Master 1 Informatique – Université Marne-la-Vallée (IGM)
05/02/2014 – Cours 2
Ingéniérie Linguistique

Espaces vectoriels et recherche d'information

UNIVERSITÉ
PARIS-EST
MARNE-LA-VALLÉE

Philippe Gambette

Sources du cours

- Cours de Matthieu Constant, *Ingéniérie Informatique 1*

http://igm.univ-mlv.fr/ens/Master/M1/2010-2011/IngenierieLinguistique1/cours.php

- Cours de Jean Véronis, *Informatique et Linguistique 1*

http://sites.univ-provence.fr/~veronis/cours/INFZ18/veronis-INFZ18.pdf

Plan

- Introduction
- Géométrie vectorielle
- Représentation des documents dans un espace vectoriel
- Recherche d'informations

Plan

- Introduction
- Géométrie vectorielle
- Représentation des documents dans un espace vectoriel
- Recherche d'informations

Introduction

Recherche d'informations

- Une requête :
  - en langue naturelle

  → Répondre à la question

  - Sous forme de mots clés

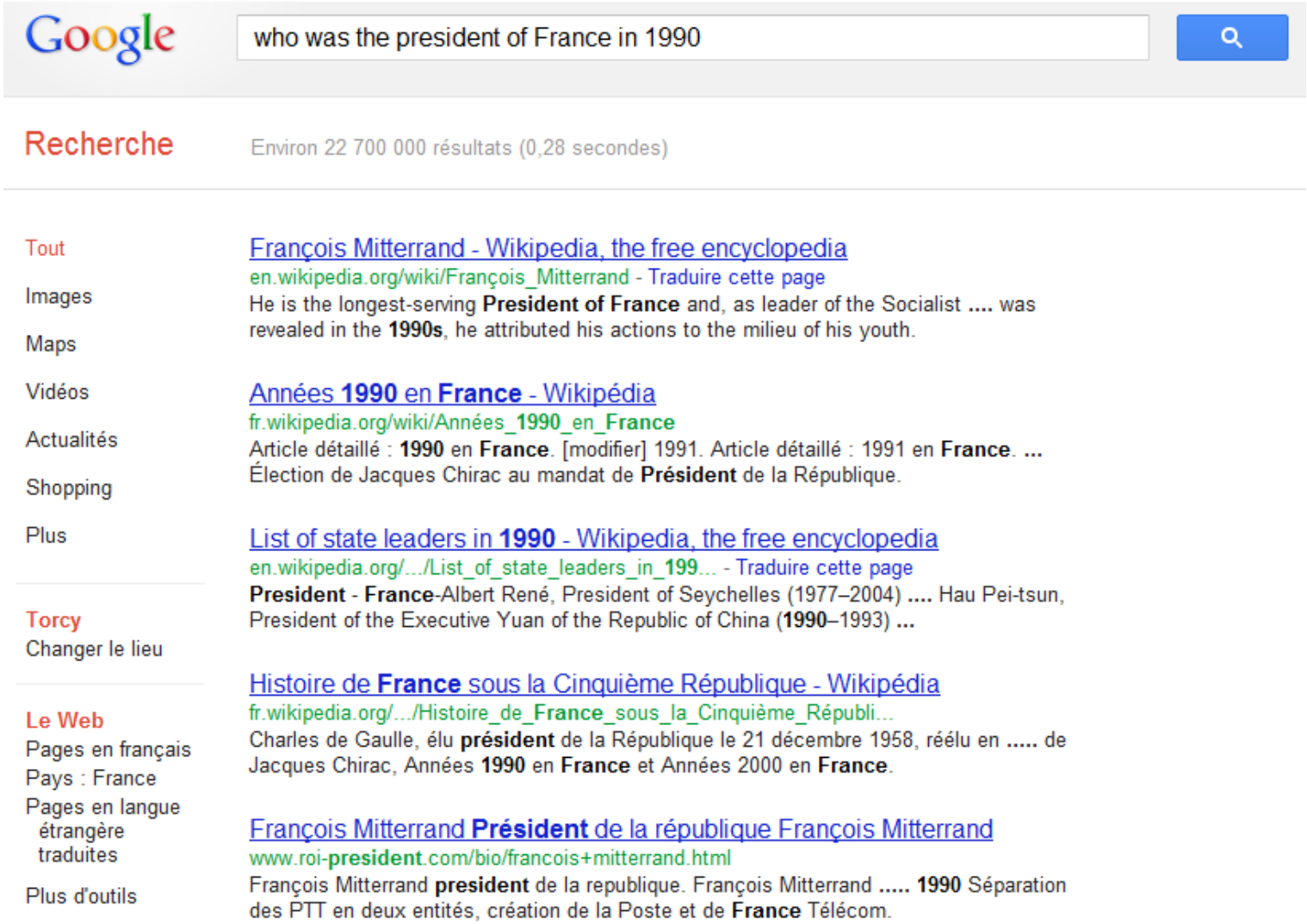
  → Trouver des documents pertinents pour y répondre

