Universität Potsdam
Institut für Informatik
Lehrstuhl Maschinelles Lernen

Information Retrieval, Vektorraummodell

Tobias Scheffer
Thomas Vanck

Information Retrieval

- Konstruktion von Systemen, die die Informationsbedürfnisse der Benutzer befriedigen.
- Repräsentation, Speicherung, Zugriff auf Dokumente.

Schlüsselwort-Modell

- Dokument repräsentiert durch Schlüsselwörter.
- Schlüsselwörter unterschiedlich speziell und relevant, unterschiedliche Gewichte.
- $K = \{k_1, \ldots, k_t\}$ sind Index-Terme.
- Dokument $d_J = (w_{1,J}, \ldots, w_{t,J})$
- $w_{i,J}=0$, wenn k_i nicht in d_J vorkommt, $w_{i,J}=g_i(d_J)$ sonst.

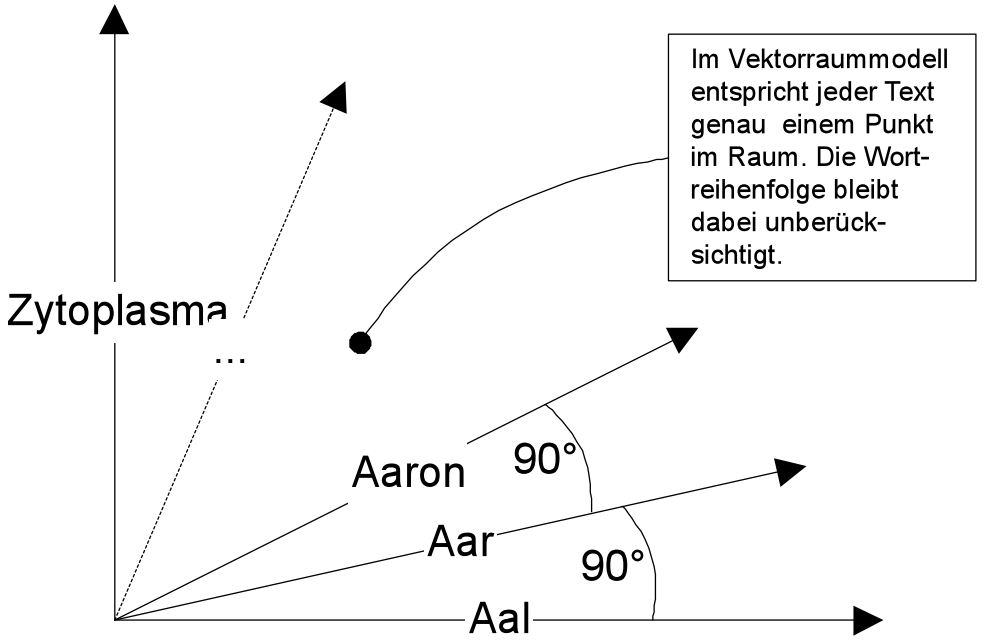
Boolesches Modell

- Dokumente: beschrieben durch Vorkommen von Schlüsselwörtern.
- Suchanfrage sind Boolesche Ausdrücke.
- Ergebnisse der Suchanfrage werden durch Mengenoperationen bestimmt.
- Binäre Entscheidungen, kein Ranking der Ergebnisse.
- Dokument $d_J = (w_{1,J}, \ldots, w_{t,J})$, für Schlüsselwörter.
- $w_{i,J}=0$, wenn k_i nicht in d_J vorkommt, $w_{i,J}=1$ sonst.

Vektorraummodell

- Bag-of-Words:
  - Nur die Menge der Wörter wird berücksichtigt.
  - Keine Berücksichtigung der Wortreihenfolge.
- Vektorraummodell:
  - Jeder Text = Punkt in hochdimensionalem Raum.
  - Raum hat eine Dimension für jedes Wort der Sprache.
- Variante: nur Wortstämme berücksichtigen, „Stop-Wörter“ entfernen.

Vektorraum-Modell



- Text wird repräsentiert durch Punkt im hochdimensionalen Raum,
- Wortreihenfolge bleibt unberücksichtigt,
- Wortstammbildung, „inverse document frequency“.

