UNIVERSITÉ DES SCIENCES ET TECHNOLOGIE HOUARI BOUMEDIENE

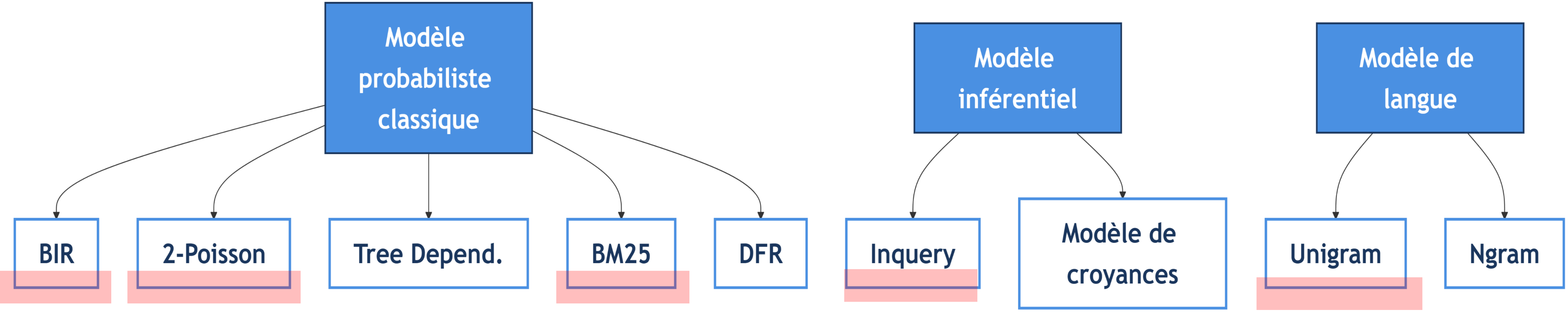
FACULTÉ D'INFORMATIQUE - MASTER 2 SII

RECHERCHE D'INFORMATION INFORMATION RETRIEVAL

CHAPITRE 9: Modèle de langue pour la Recherche d'Information

01 Décembre 2025

LA RECHERCHE D'INFORMATION BASE SUR LES PROBABILITÉS



I. INTRODUCTION AU MODÈLE DE LANGUE

- C'est un modèle statistique probabiliste de langue, vise à **modéliser l'agencement/ordre des mots dans une langue**, c'est-à-dire à **capturer la distribution des mots** dans un corpus ou une langue donnée **via des probabilités**.
- Il permet de **mesurer la probabilité d'observer une séquence de mots**.

Exemples :

$p_1 = P($« *un garćon mange une pomme* »$)$— séquence probable

$p_2 = P($« *une pomme mange un garćon* »$)$ — séquence grammaticalement correcte mais peu probable

$p_3 = P($« *apple mange un garćon* »$)$— séquence improbable car mélange de langues et incohérence sémantique

- **Définition générale :**

« *The goal of a language model is to assign a probability to a sequence of words by means of a probability distribution.* » — Wikipédia

