Chapitre 5.2 :
Modèle de langue et RI
Language model

Plan

- Introduction au modèle de langue
  - Qu'est ce qu'un modèle de langue
  - Estimation d'un modèle de langue
- Modèle de langue et RI
  - Intuition LM et RI
  - Adaptation LM à la RI
  - Modèle vraisemblance de la requête (Query Likelihood)
  - Références bibliographiques

Modèle de langue

- Modèle de langue/language Model (modèle statistique de langue)
  - Modélise « l'agencement des mots dans une langue »
  - Capturer la distribution des mots dans une langue (ou d'un texte).
  - Mesure la probabilité d'observer une séquence de mots dans une langue
    - p1=P (un garçon mange une pomme)
    - p2=P (une pomme mange un garçon)
    - p3=P (apple mange un garçon)

- "The goal of a language model is to assign a probability to a sequence of words by means of a probability distribution"

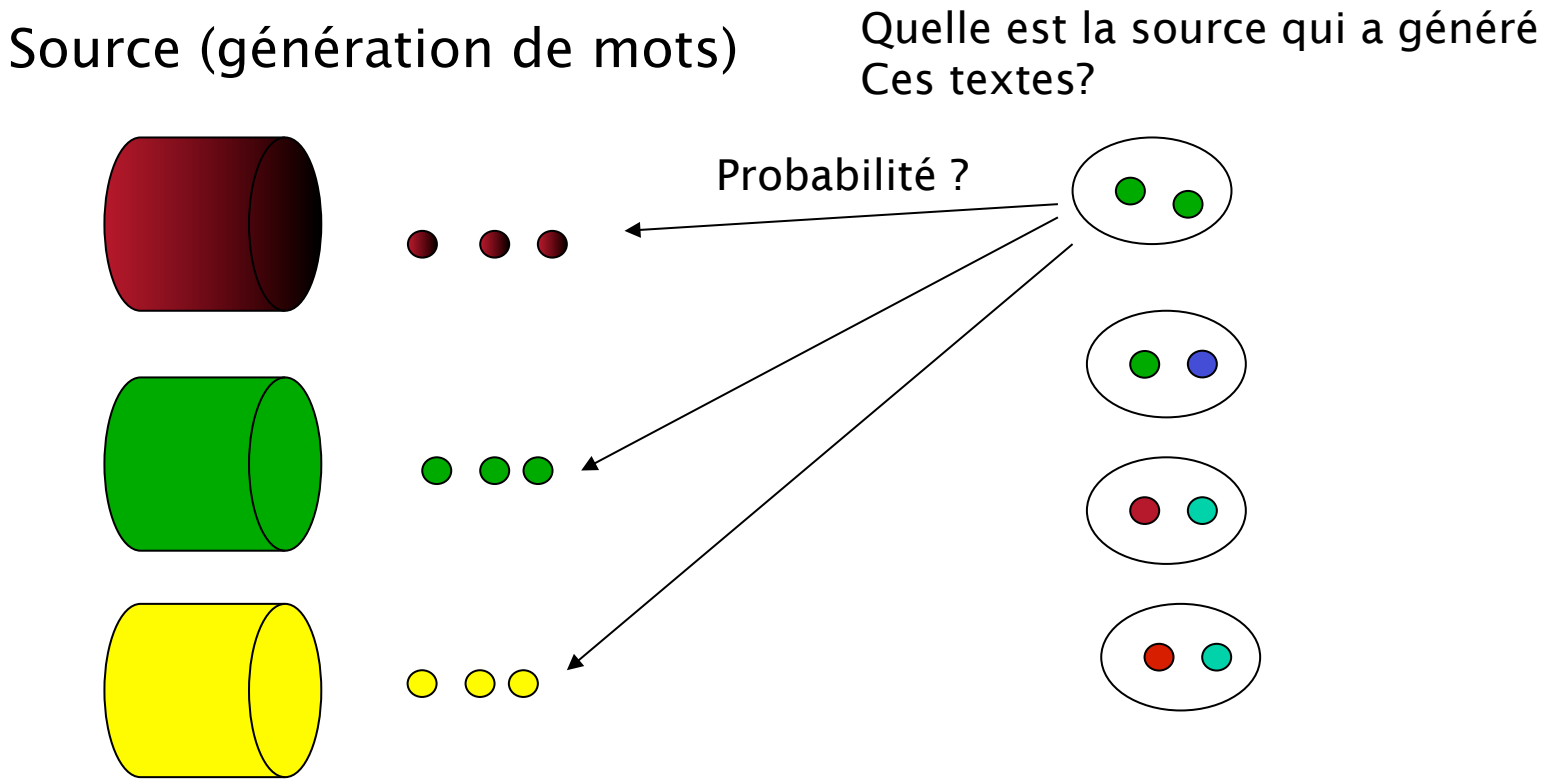
© wikipédia

Modèle de langue

- Utilisé dans plusieurs applications du traitement automatique de la langue :
  - speech recognition,
  - machine translation,
  - part-of-speech tagging,
  - parsing et information retrieval.

Modèle de langue

- Vu comme une source ou un générateur de textes
  - Mécanisme probabiliste de génération de texte (mots, séquence de mots) →On parle de modèle génératif



Modèle de langue

- Un modèle de langue est défini par son vocabulaire (mots simples, séquence de mots)
- Chaque mot (m)/séquence de mots($m_1 m_2 .. m_n$) a une probabilité d'être généré(e)
- Le but est de calculer → P(s|M)
  - s une observation (séquence de mots/texte) quelconque
  - Probabilité d'observer s dans le modèle (la langue) M

Définir un modèle de langue

- Définir la taille des séquences générées par le modèle ?
  → Séquence de 1 mot, 2 mots, 3 mots, …

- Estimer le modèle → probabilité de chaque séquence générée ?

