Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

Institut für angewandte Informationstechnologie InIT

INRG Praktikum 6: Evaluation

Ziele

1. Sie haben mit dem IR-System Terrier (und dem Front-End IRLab) erfolgreich Dokumente indexiert und können auf diesen suchen.
2. Sie können die gefunden Resultate sinnvoll in Bezug auf Präzision und Ausbeute interpretieren.

Verwendete Software und Datenkollektion

Untenstehend finden Sie eine Liste der zu verwendenden Software und die Datenkollektion.

IRLab

Sie kennen IRLab bereits aus den vorangegangenen Praktika. Es wurde als Front-End für Terrier entwickelt, damit sich Studenten auf das Verständnis der IR Problematik konzentrieren können, anstatt sich mit dem Einrichten eines Systems herumzuschlagen.

Terrier

Terrier ist eine Open-Source Information Retrieval Software, die an der Universität von Glasgow entwickelt wurde. Terrier ist geeignet, um Evaluations-Experimente durchzuführen. Weitere Informationen finden Sie unter http://ir.dcs.gla.ac.uk/terrier.

Cranfield-Kollektion

Kleine Dokumentenkollektion mit 1400 Dokumenten, 225 Anfragen und dazugehörigen Relevanzbewertungen. Sie können diese Kollektion in IRLab als Source auswählen (Dokumentensymbol mit Aufschrift „CF“). Sie finden die Dateien zu Cranfield im Ordner „collections/cranfield“. Im Unterorder “collections/cranfield/docs“ sind die 1400 Dokumente und im Unterordner „collections/cranfield/queries“ die 225 Queries gespeichert.

Vorgehen

- Starten Sie IRLab
- Wählen Sie links oben den Reiter „Lab 6“ aus
- Wählen Sie die Dokumentenkollektion „TREC Cranfield collection, English“ als Source, behalten Sie diese Einstellung die ganze Zeit über bei!

Retrieval in Terrier

Terrier benötigt für sein Batch-Retrieval eine Liste von Query-Dateien, welche zur Abfrage auf den aktuellen Index verwendet werden sollen. Die Liste ist für die Cranfield Kollektion bereits im „collections/cranfield“ Ordner hinterlegt und wird in IRLab durch die Auswahl der Kollektion als Source selektiert.

Eine typische Anfragedatei (einzelne Query) von Cranfield sieht wie folgt aus und muss zwingend die folgende XML-Struktur verwenden:

```
<doc>
<recordId>3</recordId>
<text>
.I 003
.W
what problems of heat conduction in composite slabs have been
solved so
far .
</text>
</doc>
```

Terrier gibt für jede Anfrage gewisse Informationen in der Konsole aus. Nebenbei schreibt Terrier die exakten Retrievalresultate aller Anfragen in ein Resultatfile, das im Unterordner „eval“ abgelegt wird. Das erzeugte Resultfile (.res) dient als Input für die Weiterverarbeitung.

Relevanzbewertungen

Terrier bietet uns anhand der Relevanzbewertung der Cranfield-Testkollektion und der Resultatdatei die Möglichkeit Ausbeute, Präzision und weitere Angaben zu bestimmen. Sie finden die Relevanzbewertung von Cranfield in folgender Datei:

- „cranfield/relevance.txt

Diese Datei ist vierspaltig aufgebaut und enthält folgende Information:

- Spalte 1: Anfragedokument-Identifikator. In unserem Fall <recordId> in der Anfragedatei.
- Spalte 2: keine Bedeutung in diesem Praktikum, standardmässig auf “0“
- Spalte 3: Dokument-Identifikator. Im vorliegenden Fall <recordId> in der Dokumentdatei.
- Spalte 4: binäre Relevanz (0=nicht relevant, 1=relevant)

Eine Beispielszeile “47 0 305 1“ aus der Relevanzbewertung muss wie folgt gelesen werden:
Das Dokument 305 ist für die Anfrage 47 relevant.