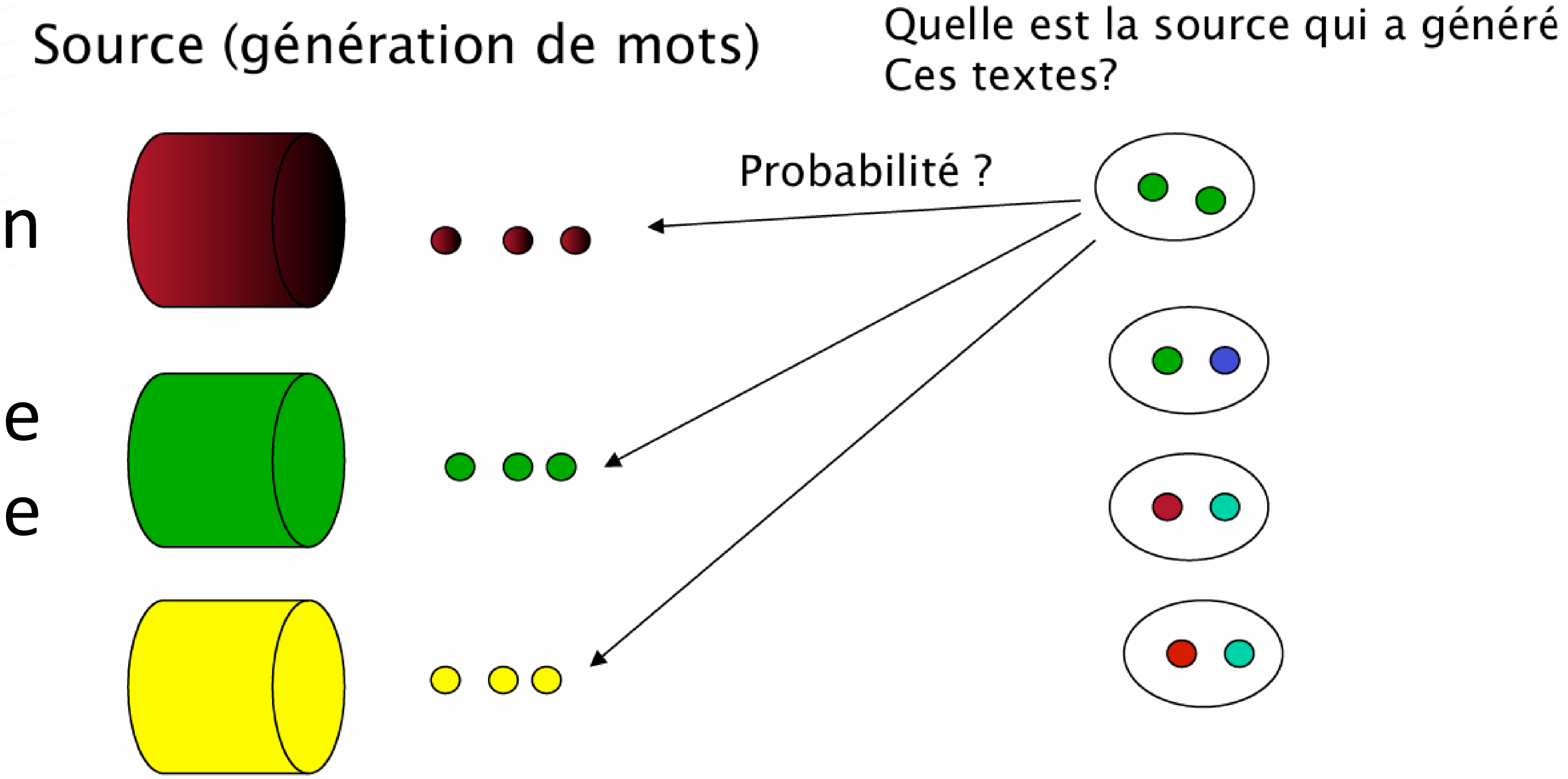
I. INTRODUCTION AU MODÈLE DE LANGUE

- Utilisé dans de nombreuses applications du traitement automatique de la langue, telles que :

– speech recognition,
– machine translation,
– part-of-speech tagging,
– parsing et information retrieval.

- Vu comme une source ou un générateur de textes

– Mécanisme probabiliste de génération de texte (mots, séquence de mots) modèle génératif.



I. INTRODUCTION AU MODÈLE DE LANGUE

Un **modèle de langue** est défini par :

- **son vocabulaire**, c'est-à-dire l'ensemble des mots (ou séquences de mots) qu'il peut reconnaître ou générer.

Dans ce modèle :

- **Chaque mot** m **ou séquence de mots** $(m_1 m_2 \dots m_n)$ possède une **probabilité d'être généré(e)**.
  Cela signifie que le modèle sait quels mots sont plus courants ou plus naturels dans la langue.

- Le but est de calculer : $P(s \mid M)$

où :

s = une séquence de mots (une phrase, un texte), **M** = le modèle de langue,
P(s | M) = la **probabilité que cette séquence s** soit observée ou produite selon le modèle M.

