Modèle Booléen

Principe

Requêtes sous forme d'expressions booléennes (AND, OR, NOT). Correspondance binaire exacte.

Index Inversé

Structure fondamentale : Dictionnaire + Listes d'occurrences

```
terme → [doc1, doc2, doc4, ...]
```

Construction :
1. Tokenisation
2. Normalisation
3. Stemming (optionnel)
4. Stop words (optionnel)
5. Tri par termes puis docID
6. Création dictionnaire + postings

Complexité : AND/OR en O(x+y) avec listes triées

Index Positionnel

Format :

```
<terme, nb_docs;
  doc1: pos1, pos2...;
  doc2: pos1, pos2...>
```

Avantages :
- Requêtes de phrase : "Applied Science"
- Requêtes de proximité : "bank /3 scandal"

Taille : 2-4× index non-positionnel (35-50% texte original)

Modèle Vectoriel

Représentation

Documents et requêtes = vecteurs dans ℝ^M (M = taille vocabulaire)

$$\boldsymbol{d} = (w_{1,d}, w_{2,d}, ..., w_{M,d})$$

Modèle du "sac de mots" : ordre non considéré, vecteurs creux

Term Frequency (TF)

Fréquence logarithmique (recommandée) :

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{TF}_{t,d} & \text{si TF} > 0 \\ 0 & \text{sinon} \end{cases}$$

Inverse Document Frequency (IDF)

Pondération selon rareté du terme :

$$\mathrm{IDF}_t = \log_{10}\left(\frac{N}{\mathrm{DF}_t}\right)$$

Où N = total documents, DF_t = docs contenant t

Interprétation :
- DF élevé → terme général → IDF faible
- DF bas → terme rare → IDF élevé

Pondération TF-IDF

Formule :

$$w_{t,d} = \left(1 + \log \mathrm{TF}_{t,d}\right) \times \log_{10}\left(\frac{N}{\mathrm{DF}_t}\right)$$

Propriétés :
- ↑ avec occurrences dans document (TF)
- ↑ avec rareté du terme (IDF)

Smart notation : `[TF].[DF].[Norm]`
- TF : n (natural), l (log), a (augmented), b (binary)
- DF : n (none), t (idf), p (prob idf)
- Norm : n (none), c (cosine)
- Standard : `lnc.ltc`

Normalisation

Normalisation L2 :

$$\boldsymbol{d}_{\text{norm}} = \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}$$

$$\|\boldsymbol{d}\| = \sqrt{\sum_i w_{i,d}^2}$$

Neutralise l'effet de la longueur du document

Similarité Cosinus

Formule générale :

$$\mathrm{sim}(\boldsymbol{q}, \boldsymbol{d}) = \cos(\theta) = \frac{\boldsymbol{q} \cdot \boldsymbol{d}}{\|\boldsymbol{q}\| \times \|\boldsymbol{d}\|}$$

Avec normalisation :

$$\mathrm{sim}(\boldsymbol{q}, \boldsymbol{d}) = \sum_i w_{i,q} \times w_{i,d}$$

BM25

Formule :

$$\text{score} = \sum_{t \in q} \mathrm{IDF}_t \times \frac{\mathrm{TF} \times (k_1 + 1)}{\mathrm{TF} + k_1 \times \left(1 - b + b \times \frac{\mathrm{dl}}{\mathrm{avdl}}\right)}$$

Paramètres : $k_1 = 1.2$, $b = 0.75$
- dl = longueur document
- avdl = longueur moyenne

Traitement des Termes

Pipeline

Tokenisation → Normalisation → Stemming → Stop words → Indexation

Normalisation

Objectif : Cohérence documents/requêtes

Techniques :
- Casse (minuscules)
- Accents (café → cafe)
- Ponctuation (U.S.A. → USA)

Stemming vs Lemmatisation

Stemming : Coupe heuristique (automate → automat)

Lemmatisation : Forme dictionnaire (am, are, is → be)

Règle cruciale : Si stemming sur docs → aussi sur requêtes

Problèmes :
- Sur-stemming : universal, university → univers
- Sous-stemming : european, europe → différents

Stop Words

Définition : Mots fréquents, peu sémantiques (a, the, and, of)

