mit den gewählten Evaluationsmaßen[1].

4.1 Korpuskonstruktion

Eine wesentliche Motivation für die Erstellung der Methoden ist die Existenz des Google Books Ngram Korpus. Dieser lässt sich jedoch als Referenzkorpus, um die Methoden zu testen, kaum nutzen. Um Methoden mitsamt deren Konstanten evaluieren zu können müssen, wenn möglich, mehrere hundert Dokumente testweise auf Text-Reuse untersucht werden, im besten Fall auch mehr.

[1]Der Quellcode zu den Anwendungen ist zugänglich auf `https://git.webis.de/code-teaching/theses/thesis-lange`.

Um die Konstanten $\alpha_1, ..., \alpha_n$ sinnvoll festsetzen zu können, muss dies sehr oft geschehen. Das Korpus von Google ist jedoch sehr groß und dementsprechend ungeeignet, um eine hohe Anzahl an Testergebnisse zu kreieren. Dadurch dass das Google Books Ngram Korpus nur Ngramme erfasst, die in mindestens 40 Büchern vorkommen (Aiden and Michel, 2011), wäre es außerdem nicht möglich zu betrachten, wie unsere Verfahren ohne einen solchen Schwellwert performen. Deswegen wird ein Korpus zur Evaluation genutzt, das vom Google Books Ngram Korpus unabhängig ist. Die Grundlage von diesem ist das PAN-PC-11-Korpus. Anhand dieses Kopus wurden im Rahmen der PAN-Konferenzen viele Algorithmen zur Text-Reuse-Erkennung evaluiert. Da dieses Korpus alle Stellen von Text-Reuse genau und einheitlich referenziert, ist es optimal, um unsere Verfahren zu testen. Außerdem enthält es Bücher, wie auch das Korpus von Google. Diese stammen aus dem Projekt-Gutenberg-Korpus. Der Aufbau des Referenzkorpus soll dem von Google weitestgehend ähneln. Beide enthalten Uni-, Bi-, Tri-, 4- und 5-Gramme. Es findet kein Stemming statt und Stoppwörter werden nicht ignoriert. Alle Wörter in einem Ngram existieren somit in genau dieser Form und in dieser Reihenfolge hintereinander im Text. Wie auch bei dem Google Books Ngram Korpus werden keine Ngramme erfasst, die über Satzgrenzen hinweg gehen. Einige Eigenschaften vom Googles Korpus wurden nicht übernommen. Dieses enthält auch Ngramme mit Wörtern, die mit Part-of-Speech-Tags bzw. POS-Tags annotiert wurden. Auch können Ngramme einen POS-Tag anstatt eines Wortes enthalten, der somit als Platzhalter für beliebige Wörter mit dem POS-Tag steht. Beides ist im Rahmen von dem für die Arbeit erstellten Korpus nicht erfasst worden. Schließlich werden solche POS-Tags bei den zu testenden Methoden nicht betrachtet. Neben den POS-Tags enthält Googles Korpus auch Abhängigkeiten des Abhängigkeits-Parsetrees der einzelnen Sätze. Diese sind in unserem Korpus ebenfalls nicht enthalten. Des Weiteren zählt Google die Ngramme von Büchern aus jedem Jahr getrennt. Dies ist wichtig, um anhand dieser Häufigkeiten Trends in Kultur und Sprache analysieren zu können. Um Text-Reuse zu erkennen, ist dies aber nicht wesentlich. Die Erscheinungsjahre der Dokumente im PAN-PC-11-Korpus stehen in diesem außerdem nicht einheitlich zur Verfügung. Daher wird in dem kreierten Korpus keine Unterteilung nach Erscheinungsjahr gemacht.
Um einen effizienten Zugriff auf die Ngramme des Korpus und deren Häufigkeiten zu gewährleisten, werden diese in einer MongoDB-Datenbank verwaltet. Dabei sind die Uni-, Bi-, Tri-, 4- und 5-Gramme in jeweils einer Kollektion untergebracht. Wir hatten uns gegen eine relationale Datenbank entschieden, da im Rahmen unsere Aufgabenstellung nur Häufigkeiten zu spezifischen Ngrameinträgen gefunden werden müssen. Aufwändigere Anfragen, wie sie bei relationalen Datenbanken üblich sind, bieten hier kaum einen Mehrwert.

Die Javaanwendung, die das Korpus erstellt, liest nacheinander alle Source-Documents des PAN-PC-11-Korpus ein und extrahiert alle Ngramme. Dazu ist es vonnöten, die sprachlichen Sätze zu identifizieren und die Wörter, die in diesen enthalten sind, voneinander zu trennen. Um dies zu bewerkstelligen, nutzt die Anwendung das Stanford CoreNLP. Dieses bietet eine zuverlässige javabasierte Pipeline für Natural Language Processing. Die Anwendung beschränkt sich jedoch auf die Schritte des Identifizierens der Wörter und des Aufteilens in Sätzen. Im CoreNLP heißen diese *tokenize* und *ssplit*. Danach werden Satz für Satz alle darin enthaltenen Ngramme identifiziert und in einer Hashmap gezählt. Erreichen diese Hashmaps eine gewisse Größe, werden die Kollektionen der MongoDB-Datenbank mit den Einträgen aus den Hashmaps aktualisiert. Die Hashmaps werden geleert und die nächsten Source-Documents eingelesen. Bis die Vorkommen aller Ngramme von allen Source-Documents in die Datenbank eingeflossen sind, wird dies fortgeführt. Diese Methode wählten wir, um die Anzahl der Aktualisierungsoperationen auf den Einträgen in den Kollektionen der MongoDB-Datenbank zu minimieren. Wie in Tabelle 4.1 zu sehen ist, haben diese Kollektionen mehrere Millionen Einträge. Trotz Index auf den Strings der Ngramme wird dies mit einer hohen Menge an Anfragen laufzeittechnisch fordernd. Würde nach jedem Vorkommen eines Ngrammes direkt die entsprechende Kollektion aktualisiert werden, wäre die nötige Anzahl der Operationen auf der Datenbank wesentlich höher, um die Ngramme alle Source-Documents hinzuzufügen. Weil in den Hashmaps die Ngramme schon vorläufig gezählt werden, verursachen häufige Ngramme nicht entsprechend viele Operationen auf der Datenbank, sondern nur maximal eine pro Hashmap. Dabei beschränkt sich die Anwendung bei den Source-Documents auf englischsprachige, um die Kollektionen einsprachig zu halten. Da wenige Source-Documents in anderen Sprachen verfasst worden sind, nehmen wir an, dass dies kaum Relevanz hat. Neben der Gesamthäufigkeit in den Dokumenten zu jedem Ngram in der Datenbank wird auch die Anzahl an Dokumenten gespeichert, in denen es mindestens einmal vorkommt. Die Suspicious-Documents aus dem PAN-PC-11-Korpus betrachtet die Anwendung zur Korpuskonstruktion nicht.

4.2 Implementierung der Methoden

Beim Testen der Methoden gilt es einerseits auf alle Suspicious-Documents, die im letzten Kapitel beschriebenen Methoden anzuwenden und die dadurch generierten potenziellen Text-Reuse-Fälle einheitlich zu speichern, sodass der Evaluator diese mit den realen vorliegenden Text-Reuse-Fälle vergleichen kann. So werden die Methoden evaluiert. Andererseits müssen für beide Methoden die

Tabelle 4.1: Anzahl von Einträgen in der MongoDB-Datenbank

Kollektion	Anzahl Einträge
Unigramme	1.521.411
Bigramme	28.716.597
Trigramme	128.432.995
4-Gramme	261.625.105
5-Gramme	346.438.094

Konstanten $\alpha_1, ..., \alpha_n$ gesetzt werden, wobei $n = 5$. Schließlich betrachtet das Referenzkorpus maximal 5-Gramme. Um sinnvolle Werte für die Konstanten zu finden, wird im Rahmen von Rastersuchen über alle fünf Parameter getestet, welche Konstanten zu besseren und welche zu eher schlechteren Evaluationsergebnissen führen. Das heißt, dass die gleichen Methoden auf die gleichen Dokumente mehrere hundert Male durchgeführt werden müssen mit unterschiedlichen Konstanten. Beide Methoden umfassen das Suchen der Ngramme, die in den Sätzen des Dokuments gefunden wurden, in der entsprechenden Kollektion der Datenbank um deren Häufigkeiten zu erhalten. Dies ist der laufzeittechnisch aufwändigste Part bei der Durchführung der Methoden. Es muss schließlich nach allen Ngrammen aller 11093 Dokumente in den Kollektionen der Datenbank gesucht werden. Dies mehrere, gar hunderte oder tausende Male durchzuführen, um für die Konstanten viele Werte testen zu können, ist von der Laufzeit her nicht vertretbar. Deshalb wird die Extraktion der Ngramme und die Durchführung der Rastersuche getrennt in zwei getrennten Anwendungen durchgeführt.
Dies funktioniert auf folgender Grundlage: Sei s ein Satz.

