Exposé zur Studienarbeit:

Erkennung naher Duplikate von Nachrichten-Dokumenten aus dem Web

HUMBOLDT-UNIVERSITÄT ZU BERLIN

Cristian Müller
Betreuer: Prof. Ulf Leser

Institut für Informatik
Humboldt-Universität zu Berlin, Berlin, Germany
(cristian.mueller@informatik.hu-berlin.de)

1 Einleitung

Das Internet ist das größte Lager der Welt für frei verfügbare Texte. Ein großer Vorteil des Internets ist das hohe Maß an Redundanz mit dem Texte hinterlegt sind. Quellen können ausfallen, ohne dass die zur Verfügung gestellte Ressource (Text) nicht mehr verfügbar ist. Diese Redundanz birgt auch Nachteile, wenn nicht mehr erkennbar ist, dass zwei Dokumente Duplikate voneinander sind. Ein solcher Fall tritt z.B. ein, wenn versucht wird, mehrere Quellen von Nachrichten zu integrieren. Diese Studienarbeit verfolgt zwei Ziele: Zum einen wird sie die populärsten Ansätze zur Identifikation naher Duplikate vorstellen. Zum anderen wird sie ein Experiment beinhalten. In diesem Experiment wird einer der Ansätze auf eine Menge von Nachrichten-Dokumenten aus dem Web angewandt. Der Rest dieses Exposés gliedert sich wie folgt: In Abschnitt 2 werden die grundlegenden Ideen hinter den Ansätzen erörtert. Abschnitt 3 stellt das Experiment vor.

2 Grundlagen

Über einen langen Zeitraum haben verschiedene Forschungszweige unterschiedlichste Vorschläge gemacht, um dem Problem der Duplikaterkennung adäquate Lösungen gegenüber zu stellen. Diese Lösungsansätze lassen sich grob in zwei Kategorien einteilen: Fingerprinting-Ansätze und Ähnlichkeitsmaße auf Term Vektoren.

2.1 Fingerprinting

Die Grundidee der Fingerprinting-Ansätze ist es, ein Dokument als Folge von Worten zu betrachten. Diese Folge kann in Teilfolgen zerlegt werden, welche auch als Shingles bezeichnet werden. Zwei Dokumente sind sich ähnlich, wenn sie genügend Teilfolgen gemein haben. Der Vergleich solcher Teilfolgen resultiert in einem Stringvergleich. Stringvergleiche sind teuer in der Berechnung. Daher werden die Teilfolgen mit Hilfe einer Hashfunktion auf Integer abgebildet, da sich diese einfacher vergleichen lassen. Eine oft in diesem Zusammenhang gewählte Hashfunktion ist der SHA1 Algorithmus.

Fingerprinting-Ansätze können sich in vielen Parametern unterscheiden. Einige der Parameter sind z.B. die Länge der Teilfolgen, die Anzahl der Teilfolgen oder die Wahl der Teilfolgen aus der Menge aller Teilfolgen für ein Dokument. Auch die Wahl der Hashfunktion hat Einfluss auf das Ergebnis.

Ein Beispiel für einen Fingerprinting-Ansatz ist der DSC Algorithmus, wie er in [1] vorgestellt wird. Um der Masse an Shingles Herr zu werden, verwendeten die Autoren nur jede 25te Shingle für ihre Vergleiche. Auf diese Art sinkt das Datenvolumen enorm. Da der Auswahl der Shingles kein semantisches Auswahlverfahren zu Grunde liegt, führt dieser Ausschluss auch zu einer willkürlichen Unschärfe. Aufgrund dieser Unschärfe werden relativ verschiedene Dokumente als ähnlich erkannt. Auch mit der Beschränkung auf jedes 25te Shingle ist der DSC Algorithmus teuer in der Berechnung.