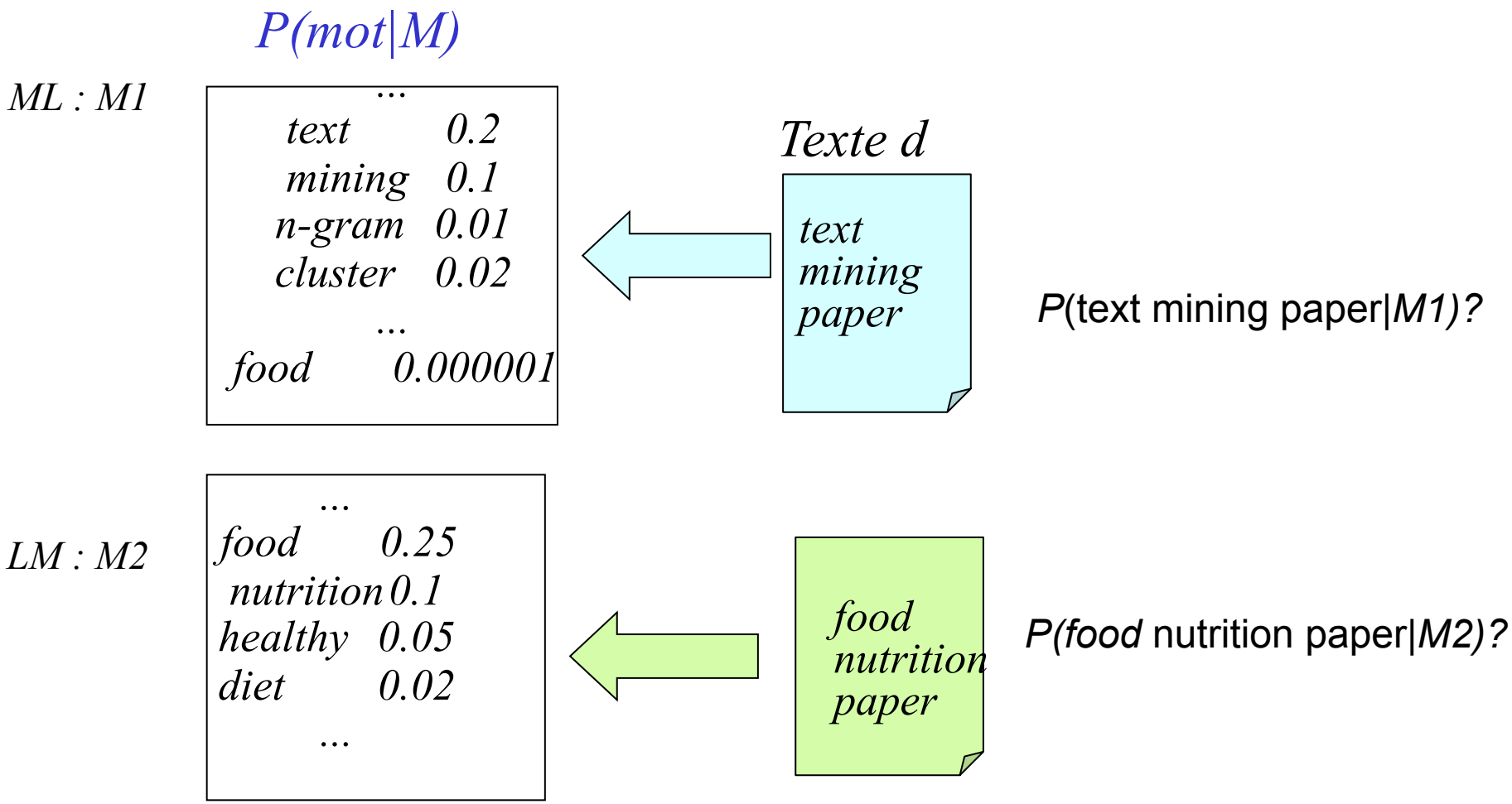
- Calculer la probabilité d'une observation (un texte) quelconque?

Taille de la séquence

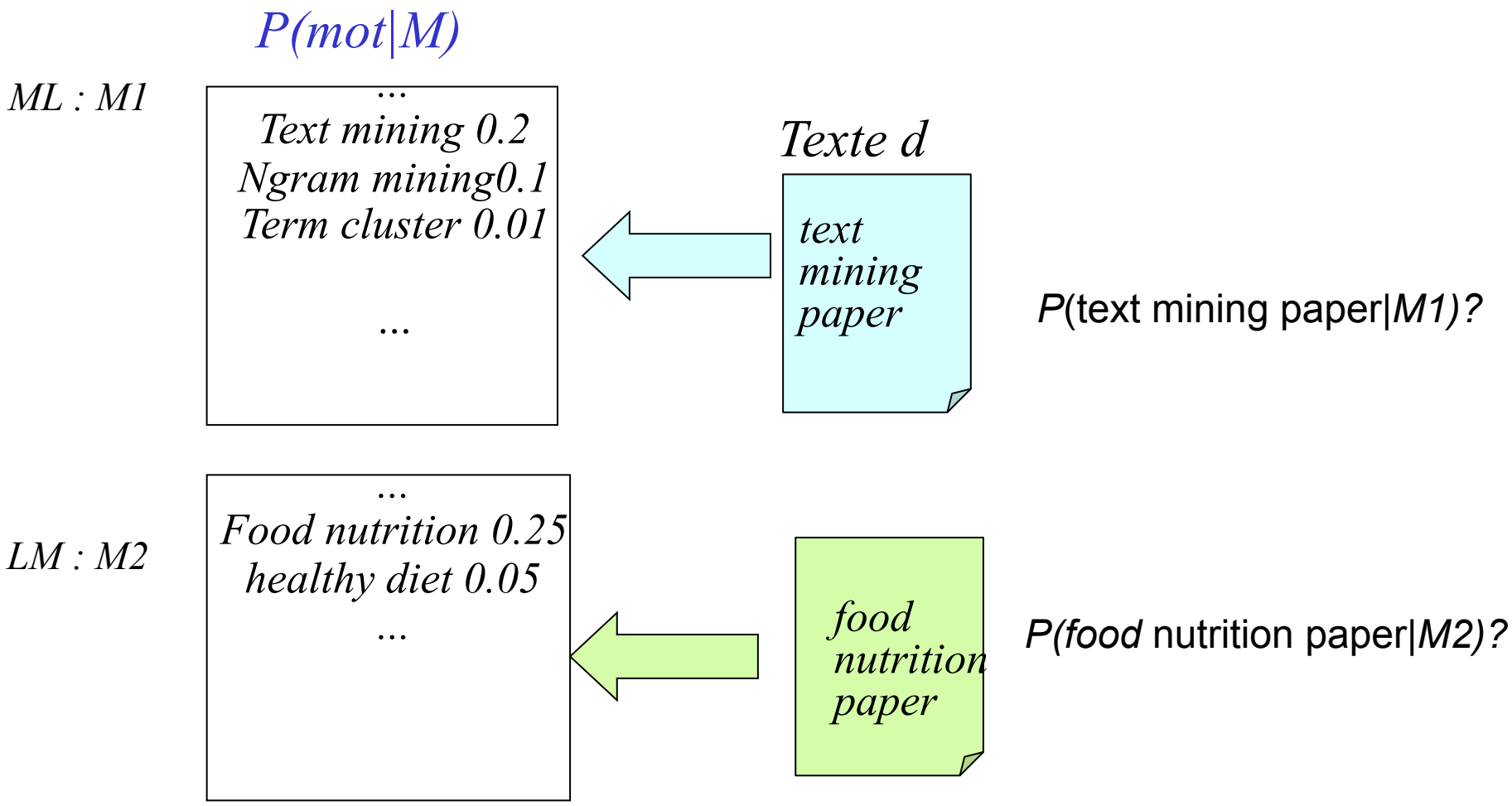
- Différents modèles
  - Séquence d'un mot → modèle *unigram*
  - Séquence de deux mots → modèle *bigram*
  - Séquence de n mots → modèle de *ngram*

- Dans le cas du modèle *unigram* (le plus utilisé en RI)
  - Les textes sont donc « générés » à partir de mots simples générés de manière indépendante les uns des autres
  - Si m_1, m_2, ..m_N est le vocabulaire (les mots acceptés) par le modèle alors
    - Chaque mot m a une probabilité →P(m|M)
    - $P(m_1)+P(m_2)+\ldots P(m_N)= 1$

Exemple de modèle unigram



Exemple de modèle bi-gram



Probabilité d'une observation (séquence)

- Dépend du modèle
  - soit s une observation (un texte) de n mots $s=m_1\ m_2 \ldots m_n$
  - Unigram – (M génère des séquences de 1 mot)

$$P(s \mid M) = P(m_1 m_2 \ldots m_n) = \prod_{i=1}^{n} P(m_i \mid M)$$

  - bigram – (M génère des séquences de deux mots)

$$P(s) = \prod_{i=1}^{n} P(m_i \mid m_{i-1}) = \prod_{i=1}^{l} \frac{P(m_{i-1} m_i)}{P(m_{i-1})}$$

  - ngram – (M génère des séquences de 3 mots)

$$P(s) = \prod_{i=1}^{n} P(m_i \mid m_{i-2} m_{i-1}) = \prod_{i=1}^{n} \frac{P(m_{i-2} m_{i-1} m_i)}{P(m_{i-2} m_{i-1})}$$