$$\begin{aligned} f(s,5) &= \frac{\sum_{n=1}^{5} \sum_{g_n \in G'_{n,s}} \frac{\alpha_n}{h(g_n)}}{|s|} \\ &= \frac{\sum_{n=1}^{5} (\alpha_n \sum_{g_n \in G'_{n,s}} \frac{1}{h(g_n)})}{|s|} \\ &= \sum_{n=1}^{5} (\alpha_n \frac{\sum_{g_n \in G'_{n,s}} \frac{1}{h(g_n)}}{|s|}) \end{aligned}$$

Somit ist $g(s,n) = \frac{\sum_{g_n \in G'_{n|s}} \frac{1}{h(g_n)}}{|s|}$ unabhängig von den gewählten Konstanten $\alpha_1, ..., \alpha_n$ und muss für unterschiedliche Konstanten nicht jedes Mal neu berechnet werden. Dies lässt sich dementsprechend aus dem Parameterdurchlauf über die α_1 bis α_n auskoppeln. Diese ausgekoppelte Berechnung ist in einer vom Parameterdurchlauf separierten Javaanwendung implementiert. Diese berechnet

$g(s, n)$ für alle Sätze aller Suspicious-Documents aus dem PAN-PC-11-Korpus für alle $n \in \mathbb{N}$ mit $1 \leqslant n \leqslant 5$. Diese Anwendung liest nacheinander die Dokumente ein und extrahiert alle Ngramme aus jedem Satz des Textes analog zur Ngramextraktion, wie sie auch bei der Koruskonstruktion verwendet wurde. Für jeden Satz s werden die Häufigkeiten der Ngramme im Korpus aus den Kollektionen der MongoDB-Datenbank abgefragt. Danach bildet die Anwendung für alle $n \in \mathbb{N}$ mit $1 \leqslant n \leqslant 5$ $G'_{klassisch,n,s}$ bzw. $G'_{erweitert,n,s}$ und berechnet die $g(s, n)$ für die Sätze. Diese werden als Zwischenergebnisse gespeichert. Zu jedem Dokument wird für beide Methoden jeweils eine Datei im CSV-Format angelegt. Jede Zeile repräsentiert einen Satz und enthält einerseits dessen Position im Text und die Länge des Satzes auf Buchstabenebe, um diesen genau identifizieren zu können. Andererseits sind dazu $g(s, 1)$ bis $g(s, 5)$ notiert. Die endgültige Berechnung von den potenziellen Text-Reuse-Fällen C_d ist in einer weiteren Javaanwendung implementiert. Diese implementiert dazu auch den Parameterdurchlauf über die Konstanten $\alpha_1, ..., \alpha_5$. In einer Configdatei wird festgelegt, welche Parameter getestet und welcher der beiden Ansätze verwendet werden soll, um C_d für die Dokumente zu berechnen. Für jedes der α_n mit $n \in \mathbb{N}$ und $1 \leqslant n \leqslant 5$ wird in der Configdatei jeweils ein Startparameter $p_{start,n}$, ein Endparameter $p_{end,n}$ und eine Schrittgröße $p_{step,n}$ definiert, insgesamt also 15 Parameter. Eine Testkonfiguration der Parameter kann als Tupel $t = (\alpha_5, \alpha_4, \alpha_3, \alpha_2, \alpha_1)$ betrachtet werden. Dann sei die Menge an Testkonfigurationen

$$\begin{aligned} T = \{&(\alpha_5, \alpha_4, \alpha_3, \alpha_2, \alpha_1) \in \mathbb{Q}^5 | n, i \in \mathbb{N}_0 \text{ und } 1 \leqslant n \leqslant 5 \\ &\text{und } p_{start,n} \leqslant \alpha_n \leqslant p_{end,n} \text{ und } \alpha_n = p_{start,n} + p_{step,n} \cdot i\} \end{aligned}$$

Mit jedem $t \in T$ werden für alle Suspicious-Documents d die potenziellen Text-Reuse-Fälle C_d mit dem gewählten Ansatz ermittelt. Die Anwendung erstellt für jede der Testkonfigurationen einen Ordner mit Dateien für Ergebnisse aller Suspicious-Documents. Wird ein Satz als Text-Reuse klassifiziert, schreibt die Anwendung dessen Startposition im Text sowie dessen Länge, welche in den Zwischenergebnissen notiert waren, in die entsprechende Datei als Ergebnis. Diese Ergebnisse können evaluiert werden.

4.3 Evaluator

Die Maße, die in dieser Arbeit zur Evaluation verwendet werden, sind stark an die Evaluationsmaße angelehnt, die auch verwendet werden, um Verfahren zur Text-Reuse-Erkennung im Rahmen der Shared Tasks von den PAN-Konferenzen zu bewerten. Diese Maße wurden bereits im Abschnitt in Kapitel 2 beschrieben. Dort wurde zu einem zu untersuchenden Dokument d_{plg} ein

existierender Text-Reuse-Fall als $s = \langle s_{plg}, d_{plg}, s_{src}, d_{src} \rangle$ definiert und ein eine Passage, die vom Verfahren geliefert wird, als $r = \langle r_{plg}, d_{plg}, r_{src}, d'_{src} \rangle$. Die Verfahren müssen somit sowohl die Passage aus dem zu untersuchenden Dokument r_{plg}, als auch die Passage, von der wiederverwendet wurde, r_{src} angeben. Da sich aus dem Korpus jedoch keine vollen Dokumente extrahieren lassen, ist eine Identifikation von d'_{src} und r_{src} mithilfe von nur diesem Korpus nicht möglich. Sowohl der klassische Ansatz als auch der erweiterte Ansatz liefern zu jedem zu untersuchenden Dokument d_{plg} nur alle potenziell wiederverwendeten Textstellen r_{plg}. Dementsprechend definieren wir die existierenden Text-Reuse-Fälle als $s = \langle s_{plg}, d_{plg} \rangle$ und die Textstellen, die unsere Verfahren liefern, als $r = \langle r_{plg}, d_{plg} \rangle$. $\mathbf{r}$ stellt somit r als eine Menge von Referenzen auf die Buchstaben der Textstelle r_{plg} in d_{plg} dar. Genauso stellt $\mathbf{s}$ s als Menge von Referenzen auf Buchstaben von s_{plg} in d_{plg} dar. Entsprechend der Definition im Rahmen der PAN-Konferenz wird bei d_{plg} s von r entdeckt, wenn gilt: $\mathbf{s}_{plg} \cap \mathbf{r}_{plg} = \emptyset$. Die Maße auf Micro- und Makroebene werden äquivalent zu den Maßen gebildet, die bereits in Kapitel 2 beschrieben werden.
Auf die Granularity sowie das damit zusammenhängende Plagdet-Maß verzichten wir im Rahmen unserer Evaluation. Diese Entscheidung hängt damit zusammen, dass die Verfahren, die in dieser Arbeit evaluiert werden, jeden Satz getrennt betrachten und gar nicht die Aufgabe haben, diese Sätze sinnvoll zu größeren Textbausteinen zusammenzuclustern. Enthält z.B. ein Text-Reuse-Fall $s \in S$ viele Sätze und diese werden von den Verfahren entdeckt, dann ist die Granularity trotzdem sehr hoch, da jeder gefundene Satz als eigenes Cluster identifiziert werden würde. Natürlich gäbe es naive Heuristiken, nach denen geclustert werden könnte. Beispielsweise könnten als Text-Reuse klassifizierte Sätze zusammengeclustert werden, wenn diese unmittelbar nebeneinander liegen oder wenn deren Abstand unter einem gewissen Schwellwert liegt. Innerhalb dieser Arbeit wird darauf verzichtet. Der Fokus soll darauf liegen zu beurteilen, wie gut die Verfahren anhand der Ngramhäufigkeiten im Korpus entscheiden können, ob ein Satz wiederverwendet wurde oder nicht. Die Evaluation von Heuristiken und Verfahren zum Clustern in diesem Kontext ist dabei nicht Teil dieser Arbeit.
R', S' und auch die Maße auf Fallebene werden äquivalent zu den Beschreibungen in Kapitel 2 gebildet. Wie auch bei der Evaluation im Rahmen von PAN wählen wir die Konstanten $\tau_1 = 0.5$ und $\tau_2 = 0.5$. Auf Dokumentebene lassen sich die in Kapitel 2 beschriebenen Maße nicht eins-zu-eins übernehmen, da diese darauf basieren, Paare an Dokumenten d_{plg} und d_{src} zu betrachten. Die d_{src} liegen uns nicht vor, daher müssen wir uns auf die Menge der zu untersuchenden Dokumente D beschränken.

$$D_S = \{d_{plg} \in D \text{ und } \exists s \in S : d_{plg} \in s\}$$
$$D_R = \{d_{plg} \in D \text{ und } \exists r \in R : d_{plg} \in r\}$$
$$D_{R'} = \{d_{plg} \in D \text{ und } \exists r \in R' : d_{plg} \in r\}$$

Darauf aufbauend sind die Maße auf Dokumentebene wie folgt definiert:

$$prec_{doc}(S, R) = \frac{|D_S \cap D_{R'}|}{|D_R|}$$

$$rec_{doc}(S, R) = \frac{|D_S \cap D_{R'}|}{|D_S|}$$

$$F_{1|doc} = \frac{2 \cdot prec_{doc}(S, R) \cdot rec_{doc}(S, R)}{prec_{doc}(S, R) + rec_{doc}(S, R)}$$

Die Maße auf Dokumentebene generalisieren noch mehr als auf Fallebene. Schließlich werden nur die Anzahl der Dokumente berücksichtigt, in denen Text-Reuse korrekt entdeckt wurde entsprechende der Definition von R'. Wie viele Text-Reuse-Fälle genau existieren in den Dokumenten, wie groß diese sind und wie viele davon korrekt entdeckt wurden, spielt kaum eine Rolle bei diesen Maßen auf Dokumentebene.