Ausgabedatei Terrier

Die Ausgabedatei, die Terrier beim Retrieval ausgibt (im Ordner terrier/var/results), hat eine ähnliche Struktur wie die Datei der Relevanzbeurteilung. Dieses TREC-Format kennen Sie bereits aus dem Praktikum MiniRetrieve.
TREC-Format:

- Spalte 1 : Anfragenummer
- Spalte 2 : Konstante „Q0“
- Spalte 3 : Dokumentennummer
- Spalte 4 : Rangierung
- Spalte 5 : RSV-Value
- Spalte 6 : Gewichtungsschema

Aufgabe 1: Präzision/Ausbeute mit Terrier

Ziel dieser Aufgabe, ist die Effektivität von Terrier zu untersuchen und anhand von **Präzision-Ausbeute** zu analysieren. Verwendet wird BM25, ein Retrieval-Algorithmus, der in Terrier implementiert ist. Sie sollen analysieren, wie Sie schrittweise die Effektivität mit Mitteln, die Sie in der Vorlesung kennen gelernt haben, erhöhen können.
Sie werden vier Konfigurations-Cases analysieren. In untenstehender Tabelle finden Sie die jeweiligen Einstellungen. Werfen Sie kurz einen Blick in eines der Ausgabefiles (Terrier-Resultfiles im Ordner „eval“) und interpretieren Sie dieses.

Features	Case 0	Case 1	Case 2	Case 3
Algorithmus	BM25	BM25	BM25	BM25
Stoppwort-Elimination	**aus**	**ein**	**ein**	**ein**
Stemming (Engl.)	**aus**	**aus**	**ein**	**ein**
(Blind) Relevance Feedback	**aus**	**aus**	**aus**	**ein**

Arbeiten Sie Case für Case durch:

- Stellen Sie die Features entsprechend der Case-Tabelle in IRLab ein
  - Stoppwörter: „English stopword filter“, ein/aus
  - Stemming: „English stemming“, ein/aus
  - Relevance Feedback: „Terrier parameter free query expansion“, ein/aus
- Erstellen Sie den Index neu
- Starten Sie die Evaluation durch Klicken auf den grünen Pfeil nach unten, analog dem Query Teil aus Praktikum 3 und 4.

Für jeden Case erhalten Sie Kennzahlen in Textform im Bereich „Results“ sowie eine Visualisierung in der Form einer Präzision-Ausbeutekurve im Bereich „Visualization“. Die Legende der Kurve beschreibt die Features, welche die jeweiligen Cases charakterisieren. Sie können die Visualisierung mit einem Rechtsklick auf die Darstellungsfläche und der Auswahl von „Clear“ bereinigen.

Analyse

Analysieren Sie nun die Visualisierungen in IRLab. Befassen Sie sich eingehend mit folgenden Fragen:

- Was können Sie aus dieser Grafik herauslesen?

Sie sehen, dass die Verwendung von Stoppwortelimination, Stemming und Expansion die DURCHSCHNITTLICHE Retrievaleffektivität verbessert

- Haben Sie ein solches Resultat erwartet?

Studentenspezifisch. Grundsätzlich wurden diese Hilfsmittel als effektivitätssteigernd in der Vorlesung eingeführt

- Was hat die Verwendung von Stoppwörtern/Stemming/Termexpansion („Blind Relevance Feedback“) prozentmässig gebracht?

Stopwortelimination +4.5%, zzgl. Stemming +12.6%, zzgl. Expansion +23.7%

- Wie wird wohl eine Präzision-Ausbeutekurve einer einzelnen Anfrage im Vergleich zur Präzision-Ausbeutekurve aller 225 Anfragen aussehen?

Die Ergebnisse variieren sehr stark von Anfrage zu Anfrage. Siehe auch Aufgabe 2.