Zu dem DSC Algorithmus existiert eine effizientere Alternative, DSC-SS. Bei diesem Ansatz werden die von DSC gewonnenen Shingles eines Dokumentes zu Super Shingles (SS) zusammengefasst. Dies ergibt wenige Super Shingles pro Dokument, im Gegensatz zu den vielen Shingles. Die Ähnlichkeit zweier Dokumente wird durch das Übereinstimmen von Super Shingles bestimmt. Dieser Algorithmus arbeitet nach Angabe der Autoren schlecht auf kurzen Dokumenten, wie sie häufig im Internet zu finden sind.

Die Autoren des I-Match Algorithmus [2] erachten zwei Eigenschaften als besonders wichtig für das Erkennen von Duplikaten in Dokumenten. Zum einen muss die Berechnung der Ähnlichkeit günstig sein. Zum anderen sollte der Algorithmus auch auf kurzen Dokumenten gute Ergebnisse erzielen. Über beide Eigenschaften verfügt der I-Match Algorithmus. Der Algorithmus arbeitet in zwei Schritten: Als Erstes wird aus den Dokumenten ein einziger Fingerprint erzeugt. Als Zweites werden diese Fingerprints verglichen, um Ähnlichkeiten zu bestimmen. Zunächst wird aus einem Dokument eine geordnete Liste der Worte im Dokument erzeugt. Jedes Wort kommt nur ein mal in der Liste vor. Aus dieser Liste werden mit Hilfe der inversen Dokumenthäufigkeit häufig vorkommende und seltene Worte entfernt. Für diese Liste wird daraufhin ein Fingerprint erzeugt, welcher im zweiten Schritt verwendet werden kann.

2.2 Term Vektoren

Die Grundidee der Bestimmung von Ähnlichkeitsmaßen auf der Basis von Term Vektoren entspringt dem Information-Retrieval. Ein Problem des Information-Retrieval ist die Bestimmung relevanter Dokumente zu einer Anfrage aus einer

Sammlung von Dokumenten. Eine populäre Lösung dieses Problems ist es, die Dokumente als Vektoren von Termen zu betrachten. Wobei eine Zeile in einem solchen Vektor der Term-Frequenz eines Terms entspricht. Die Relevanz eines Dokuments wird als Ähnlichkeitsmaß zwischen dem Vektor der Anfrage und dem Vektor des betrachteten Dokuments definiert. Ein beliebtes Ähnlichkeitsmaß im Bereich des Information-Retrieval ist der Cosinusabstand zwischen den beiden Vektoren.

Der Cosinusabstand ist gut geeignet für das Auffinden relevanter Dokumente, da er von den Term-Frequenzen abstrahiert. Für das Auffinden naher Duplikate sind Term-Frequenzen ein wichtiger Indikator. Daher wurde für das Experiment ein Ähnlichkeitsmaß ausgewählt, welches die Ternfrequenzen mit einbezieht.

3 Experiment

Das Experiment wird die Anwendbarkeit von Algorithmen zur Erkennung von nahen Duplikaten auf eine Menge von Nachrichten-Dokumenten aus dem Web untersuchen. Hierzu wird ein Java-Werkzeug implementiert werden, dass Variation 5 des Identitätsmaßes aus [3] umsetzen wird. Den Definitionen aus [3] folgend sind die nachstehenden Symbole definiert als:

N Die Anzahl der Dokumente in der Sammlung D.
n Die Anzahl unterschiedlicher Terme in der Sammlung D.
f_t Die Anzahl der Dokumente, die den Term t enthalten.
$f_{d,t}$ Die Anzahl der Vorkommen von Term t in Dokument d.
f_d Die Anzahl der Terme in Dokument d.
D Die Menge der Dokumentsammlung.
q Das Anfragedokument.
d Ein Dokument aus der Sammlung D.