4.4 Bestimmung der Konstanten

Um die Konstanten α_1 bis α_5 für die beiden Ansätze jeweils zu bestimmen, baut die erste Anwendung die Datenbank auf. Danach erzeugt die zweite einmalig die Zwischenergebnisse beider Ansätze für alle Suspicious-Documents des PAN-PC-11. Nun wählen wir in der Configdatei der dritten Anwendung $p_{start,n}$, $p_{end,n}$ und $p_{step,n}$ für jeweils jeden Parameter, wodurch alle Testkonfigurationen definiert werden und die Methode, die betrachtet wird. Für alle diese Testkonfigurationen erzeugt die dritte Anwendung, die den Parameterdurchlauf enthält, die Ergebnisse. Danach berechnet der Evaluator für jede Testkonfiguration alle zwölf Evaluationsmaße. Wir wählen ein Maß, auf deren Grundlage wir die Parameter beurteilen wollen und betrachten die Konstanten der Testkonfiguration, dessen Ergebnisse im gewählten Maß zum höchsten Wert führen. Dies ist ein Parameterdurchlauf über fünf Parameter. Selbst eine mittlere Anzahl an Schritten für jeden Parameter erzeugt eine sehr große Anzahl an Testkonfigurationen. Dies führt bei der Anwendung, die den Parameterduchlauf enthält, schnell zu Laufzeiten, die nicht mehr zu rechtfertigen sind. Dementsprechend wählen wir die Parameter in der Configdatei so, dass

die Anzahl der Testkonfigurationen weit unter tausend bleibt und lassen zu diesen Ergebnisse erstellen. Daraufhin werden diese evaluiert. Daraufhin verfeinern wir die Parameter in der Configdatei abhängig von der besten Testkonfiguration und lassen dafür Ergebnisse generieren. Dies wird fortgeführt, bis sich einem Optimum hinreichend angenähert wurde und die Veränderungen der Evaluationsergebnisse nicht mehr signifikant sind.
Als Evaluationsmaß, nach dem optimiert wird, sind die F_1-Maße am sinnvollsten. Sehr hohe Recallwerte ließen sich dadurch erreichen, wenn die α_1 bis α_5 sehr hoch gewählt werden würden. Mehr Sätze würden als Text-Reuse klassifiziert werden und der Recall steigt. Jedoch geht dies schnell auf Kosten der Precision, da dadurch potenziell auch mehr Sätze fälschlicherweise als Text-Reuse klassifiziert werden. Umgekehrt würden niedrige Kostanten eher zu einem Anstieg der Precision führen. Über die F_1-Maße lassen sich sowohl Precision als auch Recall einbeziehen. Diese könnten auch problematisch sein in dem Fall, wenn gute F_1-Werte zu einem starken Ungleichgewicht zwischen Recall und Precision führen. Bei unseren Ergebnissen ist dies jedoch nicht der Fall. Sehr selten ist Precision mehr als doppelt so hoch wie Recall oder umgekehrt, wenn nach F_1-Maßen optimiert wird.

Kapitel 5

Ergebnisse

Dieses Kapitel beschreibt die Ergebnisse und führt die Evaluation durch. Im ersten Abschnitt werden die Ergebnisse beider Verfahren vorgestellt, wenn kein Schwellwert existiert. Der zweite Abschnitt konzentriert sich auf die Konstanten $\alpha_1, ..., \alpha_5$ und untersucht, wie sich diese auf die Ergebnisqualität beider Ansätze auswirken. Als Letztes wird der Einfluss eines Schwellwertes auf die Ergebnisse betrachtet.

5.1 Ansätze

Für sowohl den klassischen als auch den erweiterten Ansatz wählen wir $\alpha_1, ..., \alpha_5$ so, dass $F_{1|macro}$, $F_{1|micro}$ oder $F_{1|case}$ weitestgehend maximiert ist. Die Ergebnisse lassen sich in den Tabellen 5.1, 5.2 und 5.3 ablesen. $F_{1|doc}$ betrachten wir nicht, da dieses hohe Maß an Generalisierung im Rahmen dieser Arbeit keinen Mehrwert bietet. Der erweiterte Ansatz erzeugt bei allen Maßen bessere Ergebnisse als der klassische. Bei beiden Ansätzen führt diese nach F_1-Maßen optimierte Konfiguration zu leicht recallorientierten Ergebnissen, die Recallwerte sind schließlich etwas höher als die der Precision. Dies sehen wir jedoch nicht als problematisch an, da bei der Optimierung nach $F_{1|macro}$ mit $prec_{macro} = 0.36$ und $prec_{macro} = 0.40$ die Precision annehmbar hoch ist.
Sowohl die optimierten Evaluationswerte bei den verschiedenen Maßen als auch die Konfigurationen der Konstanten weisen kaum massive Unterschiede auf. Am auffälligsten ist, dass beim Optimieren nach $F_{1|micro}$ für beide Ansätze $\alpha_5 = 0$ ist, im Gegensatz zu den Optimierungen nach $F_{1|macro}$ und $F_{1|case}$. Die Evaluationsmaße, die wir primär zu Evaluation verwenden sind die Macro-Maße, da diese nicht generalisieren, wie die fallorientierten Maße oder die dokumentorientieren. Außerdem betrachten die Macro-Maße jeden Text-Reuse-Fall gleichermaßen. Das heißt, dass Text-Reuse-Fälle, die viel Text umfassen, nicht mehr Einfluss aufs Maß haben als Text-Reuse-Fälle, die kürzer sind. Die Micro-Maße würden dies nicht bieten.

Tabelle 5.1: Optimiert nach $F_{1|macro}$

Ansatz	**Maße**			**Konstanten**				
	$prec_{macro}$	rec_{macro}	$F_{1\|macro}$	α_5	α_4	α_3	α_2	α_1
klassisch	0.36	0.60	0.45	2	0	0	16	250
erweitert	0.40	0.67	0.50	2.5	0	2	15	100000000

Tabelle 5.2: Optimiert nach $F_{1|micro}$

Ansatz	**Maße**			**Konstanten**				
	$prec_{micro}$	rec_{micro}	$F_{1\|micro}$	α_5	α_4	α_3	α_2	α_1
klassisch	0.40	0.58	0.47	0	0	0.2	18	160
erweitert	0.41	0.68	0.51	0	0	3	14	100000000

Tabelle 5.3: Optimiert nach $F_{1|case}$

Ansatz	**Maße**			**Konstanten**				
	$prec_{case}$	rec_{case}	$F_{1\|case}$	α_5	α_4	α_3	α_2	α_1
klassisch	0.31	0.80	0.44	1	0	1	20	300
erweitert	0.37	0.80	0.51	3	0	3	12	100000000

5.2 Einfluss der Konstanten

Die Wahl der Konstanten ist essenziell für die Qualität der Ergebnisse. Indirekt lässt sich darüber ablesen, wie wichtig Unigramme, Bigramme, Trigramme, 4-Gramme und 5-Gramme sind, um mit den Ansätzen gute Ergebnisse zu erzeugen. Beim klassischen Ansatz fällt auf, dass α_1 mit 250 bei der Optimierung nach $F_{1|macro}$ verhältnismäßig hoch ist. Jedoch ist dabei zu beachten, dass die Verteilung der Häufigkeiten von Ngrammen im Korpus nicht auf jeder der fünf Ebenen gleich ist. Die Kollektion der 5-Gramme enthält etwa 270-mal so viele Einträge wie die der Unigramme, wie in Tabelle 4.1 zu erkennen ist. Es werden in Texten jedoch nicht mehr 5-Gramme gezählt als Unigramme. Bei jedem Satz s mit $|s| \geqslant 5$, der in den Source-Documents enthalten ist, wurden $|s|$ Unigramme, $|s| - 1$ Bigramme, $|s| - 2$ Trigramme, $|s| - 3$ 4-Gramme und $|s| - 4$ 5-Gramme gezählt und in der Datenbank erfasst. Somit sind die Häufigkeiten der Unigramme im Durchschnitt über 270-mal höher als die der 5-Gramme. In diesem Kontext ist der Faktor $\alpha_1 = 250$ nicht besonders hoch. Beim Betrachten der Bigramme fällt auf, dass die Kollektion der 5-Gramme nur etwa die zwölffache Menge an Einträgen enthält gegenüber der Kollektion der Bigramme. Dementsprechend deutet $\alpha_2 = 16$ auf eine gewisse Relevanz der