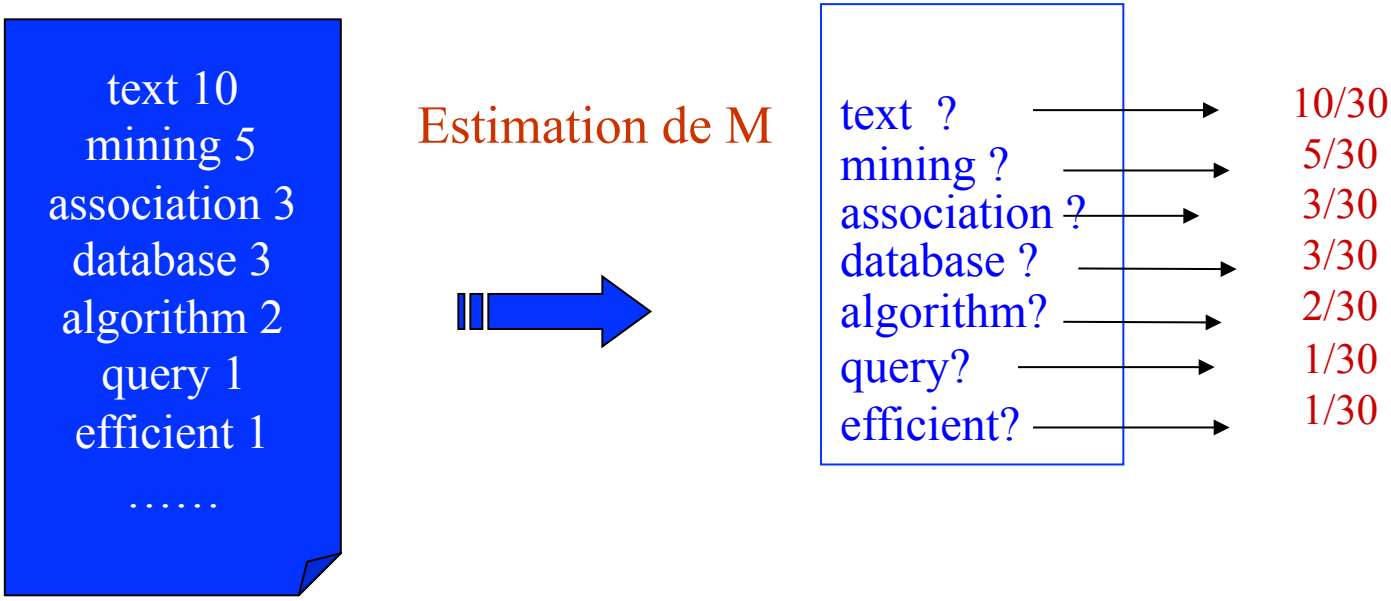
Estimation du modèle

- Selon le modèle il faut estimer
  - $P(m_i)$, $P(m_{i-1}\ m_i)$, $P(m_{i-2}\ m_{i-1}\ m_i)$, ..
- Estimation par Maximum de vraisemblance (*Maximum likelihood*) (la plus fréquente)
  - Compter la fréquence relative de l'événement (m) dans l'échantillon (C)
    - unigram $$P(m \mid C) = \frac{freq(m)}{\sum_{m \in C} freq(m)}$$
    - Bi gram $$P(m_i \mid m_{i-1}) = \frac{freq(m_{i-1}, m_i)}{freq(m_{i-1})}$$

Exemple

- Estimation d'un modèle uni-gram (simple) par ml
  - Compter la fréquence relative des mots m : $P_{ml}(m|M) = \#(m) / N$

Un texte sur
le "text mining"
(total #mots=30)



P("text query")=P(text)*P(query)=(10/30)*(1/30)

P("text retrieval")=P(text)*P(retrieval)=(10/30)*(0)

Problème des fréquences nulles (zéro)

- Si un événement (un mot de la séquence) n'apparait pas dans le modèle, le modèle lui assigne une probabilité 0

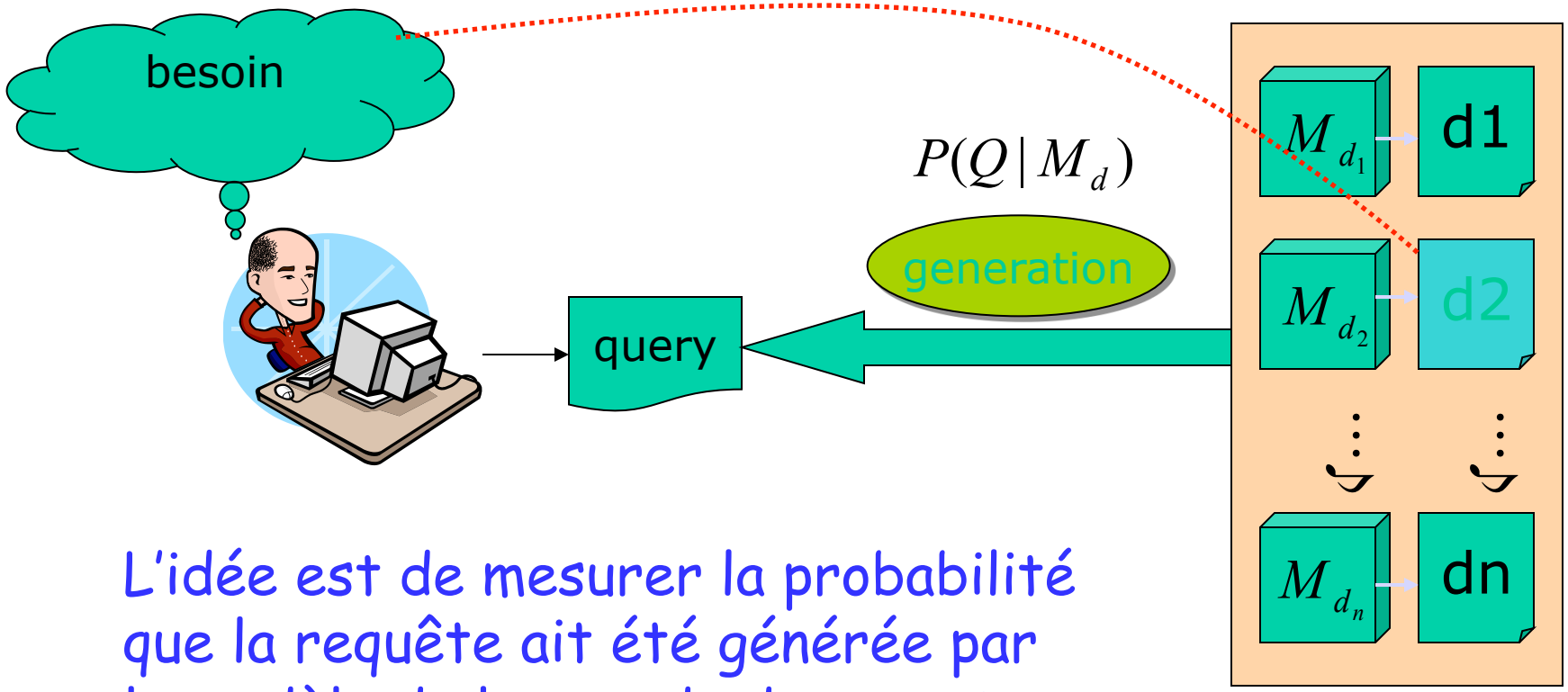
$$P(s \mid M) = \prod_{i=1}^{l} P(m_i \mid M) = 0, \quad si \quad \exists m_i \,/\, P(m_i \mid M) = 0$$