II. DÉFINIR UN MODÈLE DE LANGUE

Pour définir un modèle de langue, il faut répondre à plusieurs questions :

1. Quelle est la taille des séquences que le modèle va considérer ?

Le modèle peut travailler avec :

- des **séquences d'un mot** (modèle unigramme),
- des **séquences de deux mots** (bigramme),
- des **séquences de trois mots** (trigramme),

→Plus la séquence est longue, plus le modèle capture du contexte.

2. Comment estimer le modèle ?

Il faut **calculer la probabilité** de chaque séquence possible :

$$P(m_1),\ P(m_1, m_2),\ P(m_1, m_2, m_3), \ldots$$

3. Comment calculer la probabilité d'une observation (un texte) ?

Une fois le modèle appris, on peut calculer la probabilité d'un texte s en combinant les probabilités des séquences qui le composent. $P(s \mid M)$

II. DÉFINIR UN MODÈLE DE LANGUE

II.1. Taille de la séquence

Selon la taille des séquences de mots considérées, on distingue plusieurs types de modèles :

- **Séquence d'un mot → modèle unigram**
- **Séquence de deux mots → modèle bigram**
- **Séquence de n mots → modèle n-gram**

1) Cas du modèle unigram : (Le plus utilisé en Recherche d'Information)

✓ Un texte est considéré comme une suite de mots indépendants les uns des autres. Cela signifie que le modèle ne tient pas compte de l'ordre ni du contexte : chaque mot est généré séparément.

✓ Si le vocabulaire du modèle est composé des mots : $m_1, m_2, \dots, m_N$

Chaque mot m possède une probabilité associée : $P(m \mid M)$

La somme des probabilités de tous les mots du vocabulaire doit être égale à 1 :

$$P(m_1) + P(m_2) + \cdots + P(m_N) = 1$$

II. DÉFINIR UN MODÈLE DE LANGUE

II.1. Taille de la séquence

2) Cas du modèle bigram :

- ✓ On ne considère plus les mots comme indépendants ,**la probabilité d'un mot dépend du mot précédent.** On estime : $\boldsymbol{P(m_i \mid m_{i-1})}$

3) Cas du modèle n-gram :

- ✓ Plus généralement, un modèle n-gram considère :

$$P(m_i \mid m_{i-n+1}, \ldots, m_{i-1})$$

La probabilité d'observer le mot mi, en tenant compte des $n-1$ mots précédents.

- ✓ Plus n est grand, plus le modèle devient coûteux
- ✓ Les séquences longues deviennent rares → problème de sparsité : la plupart des séquences n'apparaissent jamais. (Pas très utile pour la RI)

II. DÉFINIR UN MODÈLE DE LANGUE

II.2. Probabilité d’une séquence (Observation) selon le modèle

1. Modèle Unigram

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

- Le modèle considère **chaque mot indépendamment des autres**.
- Il **ne regarde aucun contexte**.
- La probabilité du texte est simplement le **produit des probabilités de chaque mot**.

Exemple :

Si le texte est : ‘’ The data mining"

$$P(the) \times P(data) \times P(mining)$$

II. DÉFINIR UN MODÈLE DE LANGUE

II.2. Probabilité d'une séquence (Observation) selon le modèle

2. Modèle Bigram

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

- Le modèle regarde **un mot en arrière**.
- Chaque mot dépend **uniquement du mot précédent**.

Exemple :

Texte : "the data mining"

$$P(the) \times P(data \mid the) \times P(mining \mid data)$$

II. DÉFINIR UN MODÈLE DE LANGUE

II.2. Probabilité d'une séquence (Observation) selon le modèle

3. Modèle n-gram : ici n=3

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

Chaque mot dépend des **n-1 mots précédents.**

Exemple pour **trigram (3-gram)** :

$$P(\textit{data mining est très utile})$$
$$= P(\textit{data, mining}) \times P(\textit{est} \mid \textit{data, mining}) \times P(\textit{très} \mid \textit{mining, est})$$
$$\times P(\textit{utile} \mid \textit{est, très})$$

Ici, le modèle regarde **les deux mots précédents** pour prédire le mot suivant.

