

## IR-Grundlagen - Worksheet 4

Dieses Arbeitsblatt behandelt Language Models, im speziellen n-Gramme und smoothing. Ziel des Arbeitsblattes soll sein für Dokumente und Queries Language-models erstellen zu können, anhand der errechneten Wahrscheinlichkeiten eine ranked-Retrieval Liste zurückgeben zu können und Smoothing zu integrieren.

Grundsätzliche Struktur:

Ausgeführt wird die `MainController` Klasse. Hier werden die Dokumente in die `Collection` `ArrayList` eingelesen. Anschließend wird ein Objekt der `Index`-Klasse erzeugt. Dann wird eine Query vom User eingelesen und im Index gesucht. Zuletzt werden die jeweiligen Ergebnisse ausgegeben.

1. Zuerst erstelle wir ein einfaches Unigramm-modell. Dabei soll für jedes Dokument ein Language-Model erstellt werden. Dazu müssen folgende Funktionen implementiert werden. Erstellen sie für die Bearbeitung dieser Aufgaben wenn nötig weitere Funktionen und Klassen.

- a. Für jedes Dokument muss eine eigenes Language-Modell erstellt werden. Implementieren sie den Konstruktor der Klasse `Index` und legen Sie dort für jedes Dokument der `Collection` (gespeichert im `collection` Objekt - analog zu den letzten Arbeitsblättern) ein Objekt der `Unigramm` Klasse an. Sie können alle `UniGramm`-Modelle in der `ArrayList` `unigrams` speichern.



- b. Der Konstruktor der `Unigramm` Klasse ruft die Methode

```
float calculateProbabilities(Document doc)
```

auf, in der für alle Wörter in diesem Dokument eine Wahrscheinlichkeit errechnet wird. Die Wahrscheinlichkeiten können in der `HashMap` `probabilities` gespeichert werden. Die Formel hierfür lautet:

$$P(wort_{doc}) = \frac{count(wort)}{length(doc)}$$

Implementieren sie diese Methode.

2. Nachdem alle Schritte zur Indexierung implementiert sind, können wir uns jetzt mit der Suche beschäftigen.

- a. Die `Index` Klasse enthält die Methode

```
ArrayList<Integer> searchUnigramm(String query, int k)
```

Diese Methode soll für eine gegebene Query für jedes Language-Modell der Collection die Wahrscheinlichkeit berechnen, diese Query von dem jeweiligen Language-Modell erzeugt wurde (Denken sie daran dass in diesem Fall das Ranking nach  $P(d|q)$  und  $P(q|d)$  als gleichwertig behandelt werden können).



- b. Wenn ein Queryterm nicht im Dokument vorkommt ist der gesamte Score gleich 0. Das versuchen wir durch Smoothing auszugleichen. Wir verwenden hierfür wie in den Vorlesungsfolien ein Language-Modell für die gesamte Collection. Die Formel lautet:

$$P(wort_{col}) = \frac{count(wort)}{length(col)}$$

Diese Wahrscheinlichkeit wird folgendermaßen verwendet um die Ergebnisse zu smoothen:

$$P(wort | doc) = \lambda P(wort_i) + (1 - \lambda) P(wort_{col})$$

Benutzen sie diese Formel (vorgeschlagener Wert  $\lambda = 0.5$ ) um Smoothing in ihr System einzubauen.

---