- Solution : assigner des probabilités différentes de zéro aux événements (mots) absents
  - → Lissage (Smoothing)

Modèle de Langue en RI

Plusieurs modèles, plusieurs adaptations

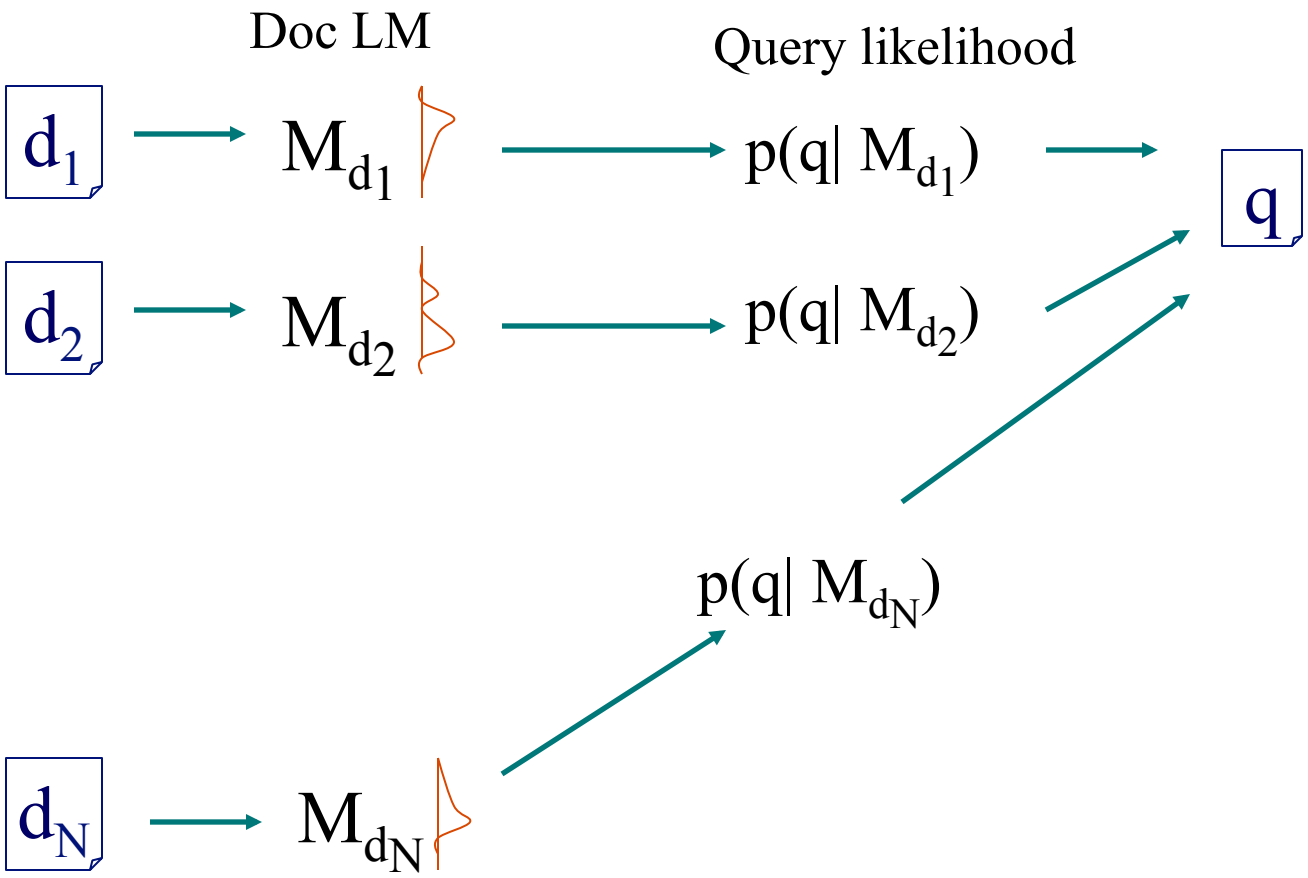
IR et LM : intuition



L'idée est de mesurer la probabilité que la requête ait été générée par le modèle de langue du document considéré

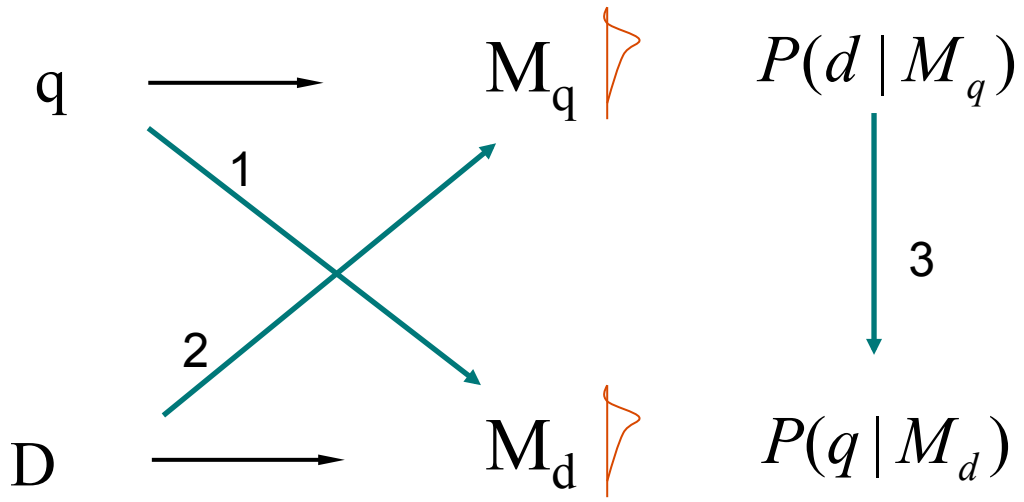
$P(Q|M_d)$

RI et ML : illustration



ML et RI : Plusieurs adaptations possibles

- Il existe plusieurs manières d'adapter les ML à la RI.



3 Principes

- (1) : Probabilité de générer la requête à partir de M_d
- (2) : Probabilité de générer le document à partir de Mq
- (3) : Combinaison (comparaison) des deux modèles

ML et RI : Plusieurs adaptations possibles (suite)

- Principe 1: Vraisemblance de la requête (Query-likelihood)
  - RSV(d,Q)= $P(Q|M_d)$
  - Document d représenté par son ML $P(w|M_d)$
  - Requête Q = séquence ou vecteurs de mots $q_1,q_2,\ldots,q_n$
- Principe 2 : Vraisemblance du document (Document likelihood)
  - RSV(d,Q)= P(D|Mq)
  - Requête Q représentée par son ML P(w|Mq)
  - Document d = séquence ou vecteurs de mots
- Principe 3: comparaison de modèles
  - Document d: LM $P(w|M_D)$
  - Requête Q: LM $P(w|M_Q)$
  - RSV(Q,d): comparer $P(w|M_d)$ and $P(w|M_Q)$

Principe 1 : Vraisemblance de la requête (Query Likelihood)

- Approche standard
  - Estimer le modèle de chaque document
  - Trier les documents selon leur probabilité de générer la requête → $P(Q|M_d)$
- Deux questions
  - Comment estimer M_d ?
    - Quel modèle ? → souvent unigram
    - Comment estimer $P(m|M_d)$?
  - Comment estimer $P(Q|M_d)$?

