# Aufgabenblatt 11 –

Einführung in die Computerlinguistik (WS19/20) Abgabe bis –

---

**Aufgabe 1)** Clustering-Evaluation **Punkte: x**

Gegeben sei die folgende Menge an Elementen:

$$M = \{a, b, c, d, e, f, g, h, i, j, k, l\}$$

Zudem ist ein Gold-Clustering $G$ vorgegeben:

$$G = \{a, b, c, d\}, \{e, f, g, h\}, \{i, j, k, l\}$$

Die Elemente dieser Menge werden von einem System $S$ wie folgt geclustert:

$$S = \{a, b, c, e\}, \{d, f, g, h, i, j, k, l\}$$

Berechnen Sie für das System $S$

1. Rand Index
2. Purity
3. Normalized Mutual Information (NMI)

in Bezug auf Gold-Cluster $G$.

**Aufgabe 2)** IR-Evaluation (aus einer alten Klausur) **Punkte:** –

Ein Korpus enthält eine Menge von Dokumenten, von denen 8 Dokumente für eine gegebene Query relevant sind.

Ein Information Retrieval System gibt die folgenden 20 Dokumente zurück, wobei das Ranking von links nach rechts angeben ist (d.h. ganz links steht das Dokument, das das System als das relevanteste erachtet). Die 5er-Gruppen haben keine semantische Bedeutung, sondern sollen nur die Lesbarkeit verbessern.

```
R R N N N    N N N R N    R N N N R    N N N N R
```

$R$ bedeutet, dass das gefundene Dokument wirklich relevant ist, $N$ bedeutet nicht-relevant.

1. Berechnen Sie precision und recall des Systems.
2. Berechnen Sie die Average Precision des Systems für diese Query.
3. Warum wird in IR meist Average Precision oder Precision at K verwendet, anstatt Precision und Recall zu verwenden? (1-Satz-Antwort)
4. Warum ist für Websuche Precision at K leichter zu verwenden als (Mean) Average Precision? (1-Satz-Antwort)

**Aufgabe 3)** Term Frequency - Inverse Document Frequency **Punkte:** —

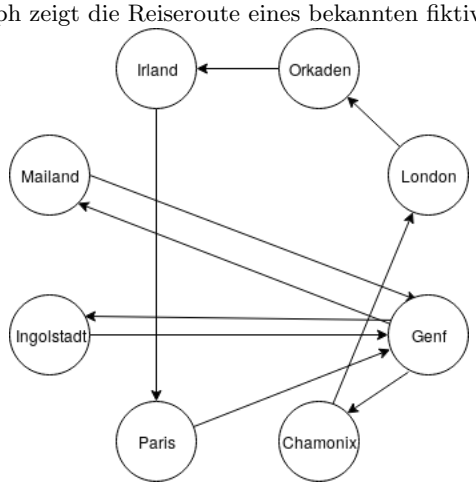
Gegeben sind 9 Zeilen aus verschiedenen Gedichten von Emily Dickinson:

$d_1$ life is but life, and death but death !
$d_2$ you left me, sweet, two legacies,
$d_3$ sweet litigants for life.
$d_4$ the quiet nonchalance of death
$d_5$ my business, just a life i left,
$d_6$ sweet debt of life, each night to owe,
$d_7$ at least to pray is left, is left.
$d_8$ a death blow is a life blow to some
$d_9$ so quiet, oh, how quiet!

1. Erstellen Sie eine Term-Dokument-Matrix $M$ (mit Term Frequency als Werte) für die 5 Types *life*, *sweet*, *death*, *quiet* und *left* für alle Dokumente.
2. Errechnen Sie die Inverse Document Frequency für die selben 5 Types und verwandeln Sie dann $M$ in eine TF-IDF-Matrix.
3. Wie würden Sie die Ähnlichkeit zwischen zwei Dokumenten berechnen?
4. Wie würden Sie die Ähnlichkeit zwischen der query *death, life* und den einzelnen Dokumenten berechnen?

**Aufgabe 4)** Page Rank Iterativ **Punkte:** –

Der folgende Graph zeigt die Reiseroute eines bekannten fiktiven Charakters:




Stellen Sie den Graphen durch eine Adjazenzmatrix dar.

Verwandeln Sie diese in eine stochastische Transition Matrix, wie im PageRank-Algorithmus beschrieben (mit Sprungwahrscheinlichkeit $\alpha = 0.1$).

Benutzen Sie dann die iterative Berechnung für den steady-state vector, um die Approximation des PageRank-Vektors zu berechnen, bis sich dieser stabilisiert hat. Programmieren Sie diesen Algorithmus, um Zeit zu sparen! Initialisieren Sie den Vektor $v$ mit einem stochastischen Vektor (was für die Konvergenz notwendig ist). Ein *stochastischer Vektor* besitzt ausschließlich nicht-negative Werte, die sich auf 1 aufaddieren, wie (0.5, 0, 0, 0.5, 0) oder (1,0,0).

**Aufgabe 5)** PageRank: Lösung durch Gleichungssystem **Punkte: -**

Anstatt iterativ kann der PageRank auch manchmal einfach als lineares Gleichungssystem berechnet werden. Dazu wird der Haupteigenvektor der Matrix berechnet, der durch die Matrix wieder auf sich selbst transformiert wird, denn der steady state vector hat die Eigenschaft $v * T = v$. Stellen Sie für folgende Matrix das Gleichungssystem auf und berechnen Sie den Eigenvektor. Berücksichtigen Sie bei der Berechnung, dass $v$ auch ein stochastischer Vektor ist.

$$\begin{pmatrix} 0.33 & 0.33 & 0.33 \\ 0.475 & 0.05 & 0.475 \\ 0.05 & 0.9 & 0.05 \end{pmatrix}$$

**Aufgabe 6)** PageRank Advanced **Punkte:** –

**a)** Eigenschaften der Iteration (Advanced) *Punkte:* –

Zeigen Sie, dass solange man bei PageRank mit einem stochastischen Vektor beginnt, man bei jeder Iteration wieder einen stochastischen Vektor erhält, sprich dass das Produkt eines stochastischen Vektors mit einer stochastischen Matrix wieder ein stochastischer Vektor ist.

**b)** Mindestgröße PageRank (Advanced) *Punkte:*

Überlegen Sie sich, dass der page rank jeder Seite mindestens $\alpha/N$ ist (wobei $\alpha$ die "Sprungwahrscheinlichkeit" ist), d.h. der steady state vector hat nur Einträge von mindestens $\alpha/N$.

Was bedeutet das für den steady state Vektor und damit den Page Rank, wenn $\alpha$ nah bei 1 liegt (also man meistens springt)?

Ich empfehle auch die Seite `http://www.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture3/lecture3.html` als alternative PageRankerklärung. Vorsicht: Auf dieser Seite sind die Reihen und Spalten vertauscht (im Gegensatz zur Vorlesung). Dies geht natürlich genauso. Man muss dann aber den Vektor rechts, nicht links ran multiplizieren bei der Iteration.