II. DÉFINIR UN MODÈLE DE LANGUE

II.3. Estimation des probabilités

Pour un modèle n-gramme, on doit estimer des probabilités du type :

- Unigram → $P(m_i)$
- Bigram → $P(m_i \mid m_{i-1})$
- Trigram → $P(m_i \mid m_{i-2}\, m_{i-1})$ ….

Ces probabilités sont estimées à partir des fréquences dans le corpus.

La méthode la plus utilisée pour estimer les probabilités d'un modèle de langage est :
L'estimation par Maximum de Vraisemblance (Maximum Likelihood Estimation — MLE)

Cas Unigram (n = 1)

$$P_{MLE}(m_i) = \frac{freq(m_i)}{\sum_{m \in V} freq(m)}$$

On compte combien de fois chaque mot apparaît dans le modèle de langue.

II. DÉFINIR UN MODÈLE DE LANGUE

Exemple Maximum de vraisemblance (Maximum Likelihood, ML)

On a un document de **100 mots** ($N = 100$).
Les fréquences observées sont :

Mot	Fréquence
text	10
mining	5
association	3
database	3
algorithm	2
query	1
efficient	1
…	…

Estimation →

Probabilité ML
(10/100 = 0.10)
(5/100 = 0.05)
(3/100 = 0.03)
(3/100 = 0.03)
(2/100 = 0.02)
(1/100 = 0.01)
(1/100 = 0.01)
…

ML(Unigramme) M
p(m| M)=?

Les mots sont indépendants entre eux → on considère uniquement leur fréquence globale.

II. DÉFINIR UN MODÈLE DE LANGUE

II.4. Problème des fréquence nulles (Zero)

- Dans un modèle de langage basé sur les fréquences (unigram, bigram, n-gram), il arrive qu'un mot ou un n-gramme **n'apparaisse pas du tout dans le modèle de langue**.

- Dans ce cas, le modèle lui attribue une **probabilité nulle** : $P(m_i \mid M) = 0$, Alors, la probabilité de toute la séquence devient **0.**

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

Solution : le Lissage (Smoothing)

✓ Assigner une probabilité non nulle aux événements (mots ou n-grammes) absents du corpus.

✓ 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 méthodes de lissages

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage

1) Méthodes de "Discounting"

Ajustent les fréquences observées pour attribuer une probabilité non nulle aux événements rares ou absents.

- Laplace correction (Add-1)
- Lidstone correction (Add-ε)
- Absolute discounting
- Leave-one-out discounting
- Good-Turing method

2) Techniques de lissage par Interpolation

- Estimation de Jelinek–Mercer
- Dirichlet

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Méthodes de “Discounting”

Ajouter une constante (1, 0,5 ou ε) à toutes les fréquences

Laplace correction (Add-1)

- Ajouter 1 à tous les événements (n-gram : s)

$$P_{\text{add-one}}(s \mid M) = \frac{\text{freq}(s) + 1}{\sum_{s_i \in V} (\text{freq}(s_i) + 1)}$$

- freq(s) = nombre d’occurrences de l’n-gramme s dans le corpus
- V = vocabulaire (ensemble de tous les n-grammes possibles)

Lidstone correction (Add-ε)

- Ajouter **ε** à tous les événements (n-gram : s), c’est une Généralisation du Laplace :

$$P_{\varepsilon}(s \mid M) = \frac{\text{freq}(s) + \varepsilon}{\sum_{s_i \in V} (\text{freq}(s_i) + \varepsilon)}$$

- freq(s) = fréquence observée de l’n-gramme s dans le corpus
- V = vocabulaire (ensemble de tous les n-grammes possibles)
- ε = petite constante $0 < \varepsilon < 1$

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Méthodes de “Discounting”

Good-Turing

Principe : Pour estimer la fréquence réelle d’un événement vu *s* fois, regardons combien d’événements apparaissent *s+1* fois.