Trade-off :
- Supprimer : 30-50% économie, pas de phrases
- Garder : requêtes de phrase possibles

Lois Statistiques

Loi de Zipf :

$$f_r \propto \frac{1}{r}$$

Fréquence inversement proportionnelle au rang

Loi de Heaps :

$$V = K \times N^{\beta} \quad (\beta \approx 0.4 - 0.6)$$

Vocabulaire croît sous-linéairement

Évaluation

Matrice de Confusion

	Pertinent	**Non Pert.**
Retourné	TP	FP
Non ret.	FN	TN

Métriques de Base

Précision (qualité) :

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Rappel (complétude) :

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

F-measure :

$$F_1 = \frac{2 \times P \times R}{P + R}$$

F-beta :

$$F_\beta = \frac{(1 + \beta^2) \times P \times R}{\beta^2 \times P + R}$$

Métriques de Classement

Précision @ K :

$$P@K = \frac{\text{Pertinents dans top-K}}{K}$$

R-Précision : $P@R$ où R = nb pertinents

Average Precision :

$$\mathrm{AP} = \frac{\sum_{k=1}^{n} P@k \times \mathrm{rel}(k)}{\text{Total pertinents}}$$

Mean Average Precision :

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AP}(q)}{Q}$$

Pertinence Graduée

DCG :

$$\mathrm{DCG}_k = \sum_{i=1}^{k} \frac{\mathrm{rel}_i}{\log_2}(i+1)$$

NDCG (normalisé [0,1]) :

$$\mathrm{NDCG}_k = \frac{\mathrm{DCG}_k}{\mathrm{IDCG}_k}$$

MRR (premier pertinent) :

$$\mathrm{MRR} = \frac{\sum_{q=1}^{Q} \frac{1}{\mathrm{rank}_q}}{Q}$$

Compromis

- Précision ↔ Rappel : inversement proportionnels
- Internautes : haute précision
- Chercheurs : haut rappel

Algorithmes

Ranking Vectoriel

```
1. Calculer TF-IDF pour requête
2. Pour chaque doc avec ≥1 terme :
   - Calculer sim(q, d)
3. Trier par sim décroissante
4. Retourner top-K
```

Requêtes Spéciales

Phrase :

```
1. Chercher chaque terme
2. Intersect positions
3. Vérifier consécutivité
```

Proximité :

```
"bank /3 scandal"
1. Chercher positions
2. Vérifier distance ≤ 3
```

Trade-offs

Index

Type	**Avantages**	**Inconvénients**
Non-positionnel	Compact	Pas de phrases
Positionnel	Phrases/ proximité	2-4× plus grand

Stemming

Choix	**Avantages**	**Inconvénients**
Avec	+rappel, compact	-précision
Sans	+précision	-rappel, large vocab

Stop Words

Choix	**Avantages**	**Inconvénients**
Supprimer	30-50% économie	Pas de phrases
Garder	Phrases OK	Index plus large

Normalisation

Approche	**Effet**	**Usage**
Agressive	+rappel, +faux pos.	Recherche large
Conservatrice	+précision, -rappel	Recherche précise

Stratégies

Haute Précision

Requêtes strictes (AND), TF-IDF élevé requis, Peu de résultats,Usage : recherche spécialisée

Haut Rappel

Requêtes larges (OR), Seuil bas, Beaucoup de résultats, Usage : e-discovery, brevets

Équilibre

BM25 avec params standards, Expansion requête modérée, Top-10 à top-100, Usage : recherche web

Formules Essentielles

TF-IDF :

$$w_{t,d} = \left(1 + \log \mathrm{TF}_{t,d}\right) \times \log\left(\frac{N}{\mathrm{DF}_t}\right)$$

Normalisation L2 :

$$w'_{t,d} = \frac{w_{t,d}}{\sqrt{\sum_{t' \in d} w_{t',d}^2}}$$

Cosinus avec normalisation :

$$\mathrm{sim}(\boldsymbol{q}, \boldsymbol{d}) = \frac{\sum_i w_{i,q} \times w_{i,d}}{\sqrt{\sum_i w_{i,q}^2} \times \sqrt{\sum_i w_{i,d}^2}}$$