Die folgende Formel beschreibt die Berechnung von Variation 5 des Identitätsmaßes. $S_{q,d}$ ist ein Maß für die Ähnlichkeit der Dokumente q und d:

$$S_{q,d} = \frac{1}{1 + \log_e\left(1 + |f_d - f_q|\right)} \cdot \sum_{t \in q \cap d} \frac{\left(\frac{N}{f_t}\right)}{1 + |f_{d,t} - f_{q,t}|} \tag{1}$$

Der Term auf der linken Seite des Produktes bestimmt eine Maßzahl der Ähnlichkeit aufgrund der Länge der Dokumente. Dokumente sind sich ähnlicher, wenn sie eine ähnliche Länge haben. Die rechte Seite bestimmt die Summe der Term-Gewichte w_t. Zwei Faktoren beeinflussen w_t: zum einen der Unterschied in den Term-Frequenzen zwischen den beiden Dokumenten q und d, zum anderen die Häufigkeit eines Terms in der Sammlung. Das Term-Gewicht w_t wird dementsprechend größer, wenn die Term-Frequenzen ähnlich sind oder ein Term selten ist.

Als Eingabe für das Java-Werkzeug werden 100000 HTML-Seiten dienen, welche Nachrichten oder Teile von Nachrichten enthalten. Diese HTML-Seiten entspringen unterschiedlichsten Nachrichtenportalen aus dem Web, wie z.B. der

New York Times [4] oder BBC News [5], und werden von Daniel Kummer zur Verfügung gestellt.

Die Seiten enthalten nicht nur den eigentlichen Text, sondern auch Beiwerk, wie Menüs und Werbung. Ein erster Schritt wird daher die Trennung der Nachricht von den unerwünschten Zusätzen sein. Hierzu werde ich einen einfachen regelbasierten Trenner implementieren. Die Aufgabe dieses Trenners ist es, das Dokument in Textmengen zu unterteilen. Der Trenner wird Text solange zusammenfügen bis ein Trennsymbol (HTML-Tag) auftritt. Das Auffinden geeigneter Kandidaten für die Trennsymbole ist ein iterativer und experimenteller Prozess. Nach der Unterteilung wird mindestens eine der Mengen als Nachrichtentext zur weiteren Verarbeitung behalten.

Das gewählte Ähnlichkeitsmaß basiert auf Term Vektoren. Diese Vektoren lassen sich zu einer Matrix vereinen. Ich werde diese Dokumentmatrix als Relation in einem Datenbank-Management-System repräsentieren. Die Relation wird dem Schema `matrix(dokument, term, termfrequenz)` entsprechen. Die zu behandelnde Zahl an Dokumenten ist groß, daher ist eine effiziente Speicherung der Dokumentmatrix sowie eine effiziente Unterstützung der häufigsten Operationen $|f_d - f_q|$, $\left(\frac{N}{f_t}\right)$ und $|f_{d,t} - f_{q,t}|$ mit $t \in q \cap d$ von großer Bedeutung. Die berechneten Ähnlichkeiten $S_{q,d}$ zwischen zwei Dokumenten q und d werden in einer weiteren Relation gespeichert.

Um die Qualität des Ergebnisses zu überprüfen, wird die Ähnlichkeit zweier Dokumente von Hand klassifiziert. Hierzu wird eine Web-Anwendung implementiert werden,welche dem Nutzer jeweils zwei Dokumente präsentiert. Der Nutzer hat daraufhin zu entscheiden, ob sich die Dokumente ähnlich sind. Anhand der Unterschiede zwischen manueller und maschineller Klassifizierung wird die Qualität gemessen.

Literatur

1. Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G.: Syntactic clustering of the web. In: Selected papers from the sixth international conference on World Wide Web, Essex, UK, Elsevier Science Publishers Ltd. (1997) 1157–1166
2. Chowdhury, A., Frieder, O., Grossman, D., McCabe, M.C.: Collection statistics for fast duplicate document detection. ACM Trans. Inf. Syst. **20**(2) (2002) 171–191
3. Hoad, T.C., Zobel, J.: Methods for identifying versioned and plagiarized documents. J. Am. Soc. Inf. Sci. Technol. **54**(3) (2003) 203–215
4. : New york times. `http://www.nytimes.com/`
5. : Bbc news. `http://news.bbc.co.uk/`