

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.

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 unigramms` 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.

- c. 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).

- d. Welche Probleme kann ein Uni-Gramm-Modell haben?

2. 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 model 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.