Vektorraum-Modell: TFIDF-Repräsentation

- Termfrequenz eines Wortes in einem Text =
  # Vorkommen des Wortes im Text.
- Problem: Einige Wörter sind weniger relevant (und, oder, nicht, wenn, …)
- Lösung: Inverse Dokumentenfrequenz

$$IDF(wort_i) = \log \frac{\#\text{Dokumente}}{\#\text{Dokumente , in denen Wort}_i \text{ vorkommt}}$$

Vektorraum-Modell

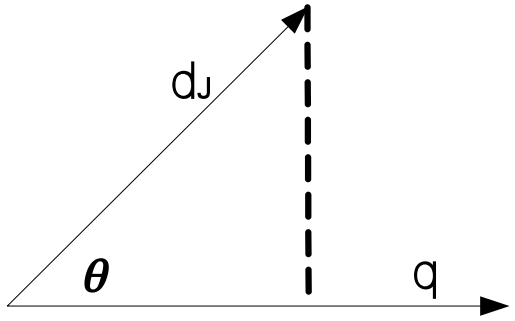
- Problem: Lange Texte haben lange Vektoren, führt zu Verzerrungen beim Ähnlichkeitsmaß.
- Lösung: Normieren

- TF-IDF ($Wort_i$) = norm(TF($Wort_i$) * IDF($Wort_i$)).
- Repräsentation eines Textes:

$$TFIDF(Text) = \frac{1}{norm}\begin{pmatrix} TF(Wort_1)\cdot IDF(Wort_1) \\ \vdots \\ TF(Wort_n)\cdot IDF(Wort_n) \end{pmatrix}$$

Vektorraum-Modell

- Ähnlichkeit zwischen Text d_J und und Anfrage q: Cosinus des Winkels zwischen den Vektoren.



- Ähnlichkeit: $sim(d_j, q) = \cos(\theta) = \frac{d_j \cdot q}{|d_j| \cdot |q|}$
- Zwischen 0 und 1.

Probabilistisches Modell

- Binary independence retrieval (BIR) model.
- Dokument $d_J = (w_{1,J}, \ldots, w_{t,J})$, für Schlüsselwörter.
- $w_{i,J}=0$, wenn k_i nicht in d_J vorkommt, $w_{i,J}=1$ sonst.
- R ist die Menge der relevanten Dokumente.
- Gesucht: Schätzer für $P(R \mid d_J)$: P(Dokument ist relevant).
- Ähnlichkeit: Odds-Ratio

$$sim(d_j, q) = \frac{P(R \mid d_j)}{P(\overline{R} \mid d_j)}$$

Probabilistisches Modell

- Bayes‘ Regel: $sim(d_j, q) = \frac{P(R \mid d_j)}{P(\bar{R} \mid d_j)}$

$$= \frac{P(d_j \mid R)P(R)}{P(d_j \mid \bar{R})P(\bar{R})}$$

- P(R) ist konstant (für alle Dokumente gleich)

$$sim(d_j, q) \sim \frac{P(d_j \mid R)}{P(d_j \mid \bar{R})}$$

- Annahme: Die Terme des Dokumente sind unabhängig:

$$P(d_j \mid R) = \left(\prod_{i:w_{ij}=1} P(k_i \mid R) \right) \left(\prod_{i:w_{ij}=0} P(\bar{k}_i \mid R) \right)$$

Probabilistisches Modell

- Dann folgt für sim:
- $$sim(d_j, q) \sim \frac{\left(\prod_{i:w_{ij}=1} P(k_i \mid R)\right)\left(\prod_{i:w_{ij}=0} P(\bar{k}_i \mid R)\right)}{\left(\prod_{i:w_{ij}=1} P(k_i \mid \bar{R})\right)\left(\prod_{i:w_{ij}=0} P(\bar{k}_i \mid \bar{R})\right)}$$
- $P(k_i|R)$: Wahrscheinlichkeit für Anfrageterm k_i in relevanten Texten.
- $P(\bar{k}_i|R)$: Wahrscheinlichkeit dafür, dass Anfrageterm k_i in relevantem Text nicht auftritt.
- Logarithmiert und leicht umgeformt:

$$sim(d_j, q) \sim \sum_{i=1}^{t} w_{iq} \cdot w_{ij} \cdot \left(\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid \bar{R})}{P(k_i \mid \bar{R})} \right)$$

Probabilistisches Modell

- Problem: $P(k_i \mid R)$ ist nicht bekannt.
- Muss irgendwie von eingestellt oder mit vielen Heuristiken aus Daten zusammengebastelt werden.