Introduction

Requête en langue naturelle : Wolfram Alpha

Introduction

Requête par mots-clés : Google

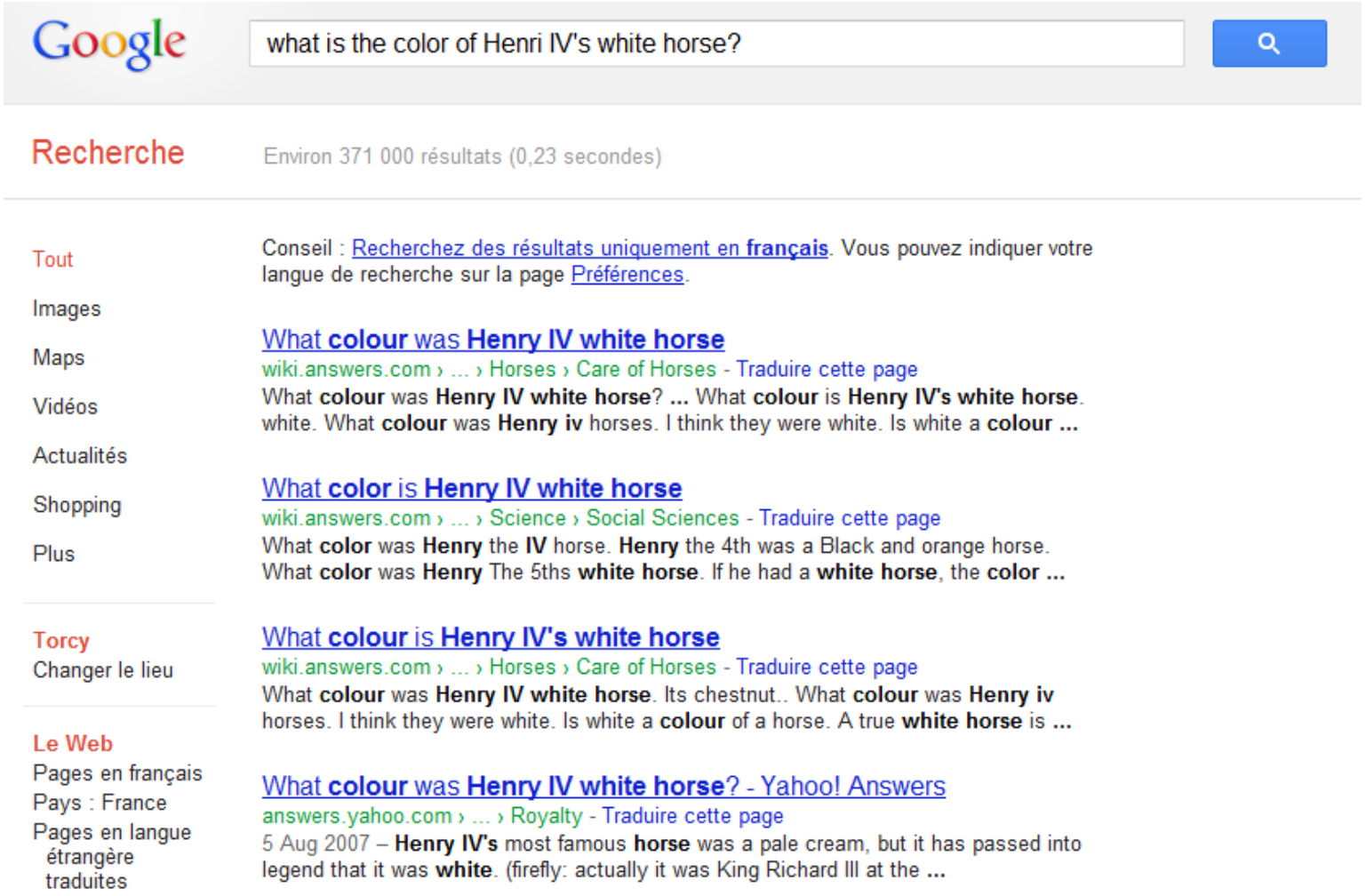


Introduction

Requête en langue naturelle : Wolfram Alpha

Introduction

Requête par mots-clés : Google



Sources

- Cours de Matthieu Constant, *Ingéniérie Informatique 1*

http://igm.univ-mlv.fr/ens/Master/M1/2010-2011/IngenierieLinguistique1/cours.php

- Cours de Marie Candito, *Recherche d'information, RI et TAL*

http://www.linguist.univ-paris-diderot.fr/~mcandito/Ens/RI/RI.RIetTAL.pdf

Plan

- Introduction
- Géométrie vectorielle
- Représentation des documents dans un espace vectoriel
- Recherche d'informations

Rappels sur les vecteurs en géométrie 2D

Un espace à deux dimensions

- Une origine
- Deux axes perpendiculaires avec des unités
- Un point défini par deux coordonnées sur ces axes

Un vecteur dans un espace à deux dimensions

- Définition informelle : une direction, un sens et une distance
- Définition formelle : deux coordonnées sur les axes

Calculs sur les vecteurs

Coordonnées

Soit un vecteur u défini par ses coordonnées (u_x, u_y).

Norme

- Longueur du vecteur : $|u| = \sqrt{u_x^2 + u_y^2}$

Produit scalaire

- Produit de deux vecteurs qui renvoie un nombre : $u.v = u_x v_x + u_y v_y$

Cosinus

- Dépend de l'angle formé entre les deux vecteurs : $\cos(u,v) = \frac{u.v}{|u||v|}$
- Comprise entre -1 et 1 pour angles entre 0 et 2π
- Comprise entre 0 et 1 pour les angles entre 0 et $\pi/2$

Généralisation à *n* dimensions

Coordonnées

Vecteur dans un espace à *n* dimensions : $(u_1, u_2, \ldots, u_n)$.

Norme

- Longueur du vecteur : $|u| = \sqrt{\sum_{i=1}^{n} u_i^2}$

Produit scalaire

- Produit de deux vecteurs qui renvoie un nombre : $u.v = \sum_{i=1}^{n} u_i v_i$

Cosinus

- $\cos(u,v) = \dfrac{u.v}{|u||v|}$

Plan

- Introduction
- Géométrie vectorielle
- Représentation des documents dans un espace vectoriel
- Recherche d'informations

Modèle des espaces vectoriels