Good-Turing remplace la fréquence observée $\mathrm{freq}(s)$ par une fréquence ajustée :

$$\mathrm{freq}^*(s) = (\mathrm{freq}(s) + 1)\frac{n_{s+1}}{n_s}$$

avec :

- n_s: nombre de n-grammes ayant une fréquence $\mathrm{freq}(s)$ (apparaissant n fois)
- n_{s+1} : nombre de n-grammes ayant une fréquence $\mathrm{freq}(s)$+1

(ex. n_0 n-grame jamais vu, n_1 n-grammes vus 1 fois …)

Problème : freq* peut être zéro s’il n y a pas de n-grames de fréquence $(\mathbf{freq}(s) + 1)$

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Méthodes de “Discounting”

Good-Turing

Exemple

Supposons :

- 100 mots vus 1 fois → $n_1 = 100$
- 20 mots vus 2 fois → $n_2 = 20$

Pour un mot vu **1 fois** :

$$\mathrm{freq}^*(1) = (1+1)\frac{20}{100} = 0.4$$

Good-Turing dit :

Même si tu vois le mot **1 fois**, sa fréquence estimée est **0,4**, pas 1.

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Exercice

Soit s=« text mining information » et soit le document suivant :

text 10
mining 5
association 3
database 3
algorithm 2
query 1
efficient 1

Calculer P(s | D) avec :

1- MLE en supposant ici un **modèle unigramme**

2- Laplace smoothing (add_one)

3- Good-Turing

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Solution

text 10
mining 5
association 3
database 3
algorithm 2
query 1
efficient 1

Nombre total de mots de document (somme des fréquences) :

$$N = 10 + 5 + 3 + 3 + 2 + 1 + 1 = 25$$

La séquence à évaluer :
$s =$« text mining information »

On cherche $P(s \mid D) = P(text \mid D)\ P(mining \mid D)\ P(information \mid D)$

1) Maximum Likelihood Estimation (MLE — unigram) : $P_{MLE}(m_i) = \dfrac{freq(m_i)}{\sum_{m \in V} freq(m)}$

- $P_{MLE}(text) = 10/25 = 0.4$
- $P_{MLE}(mining) = 5/25 = 0.2$
- $P_{MLE}(information) = 0/25 = 0$ Donc :

$$P_{MLE}(s \mid D) = 0.4 \times 0.2 \times 0 = \mathbf{0}$$

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Solution

text 10
mining 5
association 3
database 3
algorithm 2
query 1
efficient 1

2) Laplace smoothing :

$$P_{\text{add-one}}(s \mid M) = \frac{\text{freq}(s) + 1}{\sum_{s_i \in V} (\text{freq}(s_i) + 1)}$$

C'est le N

N=11+6+4+4+3+2+2=32

- $P_{add-1}(text) = (10 + 1)/32 = 11/32 \approx 0.34375$
- $P_{add-1}(mining) = (5 + 1)/32 = 6/32 = 0.1875$
- $P_{add-1}(information) = (0 + 1)/32 = 1/32 \approx 0.03125$

$$P_{add-1}(s \mid D) = \frac{11}{32} \times \frac{6}{32} \times \frac{1}{32} \approx \mathbf{0.002014}$$

Probabilité non nulle (≈ 0.0020). Laplace évite le zéro mais **sur-lisse** (donne trop de poids aux événements absents).

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Solution

text 10
mining 5
association 3
database 3
algorithm 2
query 1
efficient 1

3) Good-Turing

$$\text{freq}^*(s) = \frac{(\text{freq}(s) + 1)\, n_{s+1}}{n_s}$$

- freq(text) = 10 , $freq^*(10) = (10 + 1)\frac{n_{11}}{n_{10}}$, **n_{10} = 1** (seul *text* apparaît 10 fois)

n_{11} = 0 (aucun mot apparaît 11 fois) , $freq^*(10) = 0$

- Freq(mining) =5 , freq∗(5)= 0
- Freq(information): $freq^*(0) = 1 \cdot \frac{n_1}{n_0}$

n_0 = nombre de mots possibles mais jamais vus → **inconnu,** (on ne connaît pas tout le vocabulaire de la langue) , Donc impossible de calculer.