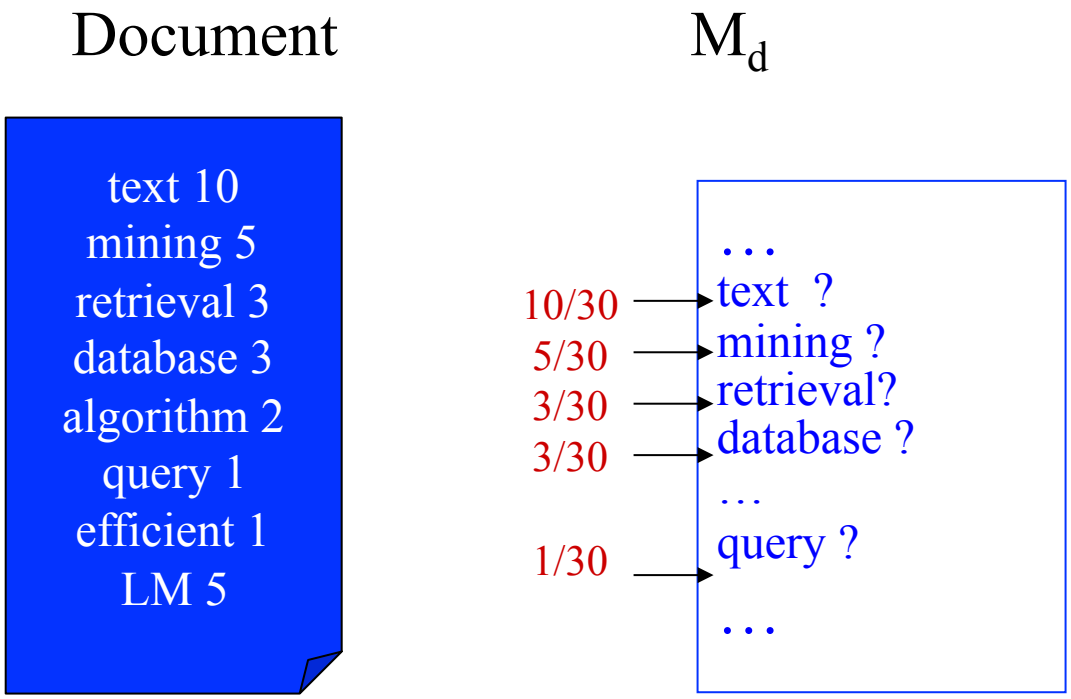
Estimation de M_d ?

- Le modèle de langue est inconnu mais, nous disposons d'un échantillon → le document
- Estimer le modèle à partir du document
  - Maximum de vraissemblance (Maximun Likelihood Estimator)

$$P_{mle}(t \mid M_d) = \frac{tf_{t,d}}{dl_d}$$

Les termes de la requête sont générés de manière indépendantes

Exemple


Document
M_d
text 10
mining 5
retrieval 3
database 3
algorithm 2
query 1
efficient 1
LM 5
…
10/30
text ?
5/30
mining ?
3/30
retrieval?
3/30
database ?
…
1/30
query ?
…

Estimation $P(Q|M_d)$

- Dépend de la « forme » du texte (requête) généré
  - Distribution multinomiale
    - Q est une séquence de variables aléatoires ($Q=(q_1,\ldots q_m)$) chaque v.a représente un mot t_i

$$P(Q=(q_1,\ldots,q_m)\,|\,M_d)=\prod_{i=1}^{m}P(q_i=t_i\,|\,M_d)=\prod_{i=1}^{m}P(q_i\,|\,M_d)$$

  - Distribution multi Bernoulli (modèle Ponte et Croft)
    - Q vecteur de v.a représentant TOUT le vocabulaire de M_d
    - $Q=(x_1,\ldots,x_{|V|})$, xi =1 terme t_i présent ; xi =0 terme absent

$$P(q=(x_1,\ldots,x_{|V|})\,|\,M_d)=\prod_{i=1}^{|V|}P(t_i=x_i\,|\,M_d)=\prod_{i=1;x_i=1}^{|V|}P(t_i=1\,|\,M_d)\prod_{i=1;x_i=0}^{|V|}P(t_i=0\,|\,M_d)$$

Estimation P(Q|M_d) (suite)

- Multinomiale ou Multi Bernoulli ?
  - Multinomiale semble être meilleure [Song & Croft 99, McCallum & Nigam 98,Lavrenko 04] que la Multi-Bernoulli [Ponte & Croft 98]

- Dans le cas d'une multinomiale (unigram)

$$P(Q \mid M_d) = P(q = (q_1,\ldots,q_m) \mid M_d) = \prod_{i=1}^{m} P(q_i = t_i \mid M_d)$$

$$= \prod_{i=1}^{m} P(q_i \mid M_d) = \prod_{i=1}^{m} \frac{tf_{w,d}}{dl_d}$$

- Cas d'une requête pondérée

$$P(Q \mid M_d) = P(q = (q_1,\ldots,q_m) \mid M_d) = \prod_{i=1}^{m} P(q_i = t_i \mid M_d)^{c(t_i,Q)}$$

$$\log(P(Q \mid M_d)) = \sum_{i=1}^{m} c(t_i,Q) * \log P(q_i = t_i \mid M_d)$$

Retour sur le problème des fréquences Zéro

- Problème des tf = 0
  - quand un document ne contient pas un ou plusieurs termes de la requête.
- Contraintes
  - On ne peut pas assigner des valeurs différentes de zéro de manière aléatoire
  - La somme des probabilités de l'ensemble des événements doit être égale à 1.
  - Plusieurs solutions ….

Techniques de lissage (Smoothing)

Techniques de lissage

- Méthodes de « discounting »
  - Laplace correction, Lindstone correction, absolute discounting, leave one-out discounting, Good-Turing method
- Techniques d'Interpolation
  - Estimations de Jelinek-Mercer, Dirichlet

Méthodes de « discounting »

- Ajouter une constante (1, 0,5 ou ε) à toutes les fréquences
  - Laplace smoothing
    - Ajouter 1 à tous les événements (n-gram : s)

$$P_{add\_one}(t \mid M_d) = \frac{tf_{t,d} + 1}{\sum_{t \in V} (tf_{t,d} + 1)}$$

  - Lindstone Smoothing:
    - Ajouter ε puis normaliser

Méthodes de « discounting »

- Good-Turing
  - Ajuster la fréquence d'un mot (d'une séquence)

$$tf^* = (tf + 1)\frac{n_{t+1}}{n_t}$$

  - n_t = nombre de n-grams de frequence t (apparaissant t fois)
  - n_0 : nombre total de mots dans le corpus
  - tf : fréquence du mot ou de la séquence

Exemple

tf	n_t	tf*	P(w)
0	7514941065	0,00015	2,01E-14
1	1132844	0,46540	4,11E-07
2	263611	1,40679	5,34E-06
3	123615	2,38767	1,93E-05
4	73788	3,33753	4,52E-05
5	49254	4,36947	8,87E-05
6	35869	5,32928	1,49E-04

Lissage par interpolation

- Les méthodes de « discounting » traitent les mots qui n'apparaissent pas dans le corpus de la même manière. Or, il y a des mots qui peuvent être plus fréquents que d'autres
- Solution
  - Interpoler le modèle en utilisant d'autres sources d'évidence (par exemple la collection de documents)

Lissage par interpolation (suite)

- Interpolation (Jelinek-Mercer)
  - Combiner le modèle M avec un modèle plus général (Modèle de référence)

$$P_{JM}(t \mid M) = \lambda . P_{ML}(t \mid M) + (1-\lambda) P_{ML}(t / REF)$$

  - Pb. "Règlage" de λ

Interpolation (Jelinek-Mercer)

- En RI le modèle de référence peut être le modèle de collection

$$RSV(Q,d) = \prod_{t \in Q} ((1-\lambda) p(t \mid M_c) + \lambda p(t \mid M_d))$$

Modèle général (collection)

