INR Teil 1 - Boolesches IR-System

Implementieren Sie ein IR-System bestehend aus Tokenizer, Index (Dictionary, Posting-Listen), Anfragebearbeitung und Rechtschreibkorrektur gemäß dem in der Vorlesung vorgestellten Booleschen Retrieval-Modell. Das System soll die nachfolgenden Anforderungen erfüllen.

Tokenizer

- Der Tokenizer liest Textdokumente vom Filesystem ein und liefert einen Strom von Token. Der Tokenizer kann davon ausgehen, dass Token durch Whitespace, Zeilenumbrüche und Interpunktionszeichen getrennt werden.
- Einfache Apostrophe zwischen Buchstaben werden wie normale Buchstaben behandelt. Anführungszeichen am Anfang oder Ende eines Wortes und andere Sonderzeichen wie %, $, / etc. werden entfernt. Zahlen werden beibehalten.
- Wandeln Sie alle Token in Kleinbuchstaben um.
- Jedes Token, das der Tokenizer ausgibt, wird als Indexterm behandelt. Sie brauchen sich also nicht um die Normalisierung von Token zu kümmern.

Index/Dictionary

- Der Index kann komplett im Hauptspeicher gehalten werden. Sie müssen also nicht die in [INR, Kapitel 4 und 5] beschriebenen Techniken zur Auslagerung des Index auf den Sekundärspeicher, zur verteilten Indexierung sowie zur Index-Komprimierung umsetzen.
- Kapseln Sie die Einträge des Dictionaries und der Posting-Listen in eigenen Klassen, so dass Sie neben dem Term bzw. der DocID später auch noch zusätzliche Informationen hinzufügen können.
- Implementieren Sie die unterschiedlichen Varianten des Merge-Algorithmus für den Vergleich zweier oder mehrerer Posting-Listen bei der Abarbeitung der Booleschen Junktoren. Denken Sie an die möglichen Optimierungen aus der Vorlesung!

Anfragebearbeitung

Ihr System soll Boolesche Junktoren, Phrase-Queries und Proximity-Queries unterstützen. Dabei gelten folgende Anforderungen:

- Gehen Sie davon aus, dass eine Anfrage in konjunktiver Normalform an Ihr System übergeben wird, d.h. als AND-Verknüpfung von OR-Verknüpfungen. Überlegen Sie sich eine möglichst einfach zu parsende Syntax.
- Phrase Queries haben die Form `"term1 term2 term3"`. Sie können sich auf Phrase Queries mit maximal drei Termen beschränken. (Welche Constraints der Form "$Term_j$ muss k Positionen nach $Term_i$ kommen" müssen Sie hier überprüfen?)
- Proximity Queries haben die Form `term1 /k term2` mit der Bedeutung, dass die Terme `term1` und `term2` in einem Dokument innerhalb eines Fensters von höchstens `k` Token auftreten müssen. Mehrfach hintereinander geschaltete Proximity-Operatoren der Form `term1 /k term2 /l term3` brauchen Sie nicht implementieren.
- Phrase und Proximity Queries sollen als Teilglieder eines Booleschen Anfrageausdrucks auftauchen können, also z.B. `("term1 term2" OR term3) AND NOT term4 /3 term5.` Dabei haben die Operatoren `" "` und `/k` Vorrang vor den Booleschen Junktoren.

INR Teil 1 - Boolesches IR-System

Benutzerschnittstelle

- Wie Sie die Benutzerschnittstelle realisieren, bleibt Ihnen überlassen. Eine einfache Möglichkeit besteht darin, das System von der Kommandozeile zu starten und nach dem Aufbau des Index auf die Eingabe der Anfrage durch den Benutzer zu warten.

Rechtschreibkorrektur

Implementieren Sie eine Rechtschreibkorrektur für Anfrageterme wie folgt:

- Ergänzen Sie Ihren Index um einen k-Gram-Index, der alle vorkommenden k-Gramme auf die Indexterme im Dictionary verlinkt. Der Parameter k soll bei der Index-Konstruktion frei wählbar sein.
- Die Rechtschreibkorrektur soll immer dann aktiviert werden, wenn zu einem Indexterm weniger als r Dokumente gefunden werden (r soll ein bei der Indexerstellung frei wählbarer Parameter sein);
- Eingrenzung der möglichen Korrekturterme über k-gramm-Index und Jaccard-Koeffizient J (J frei wählbar zwischen 0 und 1);
- Dann Auswahl der besten Korrekturterme mithilfe der Levenshtein-Distanz;

Testdaten

Zum Testen verwenden Sie den CISI-Corpus (weitere Infos auf kaggle). Der CISI-Korpus besteht aus drei Dateien:

- CISI.ALL enthält 1.460 Dokumente mit einer eindeutigen ID (.I), Titel (.T), Autor(.A), Abstract (.W) und einer Liste von Cross-Referenzen zu anderen Dokumenten (.X). Extrahieren Sie die einzelnen Dokumente mit ihrer ID und dem Abstract und speichern Sie sie als einzelne Textfiles ab.
- CISI.QRY enthält 112 Volltextanfragen. Diese werden wir allerdings erst im zweiten Teil in Zusammenhang mit dem Vektorraummodell und seinen Erweiterungen verwenden.
- CISI.REL enthält die Groundtruth-Daten für die Relevanz der Dokumente aus CISI.ALL und die Anfragen aus CISI.QRY.

Testen Sie Ihr System mit folgenden Anfragen:

- information
- information AND retrieval
- information AND NOT retrieval
- (information OR data) AND analysis
- "information retrieval"
- Information \10 retrieval
- Information \10 retrieval AND "library of congress"

Testen Sie die Rechtschreibkorrektur anhand folgender Begriffe:

- daat
- reserch
- analysi

Werten Sie folgende Laufzeiten aus:

- Aufbau des Index

- Abarbeitung der Anfragen (ohne Rechtschreibkorrektur)
- Ermittlung von Kandidaten bei der Rechtschreibkorrektur

Zeitplan

Insgesamt stehen Ihnen die Wochen 8-10 (24.05. – 07.06.) für die Bearbeitung des ersten Teils zur Verfügung. Wie genau Sie die Arbeiten zeitlich aufteilen, bleibt Ihnen überlassen. Ein Vorschlag wäre:

- W8: 24.05.2023 Aufbau eines Such-Index und Boolesches Retrieval
- W9: 31.05.2023 Erweiterung um Phrase Queries und Proximity Queries
- W10: 07.06.2023 Erweiterung um Rechtschreibkorrektur