Solution Pratique : Good-Turing Approximation

1- Étape 1 — masse pour les mots jamais vus : $p_0 = \frac{n_1}{N}$

2- Étape 2 — masse restante pour les mots vus : 1− p_0

3- Étape 3 — calcul des probabilités pour les mots de la séquence

4- Étape 4 — probabilité de la séquence (unigram, indépendant)

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Solution

text 10
mining 5
association 3
database 3
algorithm 2
query 1
efficient 1

Solution Pratique : Good-Turing Approximation

1- Étape 1 — masse pour les mots jamais vus : $p_0 = \frac{n_1}{N} = \frac{2}{25} = 0.08$

2- Étape 2 — masse restante pour les mots vus : $1 - p_0 = 0.92$

3- Étape 3 — calcul des probabilités pour les mots de la séquence

$$P^*(text) = 0.92 \times \frac{10}{25} = 0.92 \times 0.4 = 0.368$$

$$P^*(mining) = 0.92 \times \frac{5}{25} = 0.92 \times 0.2 = 0.184$$

P∗(information)≈0.08

4- Étape 4 — probabilité de la séquence (unigram, indépendant)

P∗(s|D)=P∗(text)×P∗(mining)×P∗(information) , $P^*\left(\text{“}text\ mining\ information\text{”} \mid D\right) \approx 0.00542$

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Lissage par Interpolation

- **Problème avec les méthodes de discounting**

Les méthodes comme : Laplace / Lidstone, Good-Turing, **traitent tous les mots non observés de la même façon.** Ils leur donnent une petite probabilité uniforme, peu importe leur importance réelle.

- *Ce n’est pas réaliste : certains mots sont naturellement plus fréquents dans la collection que d’autres, même s’ils n’apparaissent pas dans ce document.*

Solution : Lissage par interpolation

- **Combiner plusieurs modèles de langue** au lieu de s’appuyer uniquement sur les données d’un document.

Autrement dit : Interpoler le modèle en utilisant d’autres sources d’évidence (par exemple la collection de documents)

- On prend un **mélange pondéré** des probabilités.

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Lissage par Interpolation

Jelinek-Mercer

On combine **deux modèles** :

1. Le modèle du document → ce que dit *le document* sur la probabilité du terme

2. Un modèle plus général (modèle de corpus) → la *collection entière*

Ainsi, si un terme n'apparaît pas dans le document → il peut quand même avoir une probabilité non nulle grâce au modèle de corpus.

$$\boldsymbol{P_{JM}(m_i \mid M_d) = \lambda\ P_{MLE}(m_i \mid M_d) + (1-\lambda)\ P_{MLE}(m_i \mid M_c)}$$

- M_d :modèle du **document**
- M_c :modèle de la **collection**
- $0 < \lambda < 1$: paramètre à ajuster

Pour une requête $Q = \{m_1, m_2, .., ..m_k\}$:

$$RSV(Q, d) = \prod_{m_i \in Q} [\lambda\, P_{MLE}(m_i \mid M_d) + (1-\lambda)\, P_{MLE}(m_i \mid M_c)]$$

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Lissage par Interpolation

Dirichlet

Problème avec Jelinek–Mercer

Jelinek-Mercer **ne prend pas en compte la taille du document** → les documents longs ont un poids égal aux documents courts si λ est le même.

•Les documents longs devraient **avoir plus de confiance dans leurs propres observations**.

