Prof. Dr. Philipp Schaer, Timo Breuer, Björn Engelmann
Institut für Informationswissenschaft

I Beispielrechnung Retrieval mit dem Vektorraummodell

Stellen Sie sich vor, Sie haben einen Dokumentenkorpus, der wie folgt aussieht:

Dokument 1: „new york times“

Dokument 2: „new york post“

Dokument 3: „los angeles times“

Die Suchanfrage q an das Retrievalsystem lautet „new new times“.

- Erstellen Sie die Term-Dokument-Matrix mit einer **tf-idf-Gewichtung** von **$(1 + \log_{10}(tf_{d,f}))$ * $\log_{10}(N/df_t)$**.
- Berechnen Sie den **Score** nach für jedes Dokument in Abhängigkeit von der Suchanfrage q nach dem Vektorraummodell bzw. der Kosinus-Ähnlichkeit.
- In welcher Reihenfolge werden die Dokumente in der Ergebnisliste angezeigt?

Zunächst berechnen wir die idf-Werte. Hierbei gehen wir davon aus, dass es N=3 Dokumente gibt:

angeles	$\log_{10}(3/1) = 0{,}477$
los	$\log_{10}(3/1) = 0{,}477$
new	$\log_{10}(3/2) = 0{,}176$
post	$\log_{10}(3/1) = 0{,}477$
times	$\log_{10}(3/2) = 0{,}176$
york	$\log_{10}(3/2) = 0{,}176$

Als nächstes ermitteln wir die reinen tf-Werte und tragen diese in eine Tabelle ein:

	angeles	**los**	**new**	**post**	**times**	**york**
d1	0	0	1	0	1	1
d2	0	0	1	1	0	1
d3	1	1	0	0	1	0

Danach werden die Werte in der Tabelle entsprechend der angegebenen tf-idf-Formel ausgefüllt. Ein Beispiel für den Eintrag zu angeles in d3: (1 + log10(1)) * log10(3/1) = (1 + 0) * 0,477 = 0,477

Da in unserem Beispiel, die reinen tf-Werte nur die Werte 0 oder 1 annehmen, ist die Berechnung sehr einfach und übersichtlich. Log10(0) ist nicht definiert, daher belassen wir alles mit einem tf-Wert von 0 einfach auf diesem Wert und rechnen da gar nicht erst weiter.

	angeles	**los**	**new**	**post**	**times**	**york**
d1	0	0	0,176	0	0,176	0,176
d2	0	0	0,176	0,477	0	0,176
d3	0,477	0,477	0	0	0,176	0

Anschließend berechnen wir den tf-idf-Wert der Anfrage, wobei Sie bedenken müssen, dass der tf-Wert für „new“ bei 2 liegt!

	angeles	**los**	**new**	**post**	**times**	**york**
q	0	0	0,229	0	0,176	0

Wir berechnen die Länge der Dokumentvektoren und des Anfragevektors:

Länge von d1 $= \sqrt{0{,}176^2 + 0{,}176^2 + 0{,}176^2} = 0{,}305$

Länge von d2 $= \sqrt{0{,}176^2 + 0{,}477^2 + 0{,}176^2} = 0{,}538$

Länge von d3 $= \sqrt{0{,}477^2 + 0{,}477^2 + 0{,}176^2} = 0{,}697$

Länge von q $= \sqrt{0{,}229^2 + 0{,}176^2}\ 0 = 0{,}289$

Wir normalisieren die tf-idf-Werte mit den ermittelten Längen, indem wir durch die Länge teilen:

	angeles	**los**	**new**	**post**	**times**	**york**
d1	0	0	0,577	0	0,577	0,577
d2	0	0	0,327	0,887	0	0,327
d3	0,684	0,684	0	0	0,253	0
q	0	0	0,792	0	0,609	0

Nun können wir die Ähnlichkeit zwischen jedem Dokument und q ausrechnen:

cos(d1,q) = (0*0 + 0*0 + 0,577*0,792 + 0*0 + 0,577*0,609 + 0,577*0) = 0,808

cos(d2,q) = (0*0 + 0*0 + 0,327*0,792 + 0,887*0 + 0*0,609 + 0,327*0) = 0,259

cos(d3,q) = (0,684*0 + 0,684*0 + 0*0,792 + 0*0 + 0,253*0,609 + 0*0) = 0,154

Die finale Reihenfolge des Rankings lautet also: d1, d2, d3