Représentation simple des textes

• Un texte est un sac de mots (il n'y a plus d'ordre !)

• On associe à chaque mot un poids (nombre réel), mesurant son "importance" dans le texte

Application à la géométrie vectorielle

• Un texte est un vecteur dans un espace de grande dimension

• Chaque coordonnée correspond au degré d'importance d'un mot donné dans le texte

G. Salton, A. Wong, and C. S. Yang (1975), A Vector Space Model for Automatic Indexing, *Communications of the ACM* 18(11):613–620

Pondération des mots

Pondération naïve

• Poids binaire (1 si terme présent dans le document, 0 sinon)

• Fréquence du mot dans le document

Pondération plus intelligente

• On utilise des fonctions correctrices de la fréquence

• On prend aussi en compte la distribution du mot dans la collection où est plongée le texte

Pondération binaire

Collection de documents

d1 : we were anchored off an island in the bahamas

d2 : the couple traveled from island to island throughout the bahamas

Représentation vectorielle

	an	anchored	bahamas	couple	from	in	island	off	the	throughout	to	traveled	we	were
d1	1	1	1	0	0	1	1	1	1	0	0	0	1	1
d2	0	0	1	1	1	0	1	0	1	1	1	1	0	0

Pondération selon le nombre d'occurrences

Collection de documents

d1 : we were anchored off an island in the bahamas

d2 : the couple traveled from island to island throughout the bahamas

Représentation vectorielle

	an	anchored	bahamas	couple	from	in	island	off	the	throughout	to	traveled	we	were
d1	1	1	1	0	0	1	1	1	1	0	0	0	1	1
d2	0	0	1	1	1	0	2	0	2	1	1	1	0	0

Mesure TF.IDF

Principe

• On suppose que le texte traité est plongé dans une collection de documents

• Un mot pertinent d'un document apparaîtra plusieurs fois dans ce document

• Les mots non-pertinents sont distribués de manière homogène dans la collection

Exemple

terme	cf	df
insurance	10440	3997
try	10422	8760

statistiques sur des articles du *New York Times*
Manning et Schütze, 1999

Mesure TF.IDF

Fréquence des termes ou mots (TF)