Solution : Dirichlet Smoothing

Utilise la **taille du document** N et un **paramètre µ** pour ajuster la force du lissage.

$$P_{Dir}(m_i \mid M_d) = \frac{N}{N+\mu} P_{MLE}(m_i \mid M_d) + \frac{\mu}{N+\mu} P_{MLE}(m_i \mid M_c)$$

N :nombre total de mots dans le document M_d

μ :paramètre de régularisation (contrôle la force du lissage)

$P_{MLE}(m_i \mid M_d)$:probabilité du mot $m_i$dans le document

$P_{MLE}(m_i \mid M_c)$:probabilité du mot $m_i$dans la collection

II. DÉFINIR UN MODÈLE DE LANGUE

II.5. Techniques de lissage : Lissage par Interpolation

Dirichlet en RI

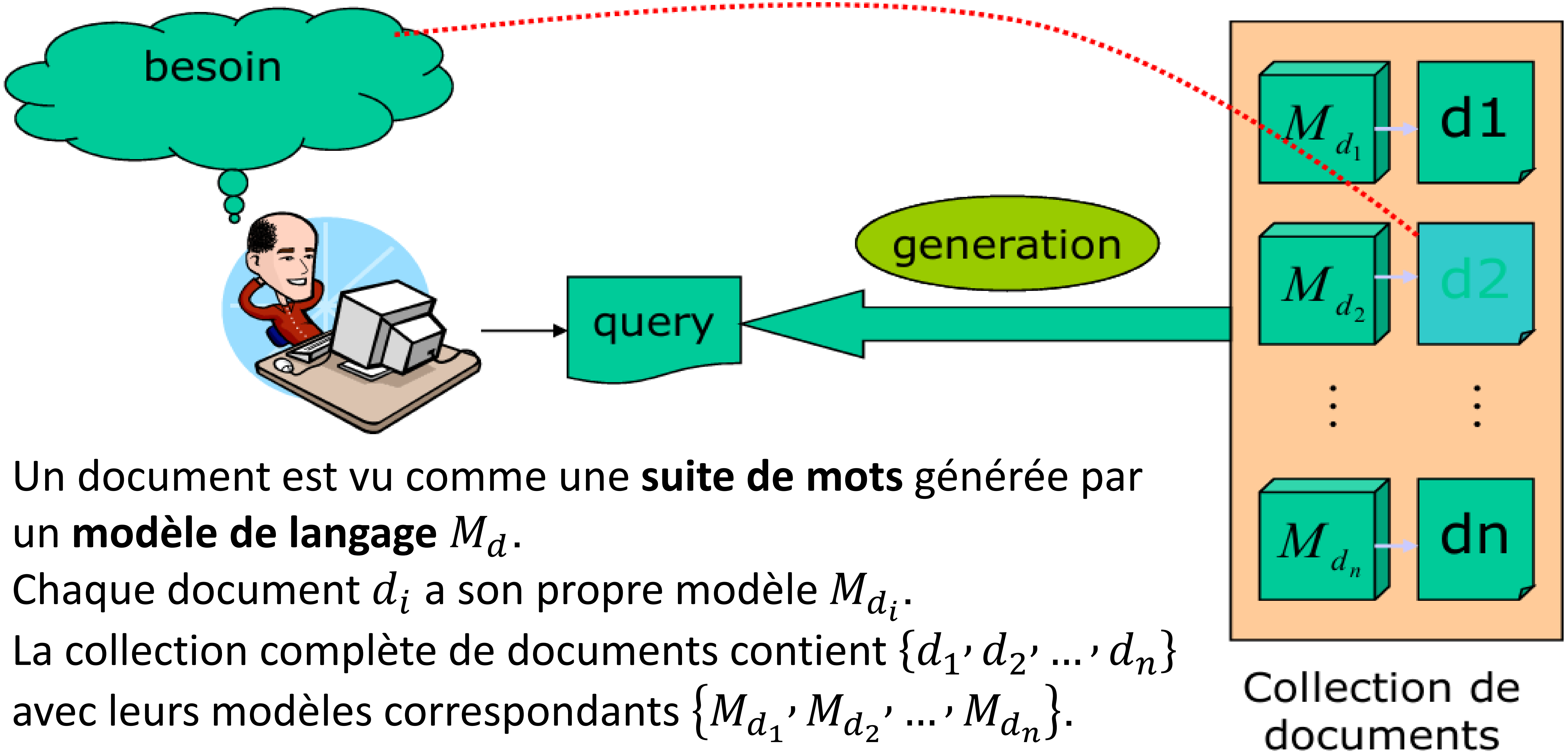
$$P_{Dir}(m_i \mid d) = \frac{tf(m_i, d) + \mu P_{MLE}(m_i \mid C)}{\mid d \mid + \mu}$$