Aufgabe 2: Analyse einzelner Anfragen

In Aufgabe 2 geht es darum, einzelne Anfragen und deren Resultat genauer unter die Lupe zu nehmen. Sie analysieren dabei die Resultate immer aus zwei verschiedenen Perspektiven. Dazu haben wir als Beispielbenutzer einerseits die präzisionsorientierte Anna und andererseits den ausbeuteorientierten Bob.

- **Anna**: Annas Informationsbedürfnis ist sehr präzisionsorientiert. Das heisst, Sie möchte innerhalb der ersten zehn gefunden Dokumente so viel relevante Treffer als möglich. Annas Informationsbedürfnis entspricht dabei dem heute gängigen Google-Benutzer, der sich lediglich die erste Trefferseite (erste 10 Hits) anschaut. Anna betrachtet deshalb nur die ersten zehn Dokumente (Präzision bei 10 gefundenen Dokumenten = P@10).
- **Bob**: Bobs Informationsbedürfnis ist im Gegensatz zu Annas ausbeuteorientiert. Das heisst, er möchte möglichst alle relevanten Dokumente zu seiner Anfrage finden. Sein Informationsbedürfnis entspricht z.B. dem eines Patentanwaltes, der alle relevanten Dokumente finden möchte. Er ist dafür bereit, die ersten 100 Dokumente anzuschauen (Präzision bei 100 gefunden Dokumenten = P@100). (Beachten Sie: Bob kann die eigentliche Ausbeute nicht kennen, denn was man nicht findet, nimmt man nicht wahr).

Vorgehen

Das Vorgehen für die Erstellung der Präzision-Ausbeutediagramme erfolgt ähnlich wie in der ersten Aufgabe. In dieser Aufgabe beschränken Sie sich auf den Vergleich von Case 0 und Case 3.

- Wiederholen Sie kurz die Auswertung der Cases 0 und 3
- Kopieren Sie die Kennzahlen aus dem „Results“-Bereich für beide Cases in einen Texteditor. Sinnvollerweise speichern Sie diese Daten ab.

Queries, die zu untersuchen sind

Versetzen Sie sich nun in die Lage der beiden fiktiven Benutzer (Anna und Bob) und beurteilen Sie, was sich zwischen Case 0 und Case 3 verändert hat. Folgende drei Hauptfragen sollen jeweils für den Unterschied zwischen Case 0 und Case 3 beantwortet werden:

- Hat sich die Effektivität gemäss Standardmass geändert (MAP)?
- Wird Anna das Resultat subjektiv als besser wahrnehmen (P@10)?
- Wird Bob das Resultat subjektiv als besser wahrnehmen (P@100)?

Die jeweiligen Werte für die Präzision bei 10 Dokumenten und die Präzision bei 100 Dokumenten können Sie den zwischengespeicherten Resultaten entnehmen. Sie können dazu die Anzahl der gefundenen relevanten Dokumente auf einfache Art und Weise bestimmen, indem Sie den jeweiligen Präzisionswert mit der Präzisionsstelle multiplizieren. Beispiel: P100 = 0.03 bedeutet, dass drei relevante Dokumente (0.03*100) innerhalb der ersten 100 Dokumente gefunden wurden.
Untersuchen Sie nun nach dem obigen Muster und Vorgehen die folgenden Anfragen in der Tabelle genauer. Tragen Sie ihre Erkenntnisse in die Tabelle ein.

Das Ausfüllen der Datentabelle sei den einzelnen Studierenden überlassen. Siehe unten für Kommentare zu den einzelnen Anfragen.

Query	Case	MAP	ret	Anna Case 0 vs. Case 3	Bob Case 0 vs. Case 3
Q37	0				
	3				
Q55	0				
	3				
Q72	0				
	3				
Q118	0				
	3				
Q145	0				
	3				
Q167	0				
	3				
Q205	0				
	3				
Q209	0				
	3				

ret: Anzahl gefundener und relevanter Dokumente innerhalb der ersten tausend Dokumente pro Query. Zu finden ist diese Angabe in den Resultaten unter „relevant retrieved“.