Modèle de document

$$p(t \mid M_c) = p(t) = \frac{total\_tf_t}{total\_tf\_col}$$

$total\_tf_t$: fréquence du terme dans la collection
total_tf : somme des fréquences de tous les termes de la collection

Lissage par interpolation (suite)

- Lissage de Dirichlet
  - Problème avec Jelinek-Mercer
    - Les documents longs seront privilégiés
  - Prendre en compte la taille de l'échantillon
  - Si N est la taille de l'échantillon et μ une constante

$$P_{Dir}(t \mid M) = \left(\frac{N}{N+\mu}\right).P_{ML}(t \mid M) + \left(\frac{\mu}{N+\mu}\right)P_{ML}(t / REF)$$

Lissage par interpolation (suite)

- Lissage de Dirichlet en RI

$$P_{Dir}(t \mid d) = \frac{|d|}{|d| + \mu} \times \frac{tf(t,d) +}{|d|} + \frac{|\mu|}{|d| + \mu} P_{ML}(t \mid C)$$

$$P_{Dir}(t \mid d) = \frac{tf(t,d) + \mu P_{ML}(t \mid C)}{|d| + \mu}$$

Meilleure méthode de lissage?

- Dépend des données et de la tâche
- Dirichlet semble bien fonctionner pour la RI

Il existe d'autres méthodes de lissage
Voir [Chen & Goodman 98]

Exemple

- (2 documents)
  - d_1: Xerox reports a profit but revenue is down
  - d_2: Lucent narrows quarter loss but revenue decreases further
- Requête: *revenue down*
- MLE unigram;
  - Lissage JM $\lambda = ½$
  - Lissage Dir, $\mu = ½$

Principe 2 : Vraisemblance du document

- Chaque requête est traitée comme un modèle de langage
- Estimer le modèle de langage M_q de chaque requête
- Classer les documents

$$P(D \mid M_Q) = \prod_{t \in Q} P(t \mid M_Q)$$

- M_q peut être vu comme un modèle qui estime le document pertinent type

Principe 3: comparaison de modèles

- Combiner les avantages de deux méthodes de tri des documents
  - Estimer le modèle de requête M_Q et celui du document M_d puis comparer les modèles
  - Mesure naturelle de la similarité entropie croisée

$$H(M_q \| M_d) = -\sum_t P(t / M_q) \log(P(t / M_d))$$

  - Autre mesure Kullback-Leiblar divergence

$$RSV(Q,d) = H(M_Q \| M_d) - H(M_Q \| M_Q)$$

$$RSV(Q,d) = \sum_t P(t / M_q) \log \frac{P(t / M_d)}{P(t / M_q)}$$

Résumé choix LM pour la RI

- Choisir un modèle unigram
  - pas besoin d'aller au-delà des mots simples
- Choisir un modèle multinomial
  - Simple et performant vis-à-vis des autres modèles
- Choisir les modèles basés sur le principe 3
  - Permettent d'intégrer le feedback, expansion
- Estimation M_d et M_q une des questions importantes en LM

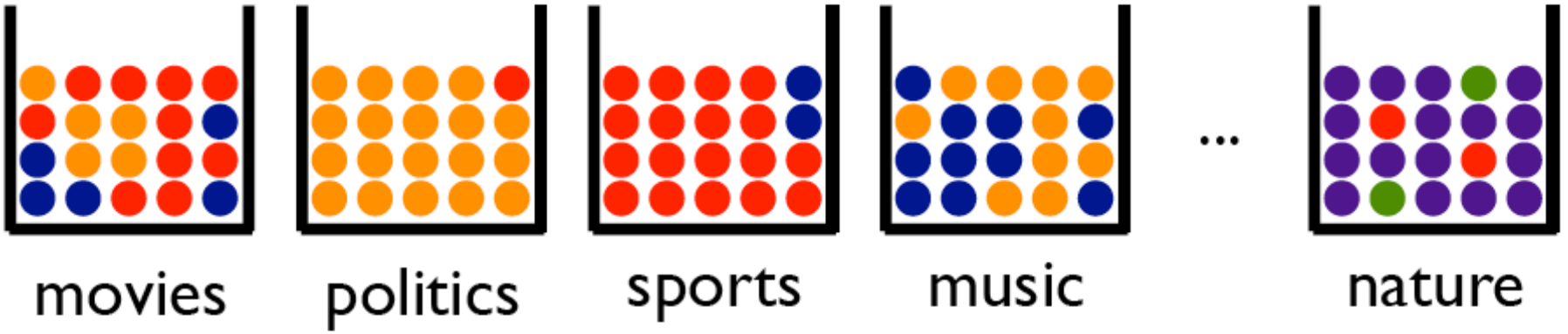
Relevance Model

Slides empruntés de la présentation de Jaime Arguello, « Pseudo-Relevance Feedback (and Document Priors) »

Slides de Jaime Arguello

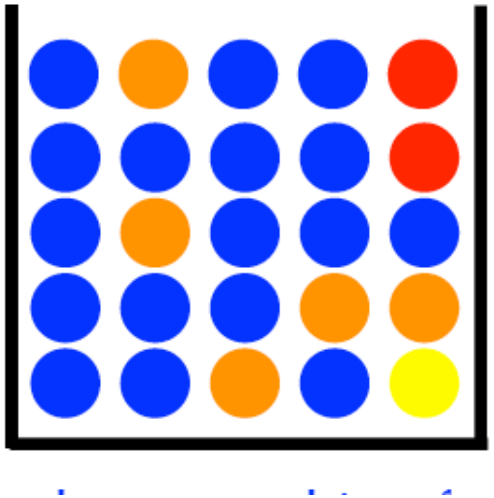
Relevance Model

- We can think of a language model as defining a topic
- Some words have a high probability and some have a low probability



Relevance Language Model

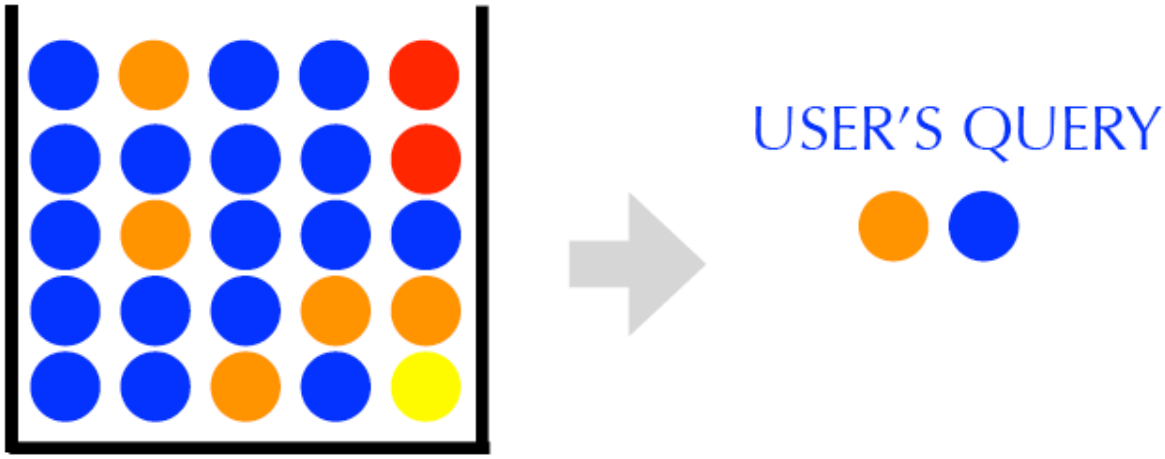
- Relevance Model
  - Let's consider a query : « Unix for dummies »
  - Let's assume that there exists a « Unix for dummies » language model out there. We can't see it, but it exists.