Bigramme für den klassischen Ansatz hin. Trigramme und 4-Gramme liefern wiederum keinen Mehrwert beim klassischen Ansatz, da $\alpha_3 = 0$ und $\alpha_4 = 0$. α_5 mit 2 relativ hoch ist.
Während beim klassischen Ansatz alle Ngramme einbezogen werden, solange sie im Korpus gefunden werden können, bezieht der erweiterte nur eine Teilmenge dieser Ngramme ein. Dementsprechend ist es nicht verwunderlich, dass die meisten Konstanten der optimierten Konfiguration des erweiterten Ansatzes höher sind als die der optimierten Konfiguration des klassischen. Beispielsweise wird ein Unigram nur einbezogen, wenn die darüberliegenden Bigramme, die das Unigram enthalten, gar nicht im Korpus gefunden werden können. Die Menge der Unigramme, die übrig bleibt, ist spezifischer und damit potenziell relevanter für die Text-Reuse-Erkennung als die Menge, welche im klassischen einbezogen wird. Interessanterweise führt das Wählen von hohen α_1 in der Regel zu einem Anstieg von $F_{1|macro}$, jedoch flacht der Anstieg bei höheren α_1 stark ab. Dies ist ein wesentlicher Unterschied zum klassischen Ansatz, bei dem bei $\alpha_1 = 250$ $F_{1|macro}$ maximiert ist. Das optimierte α_2 ist jedoch geringer als das optimierte α_2 des klassischen Ansatzes. Trigramme werden beim erweiterten Ansatz mit einbezogen, da $\alpha_3 = 2$ ist. Beim klassischen ist dies nicht der Fall. Das Nichteinbeziehen von 4-Grammen hat im erweiterten Ansatz eine andere Relevanz als im klassischen, da im klassischen zu den ignorierten 4-Grammen mindestens Uni- und Bigramme gibt, die die gleichen Textstellen referenzieren, im erweiterten jedoch nicht. Alle Stellen im Text, die auf der 4-Gramebene abgebildet werden, haben keinen Einfluss. Dennoch wird $F_{1|macro}$ dadurch maximiert. Die Gründe dafür, dass die Relevanz der höherwertigen Ngramme so gering ist, sind nicht klar. Verwendet ein Autor einen Text eines anderen Autors wieder, dann kopiert er ihn oft nicht eins-zu-eins, sondern verändert den Satzbau des Textes, lässt Wörter weg, fügt Wörter ein oder tauscht sie aus. Ist das der Fall, werden exakte Treffer an Ngrammen seltener. Nachvollziehbarerweise sind die längeren Ngramme anfälliger für dieses Problem als kürzere. Wird bei einem Text-Reuse-Fall ein Wort ausgetauscht, dann sorgt dies dafür, dass bis zu fünf der 5-Gramme im wiederverwendeten Text nicht als korrekte Treffer identifiziert werden können. Auf der Unigramebene betrifft dies wiederum nur einen Treffer. Eine Vermutung, warum das Einbeziehen höherwertiger Ngramme in die Ansätze weniger wichtig oder gar schädlich für die Qualität der Ergebnisse ist, wäre, dass die Anzahl der korrekten Treffer auf den höheren Ebenen durch die Textveränderung im Rahmen des Text-Reuse recht gering ist. So kann es eher sein, dass ein Ngram fälschlicherweise gefunden wurde. Damit ließe sich erklären, warum α_4 in Abbildung 5.1 und 5.2 so schnell fällt. Das Risiko, dass gefundene 4-Gramme zu falsch-positiven Klassifizierungen führen, übersteigt den Nutzen, den das Einbeziehen der 4-Gramme bietet.

Abbildung 5.1: Einfluss der Konstanten bei klassischen Ansatz. Gewählt ist die Konfiguration $(2, 0, 0, 16, 250)$, wobei eine der Konstanten variabel ist und entsprechend der X-Achse gewählt wird.

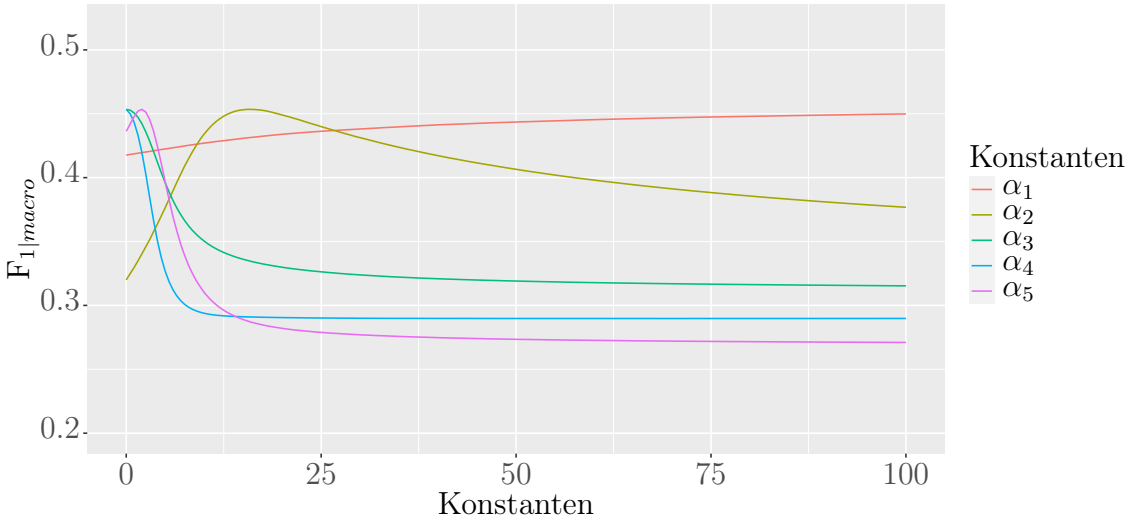
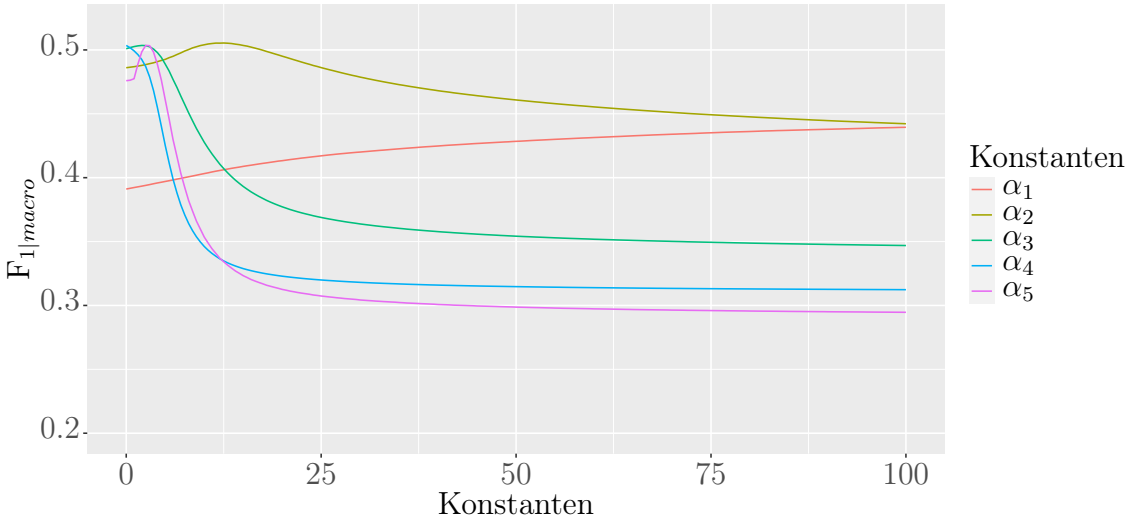


Abbildung 5.2: Einfluss der Konstanten bei erweiterten Ansatz. Gewählt ist die Konfiguration $(2.5, 0, 3, 15, 100000000)$, wobei eine der Konstanten variabel ist und entsprechend der X-Achse gewählt wird.



In Abbildung 5.1 und 5.2 wird die Konstantenkonfiguration entsprechend der Tabelle 5.1 gewählt, wobei eine der Konstanten α_1 bis α_5 variabel ist und von der x-Achse repräsentiert wird. Beispielsweise die lilafarbene Linie in Abbildung 5.1 evaluiert für die Konstantenkonfiguration $t = (x, 0, 0, 16, 250)$ über den klassischen Ansatz, wobei x entsprechend der Werte auf der x-Achse gewählt ist. Da die Konstanten optimiert sind hinsichtlich $F_{1|macro}$, liegen die Hochpunkte der Graphen jeweils auf den Werten, wie sie auch in den Konfigurationen aus Tabelle 5.1 notiert sind. Da die Definitionsbereiche bei 100 enden, lassen sich für α_1 die Hochpunkte nicht auf den Abbildungen feststellen.

Klassischer Ansatz. In Abbildung 5.1 fällt auf, dass sich hohe α_3, α_4 und α_5 stark negativ auswirken auf die Ergebnisqualität. Das Einbeziehen von Trigrammen und 4-Grammen im klassischen Ansatz verringert die Ergebnisqualität generell, wobei der Verlust durch das Einbeziehen von 4-Grammen besonders hoch ist. Im Gegensatz dazu sind Unigramme, Bigramme und Pentagramme schon relevant für den Ansatz. Auch ist das Einbeziehen von Bigrammen am wichtigsten für die Ergebnisqualität im klassischen Ansatz. Werden die Konstanten innerhalb der optimalen Konstantenkonfiguration auf 0 gesetzt, dann führt die bei α_2 zu einem weit größeren Abfall von $F_{1|macro}$ als bei den anderen Konstanten.