Q37: Case 3 bringt gegenüber Case 0 eine Verschlechterung sowohl für Anna wie auch Bob. Zwar wird ein neuntes relevantes Dokument gefunden, aber offenbar mit einem zu tiefen Ranglistenplatz, und um den Preis eines besseren Rankings
Q55: Case 3 bringt eine klare Steigerung gegenüber Case 0, sowohl für Anna, wie auch für Bob
Q72: Sowohl Anna wie Bob profitieren von Case 3, Bob besonders stark (Ausbeute von 2 relevanten auf 10 relevante Dokumente erhöht)
Q118: Ein interessanter Fall. Die Mean Average Precision ist für Case 3 verglichen mit Case 0 um fast 70% gefallen. Trotzdem bleibt das subjektive Empfinden von Anna und Bob gleich – MAP ist zwar unser wichtigstes "Einzahlmass" für die Bewertung der Retrievaleffektivität, hat hier aber keinen direkten Zusammenhang mit der Wahrnehmung unserer Beispieluser.
Q145: Der MAP fällt für Case 3, und auch die Zufriedenheit von Anna wird wahrscheinlich leiden. Anders der Fall für Bob: er findet ein relevantes Dokument mehr!

Q167: Der MAP hat sich deutlich verbessert für Case 3. Für Anna und Bob hat dies jedoch keinen Einfluss – beide bleiben gleich (un-)zufrieden.
Q205: Für diese Anfrage bringt Case 3 eine gewaltige Verbesserung: eine Steigerung des MAP um 2800%! Case 3 bringt das "perfekte" Resultat: die einzigen beiden relevanten Dokumente sind auf den Rängen 1 und 2. Für Anna ist die Verbesserung dramatisch; für Bob etwas weniger spektakulär – aber auch er findet nun das zweite relevante Dokument.
Q209: Hier ist die Wahrnehmung von Anna und Bob sehr unterschiedlich: für Anna ist Case 3 eine klare Verschlechterung, während Bob von den zusätzlichen Massnahmen in Case 3 profitiert.

Abschlussfragen

- Warum denken Sie, wurden diese Anfragen exemplarisch gewählt?
  Die Beispiele illustrieren, dass die subjektive Wahrnehmung nicht notwendigerweise mit den gemessenen Werten übereinstimmen muss. Dies ist besonders dann der Fall, wenn auf einzelne Werte (z.B. Mean Average Precision) zurückgegriffen wird.
- Spiegelt die subjektive Wahrnehmung immer die objektive Messung?
  Nein. Siehe oben und siehe die Kommentare zu den Beispielen.
- Besteht zwischen den Ansprüchen von Anna und Bob ein Widerspruch?
  Es besteht insofern ein Widerspruch, dass Präzision und Ausbeute im Allgemeinen nicht unabhängig voneinander verbessert werden können. Entsprechende Massnahmen können immer auch negative Auswirkungen nach sich ziehen. In unserem Fall schneidet Case 3 DURCHSCHNITTLICH in vielen Bereichen (MAP, Präzision @ 10, Präzision @ 100) besser als Case 0 ab. D.h., Präzision und Ausbeute wurden beide im Durchschnitt verbessert. Für einzelne Anfragen muss differenziert werden (siehe oben). Weitere Massnahmen (aggressiveres Stemming, n-Gram Suche, etc.) würden den Konflikt zwischen Präzision und Ausbeute wahrscheinlich noch verschärfen.
- Wie reagieren Sie als Systemverantwortlicher auf diese Problematik?
  Das Bewusstsein für die Problematik ist wichtig. Entsprechend der konkreten Anforderungen der Benutzer muss eine präzisions- oder ausbeuteorientierte Suche umgesetzt werden. Die Robustheit der Lösung (Minimierung der Ausreisser) ist in der Praxis oft zentral.
  Die IR-Lösung muss immer fallspezifisch konfiguriert/implementiert werden.