« Unix for dummies » LM

Relevance Model

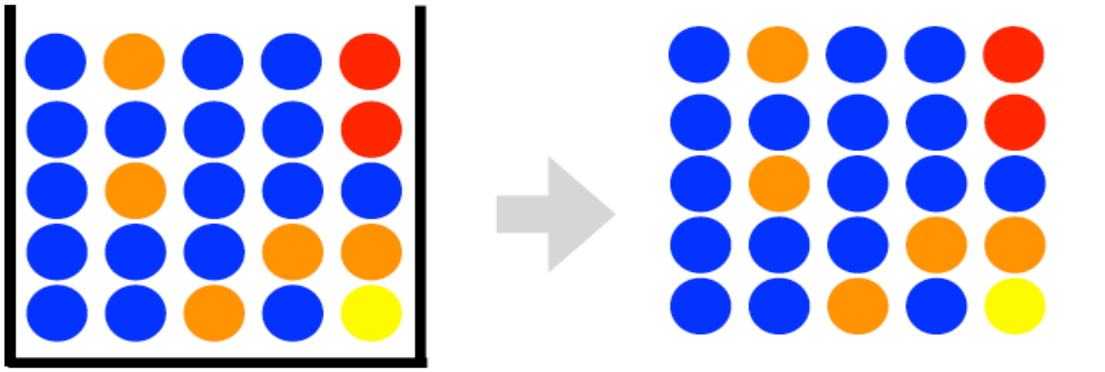
- We will call this the relevance model
- Assumption: the user's query was generated from this relevance model



Slides de Jaime Arguello

Relevance Model

- General Idea: we can improve retrieval by using the relevance model as the query

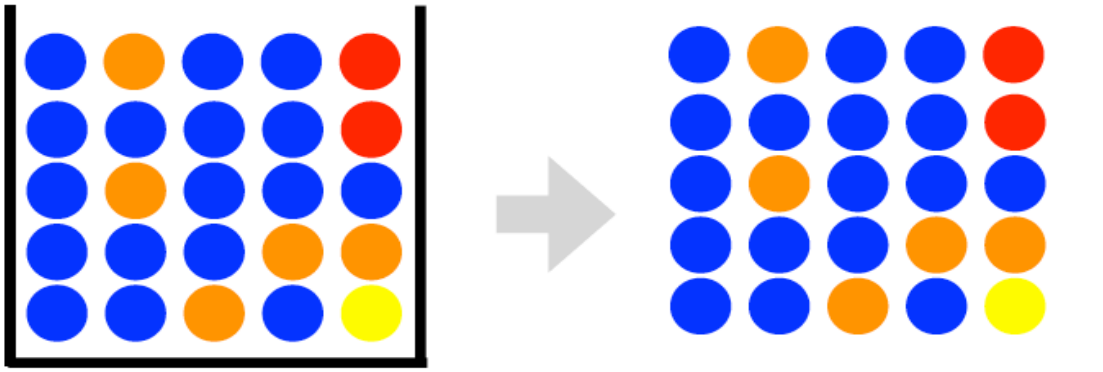


« Unix for dummies »
Language model »

RELEVANCE MODEL
QUERY

Relevance Model

- Why?
  - Because the observed query is only a sample. The relevance model is a more complete representation of the topic

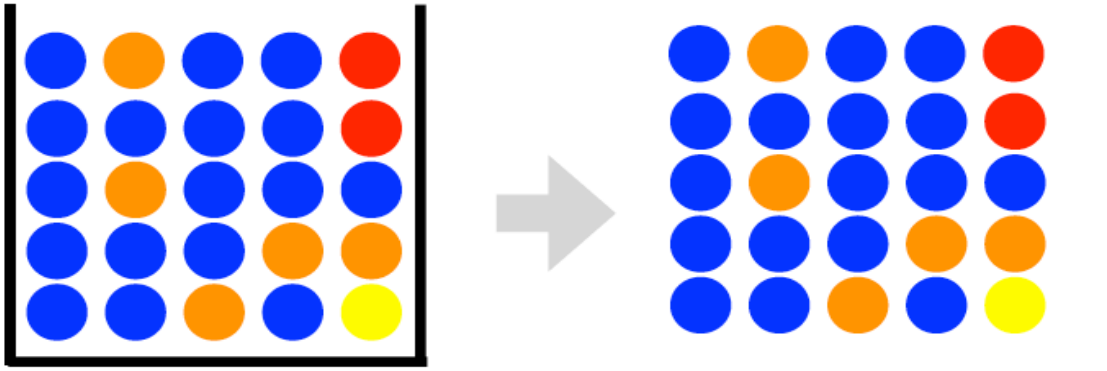


« Unix for dummies »
Language model »

RELEVANCE MODEL
QUERY

Relevance Model

- Problem:
  - how to estimate the relevance model?
- One Solution:
  - It can be estimated using the documents retrieved by the observed query

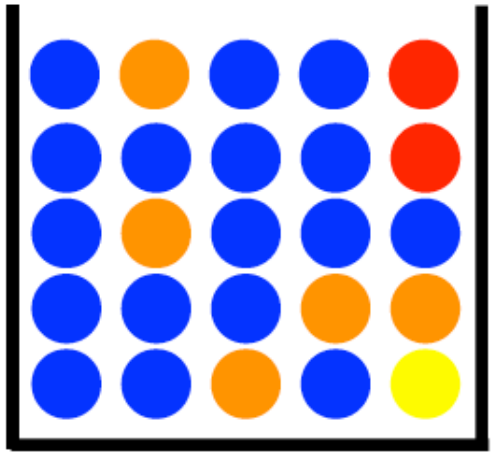


« Unix for dummies »
Language model »

RELEVANCE MODEL
QUERY

Relevance Language Model

$$P(w|\theta_R)$$



« Unix for dummies » LM

Relevance Language Model

$$P(w/\theta_R) = \sum_{D \in R} P(w/\theta_D) \frac{P(Q/\theta_D)}{\sum_{D \in R} P(Q/\theta_D)}$$

the probability of word *t* given the document language model

the document's query likelihood Score (but, normalized so that they sum to one over all documents in the collection C)

Slides de Jaime Arguello

Relevance Language Model

- 1. Given the user's observed query, conduct a retrieval using the query-likelihood model
- 2. Estimate the relevance model $P(w|\theta_R)$ (previous slide)
- 3. Conduct a second retrieval in which documents are scored according to:

$$RSV(Q,D) = \prod_{w \in R} P(w / \theta_D)^{P(w|\theta_R)}$$

Translation model

Translation Model

- Modeling the « translation » relationship between words in the query and words in a document

$$RSV(Q,D) = \prod_{w \in q} \sum_{w_j \in V} P(q_i / w_j) P(w_j / D)$$

Translation model (modèle de traduction)
à estimer via un dictionnaire par exemple,
…

Document prior

Retour sur le principe du Query Likelihood

- Règle de Bayes

$$P(D|Q) = \frac{P(Q|D) \times P(D)}{P(Q)}$$

Qu'est ce qu'on fait de ces probabilités. Ont-elles un impact sur le ranking

Retour sur le principe du Query Likelihood

- P(Q) est le même quelque soit le document
  - → n'intervient pas dans le ranking
  - → On l'ognore

$$P(D|Q) = \frac{P(Q|D) \times P(D)}{P(Q)}$$

$$P(D|Q) \propto P(Q|D) \times P(D)$$

Vraisemblance de la requête (déjà vu)

Document prior (probabilité à priori du document)

Document prior

$$P(D|Q) \propto P(Q|D) \times P(D)$$

Document prior (probabilité à priori du document)

- document prior, P(D), est la probabilité que le document soit pertinent pour n'importe quelle requête
- Probabilité spécifique à la requête
- Indépendante de la requête

Document prior

- Souvent considérée uniforme, la même , quelque soit le document
- → Nous allons coonsidér, que P(D) n'est plus uniforme
  - Il y a des documents qui sont vraisemblablement plus pertinents indépendamment de la requête

Document prior

- Qu'est ce que ca pourrait être :

→Tout ce qui peut affecter la pertinence du document

  - Popularité du document (nb de clicks sur le document, …
  - Autorité du document (nombre de liens entrant, nb tweets, …)
  - Longueur du document
  - Cohésion du sujet

  - ...