Erweiterter Ansatz. Beim Vergleich der Abbildungen 5.1 und 5.2 fällt auf, dass geringe Werte von α_1 sich im erweiterten Ansatz negativer auf die Ergebnisqualität auswirken als im klassischen. $\alpha_2 = 0$ führt beim erweiterten Ansatz im Gegensatz zum klassischen kaum zu Einbußen an Ergebnisqualität. Zwar liegt der beste Wert für α_3 nicht bei 0, jedoch ist der Abfall von $F_{1|macro}$ durch das Setzen von $\alpha_3 = 0$ sehr gering.

5.3 Einfluss eines Schwellwertes

Der Schwellwert beschreibt die minimale Häufigkeit eines Ngrams im Korpus, sodass alle Ngramme, deren Häufigkeiten geringer sind als dieser Schwellwert, nicht im Korpus enthalten sind. So kann die Größe eines Korpus stark verringert werden. Dabei gilt es herauszustellen, dass dies ein anderer Schwellwerte ist, als der der beim Google Books Ngram Korpus angesetzt wurde. Der Schwellwert in Googles Korpus gibt die Mindestanzahl an Dokumente an, die das entsprechende Ngram enthalten müssen, sodass es gezahlt ist. Da der Umpfang von dem zugrundeliegenden Evaluationskorpus in dieser Arbeit wesentlich kleiner ist als der von Google, wird in dieser Arbeit stattdessen die Gesamthäufigkeit des Ngrams betrachtet.

Abbildung 5.3: Einfluss des Schwellwertes beim klassischen Ansatz, Schwellwert bis 10

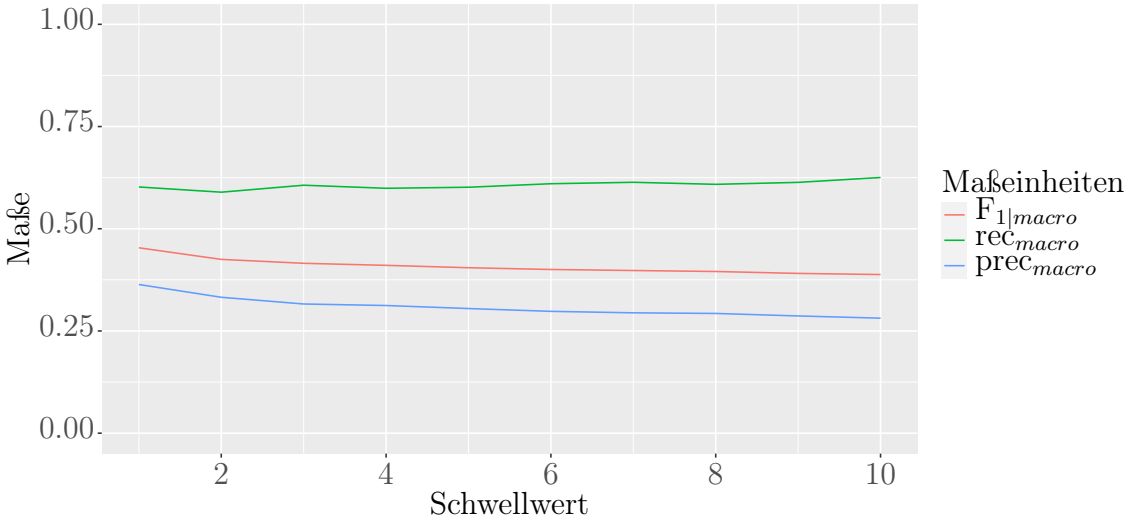
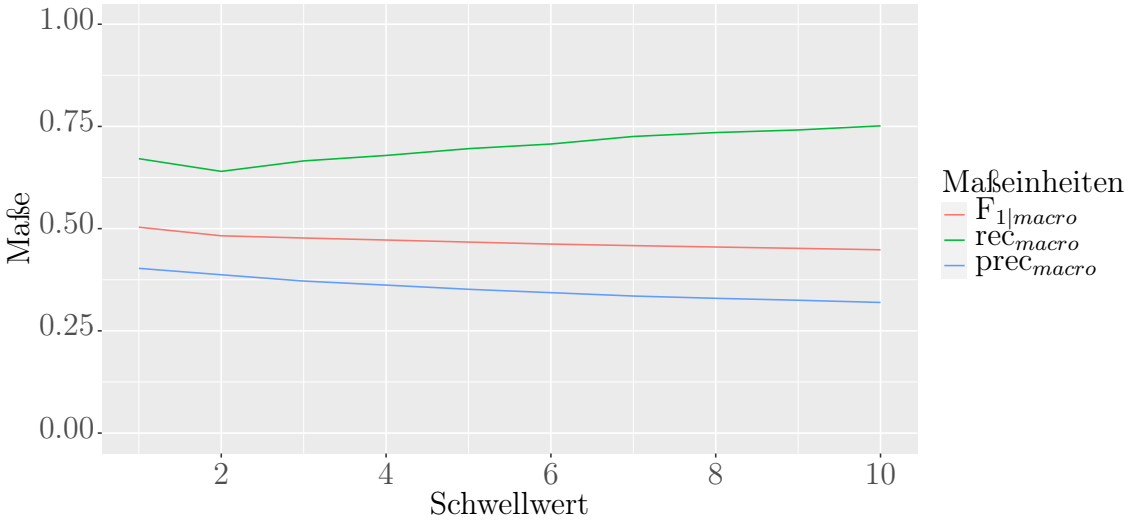


Abbildung 5.4: Einfluss des Schwellwertes beim erweiterten Ansatz, Schwellwert bis 10



Eine Befürchtung war, dass bereits geringe Schwellwerte dazu führen könnten, dass die Ergebnisse der beiden Ansätze stark unter dem Informationsverlust leiden. In Abbildung 5.3 und 5.4 wird der klassische bzw. der erweiterte Ansatz angewendet, wobei ein Schwellwert zugrunde liegt. Für jeden Schwellwert sind die Konstanten angepasst, sodass $F_{1|macro}$ maximiert ist. Die Werte für rec_{macro} und $prec_{macro}$ sind immer mit genau der Konfiguration erzeugt, die auch $F_{1|macro}$ unter dem entsprechenden Schwellwert erzeugt. Bei beiden Ansätzen fällt $F_{1|macro}$ bei steigendem Schwellwert nur gering ab. Auch flacht dieser Verlust an $F_{1|macro}$ ab. Das Steigen von rec_{macro} und das Fallen von $prec_{macro}$, wenn der Schwellwert höher wird, zeigt, dass die Precision weit mehr unter dem Schwellwert leidet als der Recall. Durch den Schwellwert können viele Ngramme, die im Korpus vorkommen, nun nicht gefunden werden. Dies führt aber nur bedingt dazu, dass Fälle, die bei geringeren Schwellwerten korrekt als Text-Reuse klassifiziert werden, nun nicht mehr gefunden werden können. Die fallende Precision deutet darauf hin, dass die Ansätze schneller Sätze fälschlicherweise als Text-Reuse klassifizieren, wodurch die Zahl der falsch-positiven Fälle steigt. Dieses Auseinanderklaffen von rec_{macro} und $prec_{macro}$ ist beim erweiterten Ansatz jedoch etwas stärker ausgeprägt als beim klassischen.

Schwellwert bis 100. Das abflachende Verhalten des Verlustes durch die Erhöhung des Schwellwertes fällt besonders bei der Betrachtung von Abbildung 5.5 und 5.6 auf. In diesen wurden Schwellwerte bis zu 100 angesetzt. Dies ist sehr hoch, vergleicht man dies mit dem Schwellwert von 40 im Google Books Ngram Korpus, besonders da dieses Korpus im Vergleich zum Evaluationskorpus Millionen an Bücher repräsentiert. Wir können davon ausgehen, dass Schwellwerte im zwei- und dreistelligen Bereich zum Ignorieren eines großen Teils der im Korpus enthaltenen Ngramme führt, besonders bei den Kollektionen der höherwertigen Ngramme. Dementsprechend ist es wichtig herauszustellen, dass beide Verfahren trotz Schwellwerten über 50, niedrige aber stabile $F_{1|macro}$-Werte von über 0.3 erreichen. Dies spricht dafür, dass nicht die niedrigen Häufigkeiten die wesentlichen Informationen liefern, um innerhalb der beiden Ansätze korrekt Text-Reuse zu identifizieren, sondern Funde von Ngrammen mit höheren ebenfalls wichtig sind. Die $F_{1|macro}$-Werte flachen schließlich ab beim schrittweisen Erhöhen des Schwellwertes und fallen nicht rapide.