$tf_{i,j}$: fréquence du mot *i* dans le document j de la collection

nombre d'occurrences du mot *i* dans le document *j*
normalisé par le nombre total de mots dans le document *j*

Fréquence inverse de document (IDF)

idf_i mesure l'importance d'un terme dans l'ensemble de la collection

$$idf_i = \log \frac{m}{D(i)}$$

avec *m* le nombre total de documents de la collection

et $D(i)$ le nombre de documents de la collection où le mot *i* apparaît

Formule TF.IDF

Le poids $d_{i,j}$ d'un mot *i* dans un document *j* est $d_{i,j} = tf_{i,j} . idf_i$

Représentation des documents en Python

Utilisation de dictionnaires

- clés : mots (dont le poids est non nul)
- valeurs : poids des mots

Classe Text et ses méthodes

- `def getName(self)`
- `def getWordTokens(self)`
- `def getWeight(self,w) # w : un mot`
- `def cosine(self,t) # t : un Text`

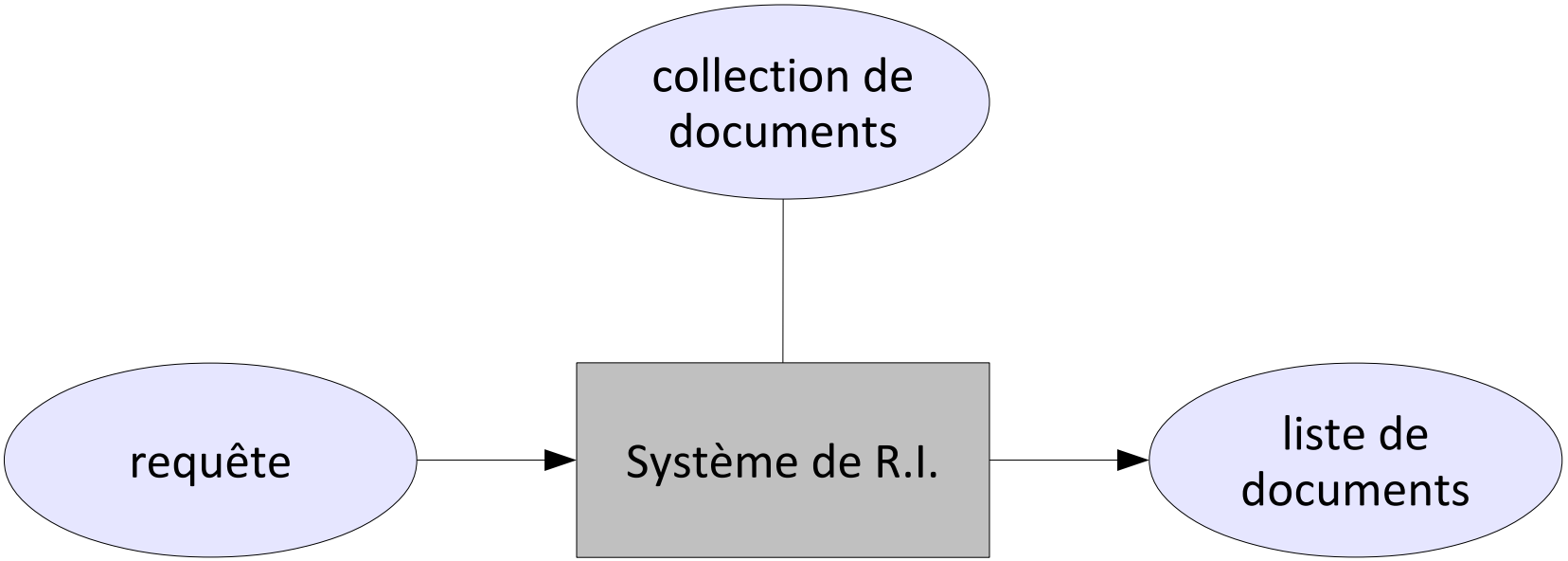
Exemple de code Python

```
# Calcul de la norme
import math
class Text:
def norm (self):
n = 0
for w in self.getWordTokens():
   n += self.getWeight(w) * self.getWeight(w)
return math.sqrt(n)
```

Plan

- Introduction
- Géométrie vectorielle
- Représentation des documents dans un espace vectoriel
- Recherche d'informations