Document prior

- C'est une probabilité
  - On peut l'estimer par

$$P(D) = \frac{score(D)}{\sum_{\forall D_i} score(D_i)}$$

La fonction Sscore mesure :

la popularité, l'autorité, confiance…,

Representative LMs for IR

Basic LM (Query Likelihood)

- Query likelihood scoring — Ponte & Croft 98; *Hiemstra & Kraaij 99; Miller et al. 99*
- Parameter sensitivity — *Ng 00*
- Smoothing examined — *Zhai & Lafferty 01a*
- Theoretical justification — *Lafferty & Zhai 01a,01b*
- Bayesian Query likelihood — *Zaragoza et al. 03.*
- URL prior — *Kraaij et al. 02*
- Two-stage LMs — *Zhai & Lafferty 02*
- Time prior — *Li & Croft 03*

Improved Basic LM

- Beyond unigram — *Song & Croft 99*
- Translation model — *Berger & Lafferty 99*
- Term-specific smoothing — *Hiemstra 02*
- Title LM — *Jin et al. 02*
- Concept Likelihood — *Srikanth & Srihari 03*
- Cluster LM — *Kurland & Lee 04*
- Dependency LM — *Gao et al. 04*
- Cluster smoothing — *Liu & Croft 04; Tao et al. 06*
- Thesauri — *Cao et al. 05*

Query/Rel Model & Feedback

- Relevance LM — *Lavrenko & Croft 01*
- Model-based FB — *Zhai & Lafferty 01b*
- Markov-chain query model — *Lafferty & Zhai 01b*
- Rel. Query FB — *Nallanati et al 03*
- Parsimonious LM — *Hiemstra et al. 04*
- Pesudo Query — *Kurland et al. 05*
- Query expansion — *Bai et al. 05*
- Rebust Est. — *Tao & Zhai 06*

Special IR tasks

- Xu & Croft 99
- Xu et al. 01
- Lavrenko et al. 02
- Zhang et al. 02
- Cronen-Townsend et al. 02
- Si et al. 02
- Ogilvie & Callan 03
- Zhai et al. 03
- Shen et al. 05
- Tan et al. 06
- Kurland & Lee 05

Dissertations

- Ponte 98
- Hiemstra 01
- Berger 01
- Zhai 02
- Lavrenko 04
- Kraaij 04
- Srikanth 04
- Tao 06
- Kurland 06

Extensions LM

- Capturer les dépendances entre mots
  - Bigrams/Trigrams [Song & Croft 99]; dépendance grammaticale [Nallapati & Allan 02, Srikanth & Srihari 03, Gao et al. 04] (amélioration minime vis-à-vis du modèle de base)
- Modèle de traduction (Translation model)
  - Synonymie, stemming, cross-language
- Lissage basé sur les clusters (Cluster-based smoothing) [Liu & Croft 04, Kurland & Lee 04,Tao et al. 06]
  - Lisser les documents à partir de documents similaires (améliore le modèle de base)
- LM et Pertinence ?
  - Pertinence n'est pas explicitement représentée [Lafferty & Zhai, 2003]
  - Utilisation du feedback (construire un modèle de pertinence à partir des top-documents)
- LM et recherche d'image
- LM et recherche web (prise en compte des liens)

Références

M. Boughanem, W. Kraaij and J-Y. Nie. Modèles de langage pour la recherche d'information. 2004 (lavoisier)

Jing Bai, Dawei Song, Peter Bruza, Jian-Yun Nie, Guihong Cao, Query expansion using term relationships in language models for information retrieval, *Proceedings of ACM CIKM 2005*, pages 688-695.

Berger and J. Lafferty. *Information retrieval as statistical translation.* Proceedings of the ACM SIGIR 1999, pages 222-229.

A. Berger. *Statistical machine learning for information retrieval*. Ph.D. dissertation, Carnegie Mellon University, 2001.

Guihong Cao, Jian-Yun Nie, Jing Bai, Integrating word relationships into language models, Proceedings of ACM SIGIR 2005, Pages: 298 - 305.

S. F. Chen and J. T. Goodman. *An empirical study of smoothing techniques for language modeling.* Technical Report TR-10-98, Harvard University.

W. B. Croft and J. Lafferty (ed), *Language Modeling and Information Retrieval*. Kluwer Academic Publishers. 2003.

J. Gao, J. Nie, G. Wu, and G. Cao, Dependence language model for information retrieval, In *Proceedings of ACM SIGIR 2004*.

D. Hiemstra and W. Kraaij, Twenty-One at TREC-7: Ad-hoc and Cross-language track, In *Proceedings of the Seventh Text REtrieval Conference (TREC-7),* 1999.

D. Hiemstra. *Using Language Models for Information Retrieval*. PhD dissertation, University of Twente, Enschede, The Netherlands, January 2001.

D. Hiemstra. Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term. In *Proceedings of ACM SIGIR 2002*, 35-41

D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval, In *Proceedings of ACM SIGIR 2004*.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings on the 22nd annual international ACM-SIGIR 1999,* pages 50-57.

F. Jelinek, *Statistical Methods for Speech Recognition*, Cambirdge: MIT Press, 1998.

F. Jelinek and R. L. Mercer. Interpolated estimation of markov source parameters from sparse data. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice*. 1980. Amsterdam, North-Holland

References (cont.)

J. Jeon, V. Lavrenko and R. Manmatha, *Automatic Image Annotation and Retrieval using Cross-media Relevance Models,* In Proceedings of ACM SIGIR 2003

T. Kalt. *A new probabilistic model of text classication and retrieval.* University of Massachusetts Technical report TR98-18,1996.

S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume ASSP-35:400--401.

W. Kraaij,T. Westerveld, D. Hiemstra: The Importance of Prior Probabilities for Entry Page Search. *Proceedings of SIGIR 2002*, pp. 27-34

W. Kraaij. *Variations on Language Modeling for Information Retrieval,* Ph.D. thesis, University of Twente, 2004,

J. Lafferty and C. Zhai, Probabilistic IR models based on query and document generation. In *Proceedings of the Language Modeling and IR workshop*, pages 1--5.

J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the ACM SIGIR 2001*, pages 111-119.

V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of the ACM SIGIR 2001*, pages 120-127.

V. Lavrenko, M. Choquette, and W. Croft. Cross-lingual relevance models. In *Proceedings of SIGIR 2002*, pages 175-182.

V. Lavrenko, *A generative theory of relevance*. Ph.D. thesis, University of Massachusetts. 2004.

X. Li, and W.B. Croft, Time-Based Language Models, In *Proceedings of CIKM'03*, 2003

X. Liu and W. B. Croft. Passage retrieval based on language models. In *Proceedings of CIKM 2002*, pages 15-19.

J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of ACM-SIGIR 1998*, pages 275-281.

. M. Ponte. *A language modeling approach to information retrieval*. Phd dissertation, University of Massachusets, Amherst, MA, September 1998.