Abbildung 5.5: Einfluss des Schwellwertes beim klassischen Ansatz, Schwellwert bis 100

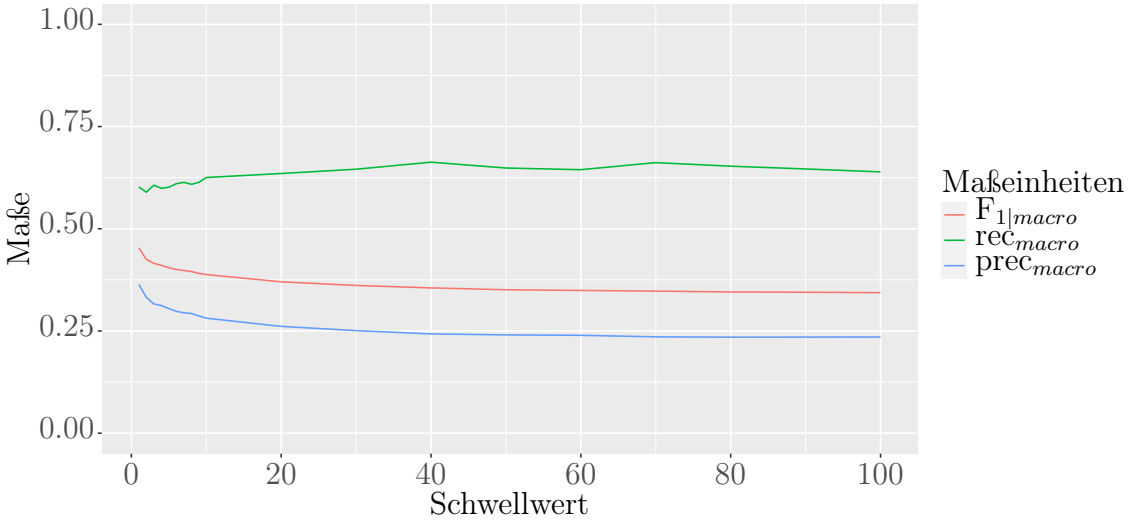
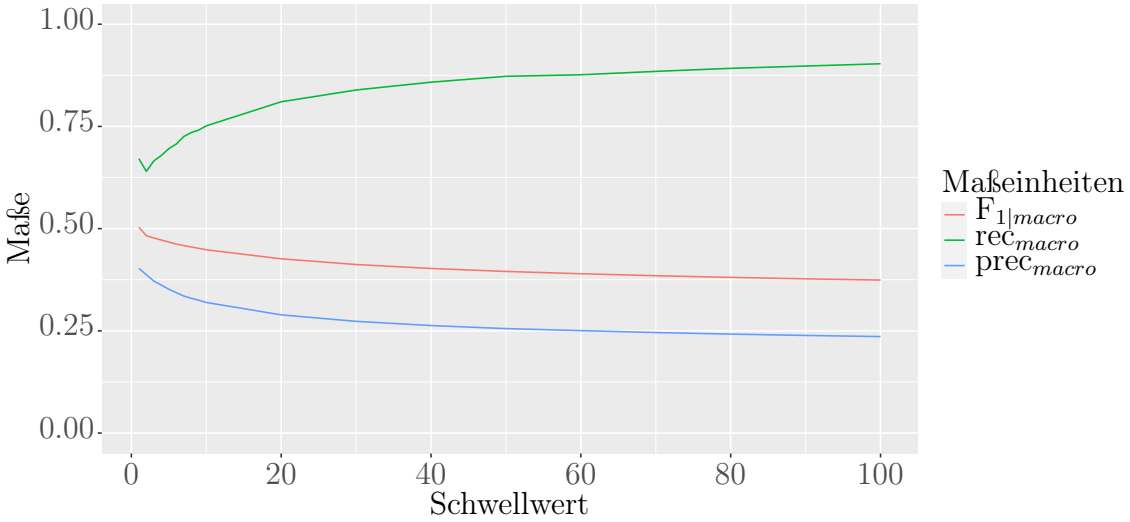


Abbildung 5.6: Einfluss des Schwellwertes beim erweiterten Ansatz, Schwellwert bis 100



Kapitel 6

Diskussion

Die erzeugten Ergebnisse halten wir für annehmbar. Trotz der Aggregation von mehreren Tausend Dokumenten in einer Ngram-Datenbank schafft es der Algorithmus, einen wesentlichen Anteil des Text-Reuses korrekt zu identifizieren. Die $F_{1|macro}$-Werte im mittleren Bereich zeigen dies. Durch diese Evaluationsergebnisse gepaart mit der geringen Komplexität der beiden Ansätze, sind diese Ansätze ein vielversprechender Grundbaustein für die Entwicklung von Methoden und Algorithmen, die einseitig Text-Reuse anhand von einem, wie in dieser Arbeit beschriebenen Ngramkorpus, erkennen sollen. Der erste Abschnitt behandelt mögliche Anpassungen der Ansätze und Anpassungen des Ngramkorpus, die nicht Teil dieser Arbeit sind. Der zweite Abschnitt betrachtet die Ergebnisse der Arbeit im Bezug zum Google Books Ngram Korpus, mit dem Fokus darauf, dass nicht klar ist, wie skalierbar die Ansätze im Bezug zur Korpusgröße sind. Zuletzt wird die Anwendbarkeit von Verfahren zur einseitigen Text-Reuse-Erkennung hinterfragt.

6.1 Mögliche Anpassungen

Die Methoden betrachten nur exakte Treffer auf den rohen Ngrammen. Diese erzeugen annehmbare Ergebnisse. Zu diesem Betrachten von exakten Treffern auf rohen Ngrammen existieren zahlreiche Alternativen sowie Möglichkeiten zur Anpassung. Wie in Kapitel 5 schon angedeutet, kann das Suchen von exakten Treffern, besonders im Bereich von Trigrammen, 4-Grammen und 5-Grammen hinderlich sein, wenn die Text-Reuse-Passagen gegenüber dem Originaltext verändert worden sind. Besonders in dem Kontext kann es sinnvoll sein, sich nicht auf exakte Treffer zu beschränken, sondern auch ähnliche Ngramme als Treffer zuzulassen. Ideen zu solcher Generalisierung wären zum Beispiel: Stemming, das Verwenden von sortierten Ngrammen, das Akzeptieren von Treffern mit einer Jaccard-Ähnlichkeit unter 1 und Stoppwortentfernung

(Wiegmann, 2018).

Stemming. Stemming beschreibt die in der Computerlinguistik verbreitete Methodik, gebeugte sprachliche Wörter auf einen einheitlichen Wortstamm zurückzuführen. Im Kontext dieser Arbeit müsste dies sowohl bei den Ngrammen im Korpus als auch bei den Ngrammen des zu untersuchenden Dokumentes angewendet werden. Dadurch gingen mit den Beugungen Informationen verloren, ob diese jedoch relevant sind, lässt sich bezweifeln. Text-Reuse verwendet schließlich nicht zwingend dieselben Beugungen des Ursprungstextes.

Sortierte Ngramme. Eine alphabetische Sortierung der Wörter innerhalb der Ngramme bewirkt, dass die Reihenfolge, in der diese in den Texten stehen, obsolet wird. Je mehr Wörter ein Ngram erfasst, desto mehr Ngramme können dadurch zusammengefasst werden. Dementsprechend wäre dies eine Maßnahme, um primär auf den Ebenen der Tri-, 4- und 5-Gramme zu generalisieren. Intuitiv wäre in Ergänzung dazu potenziell Stemming sinnvoll.

Jaccard-Ähnlichkeit unter 1. Die Jaccard-Ähnlichkeit beschreibt das Maß an Übereinstimmung zweier Mengen A und B mit $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$. Im Kontext dieser Arbeit werden zwei Ngramme in Bezug auf die darin enthaltenen Wörter verglichen. Die Idee ist, Ngramme als Treffer zu akzeptieren, auch wenn sie teilweise nicht mit dem gesuchten Ngram übereinstimmen, die Jaccard-Ähnlichkeit liegt dann unter 1. Die gesuchten Ngramme müssen nicht identisch sein zu den gefundenen, sondern es ist eine feste Anzahl an Wörtern gewählt, die in beiden Ngrammen unterschiedlich sein darf. Bei Uni- und Bigrammen macht dies offensichtlich keinen Sinn, aber bei Tri-, 4- und 5-Grammen könnte es die Methoden resistenter gegen Veränderungen von Text im Rahmen von Text-Reuse machen. Bei 4- und 5-Grammen wäre es auch möglich, einen Unterschied von zwei Wörtern zuzulassen.

Stoppwortentfernung. Bei der Stoppwortentfernung werden vor der Weiterverarbeitung des Textes eine spezifische Menge an Wörtern unwiderruflich aus dem Text entfernt. Die Annahme ist, dass einige Wörter generell kaum bis gar keine Informationen zum Inhalt des Dokumentes liefern, dabei aber verhältnismäßig häufig sind. Ob das Betrachten dieser im Rahmen der Text-Reuse-Erkennung sinnvoll sind, lässt sich ebenfalls bezweifeln.

POS-Tags. Werden Ngramme betrachtet, die mit POS-Tags annotiert sind, dann ist das im Gegensatz zu den zuvor genannten Anpassungen keine Generalisierung. Vielmehr macht dies die Ngramme sehr viel spezifischer und

Treffer auf diesen unwahrscheinlicher. Ob es sich positiv auf die Qualität der Ergebnisse auswirkt, wenn die in dieser Arbeit beschriebenen Methoden nach Ngrammen, die mit POS-Tags annotiert sind, gesucht werden, ist nicht klar. In anderen Verfahren zur Text-Reuse-Erkennung werden jedoch POS-Tags gewinnbringend verwendet, schreiben Vani and Gupta (2016). Neben Annotationen ist es ebenso möglich POS-Tags als Platzhalter zu verwenden. Dies ähnelt der Idee zum Akzeptieren von Ngrammen mit Jaccard-Ähnlichkeit unter 1 mit der Einschränkung, dass die nicht übereinstimmenden Wörter dennoch den gleichen POS-Tag haben.