Recherche d'informations



Recherche d'informations

Principe

• L'utilisateur entre une requête décrivant une information qu'il cherche

• Le système renvoie une liste de documents pertinents par rapport à la requête

Deux approches

• Recherche exacte (ex. systèmes booléens)

• Recherche floue (ex. modèles à espaces vectoriels)

Recherche d'informations et modèles à espaces vectoriels

Représentation

Requêtes (des séquences de mots) et documents de la collection représentés sous la forme de vecteurs

Métaphore entre proximité spatiale et proximité sémantique

• Les documents les plus pertinents sont ceux qui ont les vecteurs les plus proches de celui de la requête

• Les documents les plus pertinents contiennent des mots similaires à ceux de la requête

→ La mesure de la pertinence d'une requête par rapport à un document consiste à comparer leurs vecteurs respectifs :

ex. cosinus de leur angle

Exemple de requête

Requête q : “island couple”

	an	anchored	bahamas	couple	from	in	island	off	the	throughout	to	traveled	we	were
q	0	0	0	1	0	0	1	0	0	0	0	0	0	0
d1	1	1	1	0	0	1	1	1	1	0	0	0	1	1
d2	0	0	1	1	1	0	2	0	2	1	1	1	0	0

Pertinence des documents de la collection :

cos(q,d1) =

cos(q,d2) =

Exemple de requête

Requête q : “island couple”

	an	anchored	bahamas	couple	from	in	island	off	the	throughout	to	traveled	we	were
q	0	0	0	1	0	0	1	0	0	0	0	0	0	0
d1	1	1	1	0	0	1	1	1	1	0	0	0	1	1
d2	0	0	1	1	1	0	2	0	2	1	1	1	0	0

Pertinence des documents de la collection :

cos(q,d1)= 1/(1.41*3) = 0.2

cos(q,d2)= 3/(1.41*3.74) = 0.6

Exemple de code Python

```
# Moteur de recherche
# collection.lst est un fichier
# contenant la liste des fichiers de la collection
c = TextCollection('collection.lst')
while True :
    query = raw_input('Enter a query :')
    print 'RESULT :'
    print c.search(query,10)
```

Exemple de code Python

```
# Recherche des meilleurs documents
class TextCollection:
def search(self,query,N):
q = Text(text=query)
scores = {}
for t in self.getTexts():
   score[t.getName()] = t.cosine(q)
return sortScores(scores,N)
```

Évaluation des systèmes

Qualité d'un système de RI

Dans quelle mesure les documents pertinents sont retournés avant les documents non pertinents ?

Mesures traditionnelles

- **précision** : proportion de documents pertinents dans la liste retournée
- **rappel** : proportion de documents pertinents dans la collection qui sont dans la liste retournée (difficile à évaluer !)

Exemple

Evaluation	Ranking 1	Ranking 2	Ranking 3
	d0 : v	d9 : x	d5 : x
	d1 : v	d8 : x	d0 : v
	d2 : v	d7 : x	d1 : v
	d3 : v	d6 : x	d9 : x
	d4 : v	d5 : x	d8 : x
	d5 : x	d0 : v	d2 : v
	d6 : x	d1 : v	d4 : v
	d7 : x	d2 : v	d3 : v
	d8 : x	d3 : v	d6 : x
	d9 : x	d4 : v	d7 : x
Précision :	0.5	0.5	0.5

Manning et Schütze, 1999

Précision ou cutoff

Précision traditionnelle pas suffisante

- ne tient pas compte du rang du document
- ex. Ranking 1 est clairement meilleur que Ranking 2 !

Solution alternative : le cutoff

- On regarde la précision de segments initiaux plus petits
- Ex. on peut calculer la précision au rang 5

Exemple

Evaluation	Ranking 1	Ranking 2	Ranking 3
	d0 : v	d9 : x	d5 : x
	d1 : v	d8 : x	d0 : v
	d2 : v	d7 : x	d1 : v
	d3 : v	d6 : x	d9 : x
	d4 : v	d5 : x	d8 : x
	d5 : x	d0 : v	d2 : v
	d6 : x	d1 : v	d4 : v
	d7 : x	d2 : v	d3 : v
	d8 : x	d3 : v	d6 : x
	d9 : x	d4 : v	d7 : x
Précision à 10	0.5	0.5	0.5
Précision à 5	**1**	**0**	**0.4**

Manning et Schütze, 1999

Précision moyenne

Principe

• Précision calculée pour chaque point de la liste où l'on trouve un document pertinent

• Puis on fait la moyenne

Dans l'exemple :

• points pertinents : d0, d1, d2, d3 et d4

• précisions dans le Ranking 3 :
1/2 (d0), 2/3 (d1), 3/6 (d2), 4/7 (d4), 5/8 (d3)

• moyenne : 0.5726

Exemple

Evaluation	Ranking 1	Ranking 2	Ranking 3
	d0 : v	d9 : x	d5 : x
	d1 : v	d8 : x	d0 : v
	d2 : v	d7 : x	d1 : v
	d3 : v	d6 : x	d9 : x
	d4 : v	d5 : x	d8 : x
	d5 : x	d0 : v	d2 : v
	d6 : x	d1 : v	d4 : v
	d7 : x	d2 : v	d3 : v
	d8 : x	d3 : v	d6 : x
	d9 : x	d4 : v	d7 : x
Précision à 10	0.5	0.5	0.5
Précision à 5	1	0	0.4
P. moyenne			

Manning et Schütze, 1999

Exemple

Evaluation	Ranking 1	Ranking 2	Ranking 3
	d0 : v 1/1	d9 : x	d5 : x
	d1 : v 2/2	d8 : x	d0 : v 1/2
	d2 : v 3/3	d7 : x	d1 : v 2/3
	d3 : v 4/4	d6 : x	d9 : x
	d4 : v 5/5	d5 : x	d8 : x
	d5 : x	d0 : v 1/6	d2 : v 3/6
	d6 : x	d1 : v 2/7	d4 : v 4/7
	d7 : x	d2 : v 3/8	d3 : v 5/8
	d8 : x	d3 : v 4/9	d6 : x
	d9 : x	d4 : v 5/10	d7 : x
Précision à 10	0.5	0.5	0.5
Précision à 5	1	0	0.4
P. moyenne	**1**	**0.35**	**0.57**

Manning et Schütze, 1999

Quelques techniques complémentaires

Filtrage

- Parcours de l'ensemble des documents de la collection

  → coûteux

- Filtrage des documents non-pertinents par un index

Réduction de l'espace

- Racinisation des mots (ex. algorithme de Porter avec nltk)
- Filtrage des mots grammaticaux (ex. le, la, un, à, de, ...)

Quelques techniques complémentaires

Extension de la requête

• Ajout de synonymes à l'aide de ressources linguistiques

• Précision des requêtes avec de nouveaux mots (calculés à partir de statistiques de cooccurrence)

Algorithme pseudo-feedback

Vecteur de la nouvelle requête : $r' = \alpha r + \beta \frac{\sum_{d_j \in R} d_j}{|R|} - \gamma \frac{\sum_{d_j \in NR} d_j}{|NR|}$

• R, ensemble de documents pertinents

• NR, ensemble de documents non pertinents

• Pseudo-feedback : $\gamma = 0$

• Vrai feedback : $\gamma << \beta$

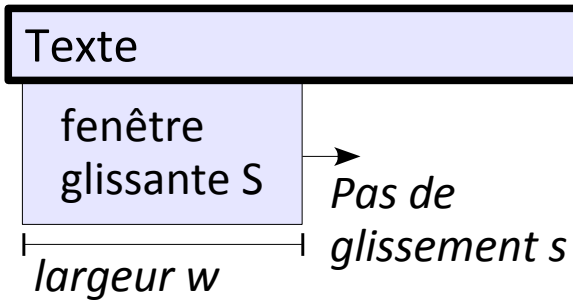
Quelques techniques complémentaires

Autres critères de recherche

- PageRank (Google)
- Positionnement des mots de la requête dans le document (ex. titre)
- Distance entre les mots de la requête dans le document
- ...

Distances de cooccurrence

Calcul de la matrice de distance entre mots



La dissimilarité sémantique entre deux mots *u* et *v* dépend du nombre de fenêtre *S* où ils apparaissent ensemble.

matrices de cooccurrence
O_{11}, O_{12}, O_{21}, O_{22}

Pour 2 mots *u* et *v*	$v \in S$	$v \notin S$
$u \in S$	O_{11}	O_{12}
$u \notin S$	O_{21}	O_{22}

matrice de dissimilarité sémantique
chi squared, mutual information, liddel, dice, jaccard, gmean, hyperlex, minimum sensitivity, odds ratio, zscore, log likelihood, poisson-stirling...

Evert, *Statistics of words cooccurrences*, thèse, 2005
Gambette, *User manual for TreeCloud,* 2009