Anpassungen des Korpus. Für alle der Anpassungen wäre es sinnvoll die Struktur des Referenzkorpus zu verändern. Fürs Stemming, das Betrachten von sortierten Ngrammen sowie das Akzeptieren von Treffern mit Jaccard-Ähnlichkeit unter 1 sind zwar alle Informationen im genutzten Referenzkorpus enthalten, lassen sich jedoch nur umständlich extrahieren. Bevor beispielsweise gestemmte Ngramme im Korpus gefunden werden können, müssen die Ngramme im Korpus auch gestemmt sein. Theoretisch ist es möglich, ein solches Korpus aus dem vorliegenden Korpus zu erstellen, indem die Ngramme gestemmt und dann gleiche vereinigt werden. Am einfachsten ist es jedoch sicher, die zugrundeliegenden Texte zu stemmen und von da aus das Ngramkorpus zu erstellen. Beim Betrachten sortierter Ngramme sowie Treffern mit Jaccard-Ähnlichkeit unter 1 ist dies ähnlich.
Beim Entfernen von Stoppwörtern sowie bei der Verwendung vom POS-Tags ist dies anders. Die Korpora, die von Nöten wären, um diese veränderten Methoden sinnvoll durchzuführen, müssen von den Volltexten her erstellt werden. Die benötigten Informationen sind nicht in dem Referenzkorpus, das in der Arbeit genutzt wurde, enthalten. Es bietet keine Informationen darüber, welche 5-Gramme aus den Texten extrahiert werden können, wenn zuvor Stoppwörter entfernt wurden. Um POS-Tags setzen zu können, sind ebenso volle Sätze vonnöten. Es werden also andere oder zusätzliche Informationen aus den Texten benötigt. Die ersten drei möglichen Anpassungen beschreiben Möglichkeiten, um anders an die Aufgabe, mit einem solchen Korpus Text-Reuse zu identifizieren, heranzugehen. Dies ist wichtig herauszustellen. Da die Verwendung von Stoppwortentfernung und POS-Tags weitere Informationen benötigt, setzt dies auch eine Anpassung der grundsätzlichen Aufgabe voraus. Betrachtet man das Google Books Ngram Korpus, dann ließen sich Methoden, die POS-Tags verwenden auf dieses Korpus anwenden. Dieses enthält schließlich sowohl mit POS-Tags annotierte Ngramme, als auch Ngramme mit POS-Tags als Platzhalter.

6.2 Skalierbarkeit

Der Google Books Ngram Korpus war eine Grundmotivation, um Methoden zur Text-Reuse-Erkennung auf Korpora dieser Struktur zu entwickeln und zu testen. Auf Googles Korpus erfolgreich Text-Reuse erkennen zu können, ist letzten Endes der Wunsch. Jedoch ist nicht klar, wie weit oder ob die in dieser Arbeit betrachteten Methoden auch auf dem Korpus von Google funktionieren. Dies liegt daran, dass nicht angenommen werden kann, dass die Methoden unabhängig sind gegenüber der Größe des zugrundeliegenden Korpus. Je größer das Korpus ist, desto mehr Dokumente sind in diesem aggregiert. Wir nehmen darum an, dass es dadurch schwieriger wird zu unterscheiden, ob der Treffer eines Ngram im Korpus von Text-Reuse herrührt oder ob er aus einem davon gänzlich unabhängigen Dokument stammt. Der letztere Fall wird mit steigender Anzahl an Dokumenten im Korpus immer wahrscheinlicher. Da das Google Books Ngram Korpus mehr als die 400-fache Menge an Dokumenten repräsentiert gegenüber den Source-Documents des PAN-PC-11, auf dessen Basis die Methoden in der Arbeit evaluiert wurden, ist es nicht möglich einzuschätzen, wie übertragbar die Methoden aus der Arbeit auf Googles Korpus sind.

Ansätze. Es gilt somit, die Skalierbarkeit der Methoden gegenüber der Größe des Referenzkorpus zu untersuchen. Der naheliegende Ansatz wäre, Evaluationskorpora verschiedener Größe aufzustellen und zu untersuchen, wie sich dies auf die Evaluationsmaße auswirkt. Evaluationskorpora, die kleiner sind als die Anzahl der Source-Documents des PAN-PC-11, lassen sich problemlos erstellen, indem nur Teilmengen der Source-Documents als Evaluationskorpus genutzt werden. Signifikant größere Evaluationskorpora müssten jedoch von Grund auf neu erstellt werden. Dabei die Größenordnung von Googles Korpus anzunähern, wirkt jedoch sehr schwer. Einerseits existieren keine frei verfügbaren Korpora, die in der Größenordnung Bücher enthalten, wie Google Books es tut. Man müsse auf andere Medien ausweichen, wie Internetseiten. Dazu existieren schließlich die Clueweb-Korpora als sehr große Sammlungen (The Lemur Project, 2022). Andererseits wird der Rechenaufwand, um den Korpus zu erstellen, immer höher, je größer das Korpus werden soll. Dieser dürfe nicht unterschätzt werden.
Wären die in der Arbeit vorgestellten Methoden bei größeren Korpora gänzlich unbrauchbar, wäre dies fatal. Schließlich verlieren Verfahren zur Text-Reuse-Erkennung viel von ihrer Sinnhaftigkeit, wenn sie nicht auf Korpora angewendet werden können, die einen signifikanten Teil der entsprechenden Dokumente abdecken.

6.3 Einseitige Text-Reuse-Erkennung

Einseitige Text-Reuse-Erkennung hat gegenüber der beidseitigen Text-Reuse-Erkennung bei der praktischen Anwendung ein konkretes Problem. Es kann nicht händisch verifiziert werden, ob eine als Text-Reuse klassifizierte Textstelle korrekt klassifiziert wurde oder fälschlicherweise. Diese falsch-positiven Fälle können bei beidseitigen Verfahren mit einem händischen Vergleich beider Textstellen gefunden werden. Da die nach $F_{1|macro}$ optimierten Werte $prec_{macro}$-Werte von 0.36 bzw. 0.40 sind, sind mehr als die Hälfte der Funde falsch-positiv. Damit ist diese Art der Text-Reuse-Erkennung von der wissenschaftlichen Seite gesehen interessant, jedoch die praktische Anwendbarkeit ist fraglich. Dies gleicht der intrinsischen Text-Reuse-Erkennung, die auch einseitig ist und deren beste Verfahren auch Evaluationsergebnisse im mittleren Bereich liefern (Potthast et al., 2011a).

Kapitel 7

Fazit

Diese Arbeit stellt zwei Ansätze vor, um Text-Reuse anhand eines Ngramkorpus erkennen zu können. Während der klassische Ansatz alle Ngramme, die gefunden werden, miteinbezieht, ignoriert der erweiterte Ansatz alle Ngramme, die bereits in höherwertigen gefundenen Ngrammen enthalten sind. Die fünf Parameter $\alpha_1 ... \alpha_5$ müssen gewählt werden, um festzusetzen, wie viel Einfluss Funde auf den entsprechenden Ngramebenen auf die Klassifizierung von Text-Reuse haben. Wählt man diese, um $F_{1|macro}$ zu optimieren, dann erreicht der klassische Ansatz Evaluationsergebnisse von $F_{1|macro} = 0.45$ und der erweiterte $F_{1|macro} = 0.50$. Eine Annahme ist gewesen, dass die größeren Ngramme wie 4-Gramme oder 5-Gramme wichtiger seien als die Ngramme mit geringeren n, da die größeren spezifischere Informationen liefern.
Bei der Betrachtung des Einflusses der fünf Parameter wird jedoch klar, dass das Einbeziehen von Uni- und Bigrammen relevanter ist, um gute Evaluationsergebnisse zu erreichen, als Trigramme und 4-Gramme. Beim klassischen Ansatz wirkt sich das Einbeziehen dieser Ngramebenen sogar gänzlich negativ auf die Performance aus. Beim erweiterten ist dies beim Einbeziehen von 4-Grammen auch der Fall. Diese Tendenz zu Uni- und Bigrammen kann damit zusammenhängen, dass das Suchen von höherwertigen Ngramen anfälliger ist gegenüber Veränderungen von Wörtern in den wiederverwendeten Textstellen oder der Paraphrasierung dieser Textstellen. Im Google Books Ngram Korpus wurde ein Schwellwert von 40 angesetzt. Das bedeutet, es sind nur die Ngramme erfasst, welche in mehr als 40 Büchern vorkommen (Aiden and Michel, 2011). Eine Annahme war, dass das Ansetzen eines solchen Schwellwertes die Performance der Ansätze schnell stark beeinträchtigt. Dies ist jedoch nicht der Fall. Die Werte der Evaluationsergebnisse flachen zwar ab bei steigenden Schwellwerten, jedoch auf einer stabilen Höhe. Selbst bei Schwellwerten von 50 bis 100 erreichen beide Ansätze $F_{1|macro}$Werte von über 0.3. Des Weiteren fällt auf, dass besonders die Precision unter hohen Schwellwerten leidet und

weniger der Recall.
Wie weit die beiden in der Arbeit vorgestellten Ansätze anwendbar sind auf das Google Books Ngram Korpus, ist nicht klar. Wir können nicht annehmen, dass die Methoden und deren Ergebnisqualität invariant sind gegenüber der Größe des Referenzkorpus. Auf der einen Seite ist Googles Korpus gerade deshalb für die Text-Reuse-Erkennung interessant, weil es diese enorme Größe hat. Auf der anderen Seite kann genau dies der Faktor sein, der problematisch für die Performance von Methoden, die Text-Reuse über solche Korpora erkennen sollen, werden kann. Inwiefern dies der Fall ist, muss noch untersucht werden.

Die zuvor genannten Evaluationsergebnisse der beiden Ansätze sind nur bedingt vergleichbar mit den Ergebnissen der State-Of-The-Art-Verfahren im Text-Alignment. Sie sind kein Durchbruch, dennoch liefern die in der Arbeit vorgestellten Verfahren Ergebnisse vergleichbarer Qualität zu den besten Verfahren zu PANs Shared Task zur intrinsischen Text-Reuse-Erkennung. Die Verfahren ähneln sich nicht im Geringsten. Den intrinsischen Verfahren fehlt jede Art von externer Informationsquelle. Dementsprechend ist diese Aufgabenstellung schwerer als die, der die Arbeit zugrunde liegt. Schließlich nutzen die Ansätze der Arbeit ein Referenzkorpus. Dennoch erkennen sie Text-Reuse einseitig, wie auch die intrinsischen Verfahren. Dementsprechend bietet sich der Vergleich an. Wichtig herauszustellen ist, dass sowohl der klassische als auch der erweiterte Ansatz, deren Aufbau verhältnismäßig einfach ist, und viele mögliche Anpassungen, wie die, die in Kapitel 6.1 vorgestellt sind, noch nicht getestet wurden. Dennoch erreichen sie Evaluationsergebnisse im mittleren Bereich. Dementsprechend könnten die in der Arbeit vorgestellten Ansätze eine sinnvolle Baseline bilden, um weitere Verfahren zu entwickeln und zu testen, welche Text-Reuse anhand ngram-basierter Korpora erkennen können.

Literaturverzeichnis

Erez Lieberman Aiden and Jean-Baptiste Michel. Culturomics: Quantitative analysis of culture using millions of digitized books. In *6th Annual International Conference of the Alliance of Digital Humanities Organizations, DH 2011, Stanford, CA, USA, June 19-22, 2011, Conference Abstracts*, page 8. Stanford University Library, 2011. URL `https://www.science.org/doi/full/10.1126/science.1199644`.

Navot Akiva. Using clustering to identify outlier chunks of text - notebook for PAN at CLEF 2011. In Vivien Petras, Pamela Forner, and Paul D. Clough, editors, *CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands*, volume 1177 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011. URL `http://ceur-ws.org/Vol-1177/CLEF2011wn-PAN-Akiva2011.pdf`.

Demetrios G. Glinos. A hybrid architecture for plagiarism detection. In Linda Cappellato, Nicola Ferro, Martin Halvey, and Wessel Kraaij, editors, *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, volume 1180 of *CEUR Workshop Proceedings*, pages 958–965. CEUR-WS.org, 2014. URL `http://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-Glinos2014.pdf`.

Mike Kestemont, Kim Luyckx, and Walter Daelemans. Intrinsic plagiarism detection using character trigram distance scores - notebook for PAN at CLEF 2011. In Vivien Petras, Pamela Forner, and Paul D. Clough, editors, *CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands*, volume 1177 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011. URL `http://ceur-ws.org/Vol-1177/CLEF2011wn-PAN-KestemontEt2011.pdf`.

Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. Syntactic annotations for the google books ngram corpus. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10,*

2012, Jeju Island, Korea, pages 169–174. The Association for Computer Linguistics, 2012. URL https://aclanthology.org/P12-3029/.

Gabriel Oberreuter and Andreas Eiselt. Submission to the 6th international competition on plagiarism detection. 2014. URL http://www.webis.de/research/events/pan-14.

Gabriel Oberreuter, Gaston L'Huillier, Sebastián A. Ríos, and Juan D. Velásquez. Approaches for intrinsic and external plagiarism detection - notebook for PAN at CLEF 2011. In Vivien Petras, Pamela Forner, and Paul D. Clough, editors, *CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands*, volume 1177 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011. URL http://ceur-ws.org/Vol-1177/CLEF2011wn-PAN-OberreuterEt2011.pdf.

Yurii Palkovskii and Alexei Belov. Developing high-resolution universal multy-type n-gram text similarity detector. In Linda Cappellato, Nicola Ferro, Martin Halvey, and Wessel Kraaij, editors, *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, volume 1180 of *CEUR Workshop Proceedings*, pages 984–989. CEUR-WS.org, 2014. URL http://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-PalkovskiiEt2014.pdf.

Martin Potthast. *Technologies for Reusing Text from the Web*. doctoralthesis, Bauhaus-Universität Weimar, 2012. URL https://e-pub.uni-weimar.de/opus4/frontdoor/index/index/docId/1566.

Martin Potthast, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. An evaluation framework for plagiarism detection. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 997–1005. Chinese Information Processing Society of China, 2010. URL https://aclanthology.org/C10-2115/.

Martin Potthast, Andreas Eiselt, Alberto Barrón-Cedeño, Benno Stein, and Paolo Rosso. Overview of the 3rd international competition on plagiarism detection. In Vivien Petras, Pamela Forner, and Paul D. Clough, editors, *CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands*, volume 1177 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011a. URL http://ceur-ws.org/Vol-1177/CLEF2011wn-PAN-PotthastEt2011a.pdf.

Martin Potthast, Benno Stein, Andreas Eiselt, Alberto Barrón-Cedeño, and Paolo Rosso. Pan plagiarism corpus 2011 (pan-pc-11), June 2011b. URL https://doi.org/10.5281/zenodo.3250095.

Martin Potthast, Tim Gollub, Matthias Hagen, Johannes Kiesel, Maximilian Michel, Arnd Oberländer, Martin Tippmann, Alberto Barrón-Cedeño, Parth Gupta, Paolo Rosso, and Benno Stein. Overview of the 4th international competition on plagiarism detection. In Pamela Forner, Jussi Karlgren, and Christa Womser-Hacker, editors, *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, volume 1178 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012. URL `http://ceur-ws.org/Vol-1178/CLEF2012wn-PAN-PotthastEt2012.pdf`.

Martin Potthast, Matthias Hagen, Anna Beyer, Matthias Busse, Martin Tippmann, Paolo Rosso, and Benno Stein. Overview of the 6th international competition on plagiarism detection. In Linda Cappellato, Nicola Ferro, Martin Halvey, and Wessel Kraaij, editors, *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, volume 1180 of *CEUR Workshop Proceedings*, pages 845–876. CEUR-WS.org, 2014. URL `http://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-PotthastEt2014.pdf`.

Projekt Gutenberg. Projekt Gutenberg zweck and ziel. https://web.archive.org/web/20220303160645/https://www.projekt-gutenberg.org/info/texte/info.html, March 2022a.

Projekt Gutenberg. Welcome to project gutenberg. https://web.archive.org/web/20220304091350/https://www.gutenberg.org/, March 2022b.

Sameer Rao, Parth Gupta, Khushboo Singhal, and Prasenjit Majumder. External & intrinsic plagiarism detection: VSM & discourse markers based approach - notebook for PAN at CLEF 2011. In Vivien Petras, Pamela Forner, and Paul D. Clough, editors, *CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands*, volume 1177 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011. URL `http://ceur-ws.org/Vol-1177/CLEF2011wn-PAN-RaoEt2011.pdf`.

Miguel Sanchez-Perez, Grigori Sidorov, and Alexander Gelbukh. The winning approach to text alignment for text reuse detection at pan 2014: Notebook for pan at clef 2014. *CEUR Workshop Proceedings*, 1180:1004–1011, 01 2014.

The Lemur Project. The clueweb09 dataset. https://web.archive.org/web/20220304091619/https://lemurproject.org/clueweb09.php/, January 2022.

K. Vani and Deepa Gupta. Study on extrinsic text plagiarism detection techniques and tools. *Journal of Engineering Science and Technology Review*, 9: 150–164, 2016.

Matti Wiegmann. Applying the Seed-and-Extend Strategy to Text-Alignment. Master's thesis, Bauhaus-Universität Weimar, Fakultät Medien, Computer Science and Media, June 2018.

Sven Meyer zu Eissen and Benno Stein. Intrinsic plagiarism detection. In Mounia Lalmas, Andy MacFarlane, Stefan M. Rüger, Anastasios Tombros, Theodora Tsikrika, and Alexei Yavlinsky, editors, *Advances in Information Retrieval, 28th European Conference on IR Research, ECIR 2006, London, UK, April 10-12, 2006, Proceedings*, volume 3936 of *Lecture Notes in Computer Science*, pages 565–569. Springer, 2006. doi: 10.1007/11735106_66. URL `https://doi.org/10.1007/11735106_66`.