- $tf(m_i, d)$:fréquence du mot $m_i$dans le document d
- $\mid d \mid$: nombre total de mots dans le document
- μ: paramètre de lissage, µ>0

$P_{MLE}(m_i \mid C)$:probabilité du mot dans la collection

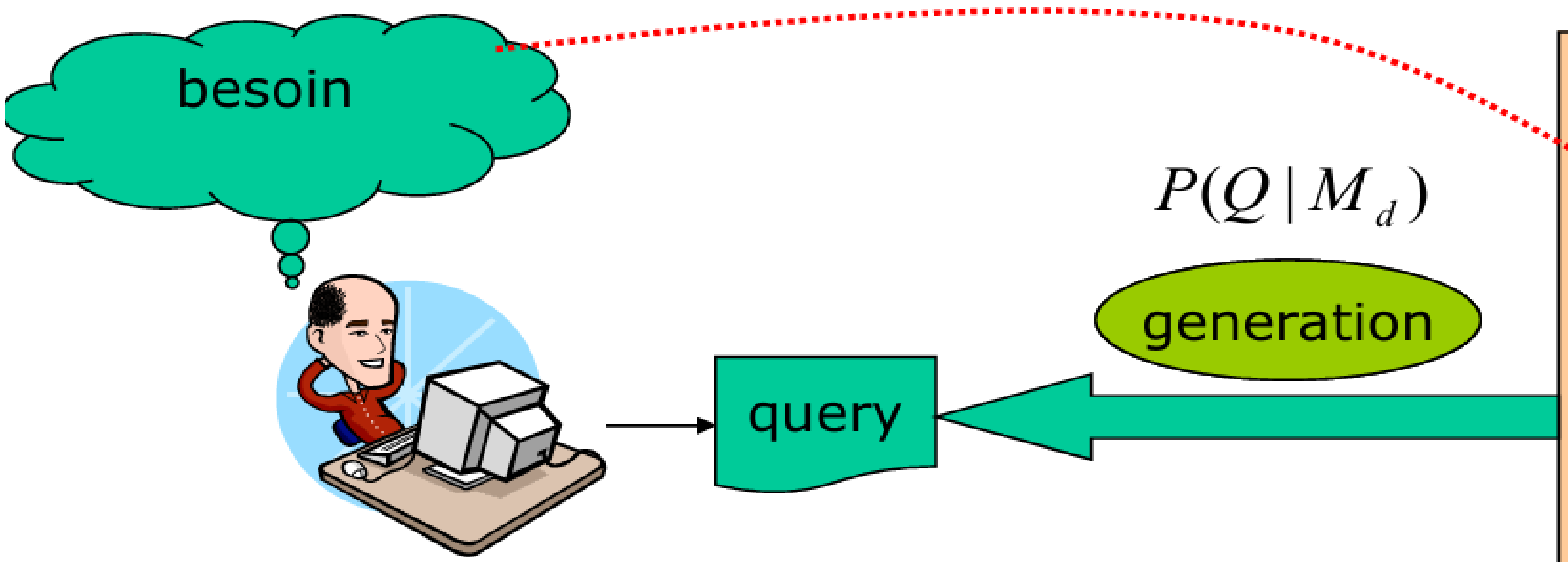
$$P_{MLE}(m_i \mid C) = \frac{freq(m_i \textit{ dans la collection})}{\textit{nombre total de mots dans la collection}}$$

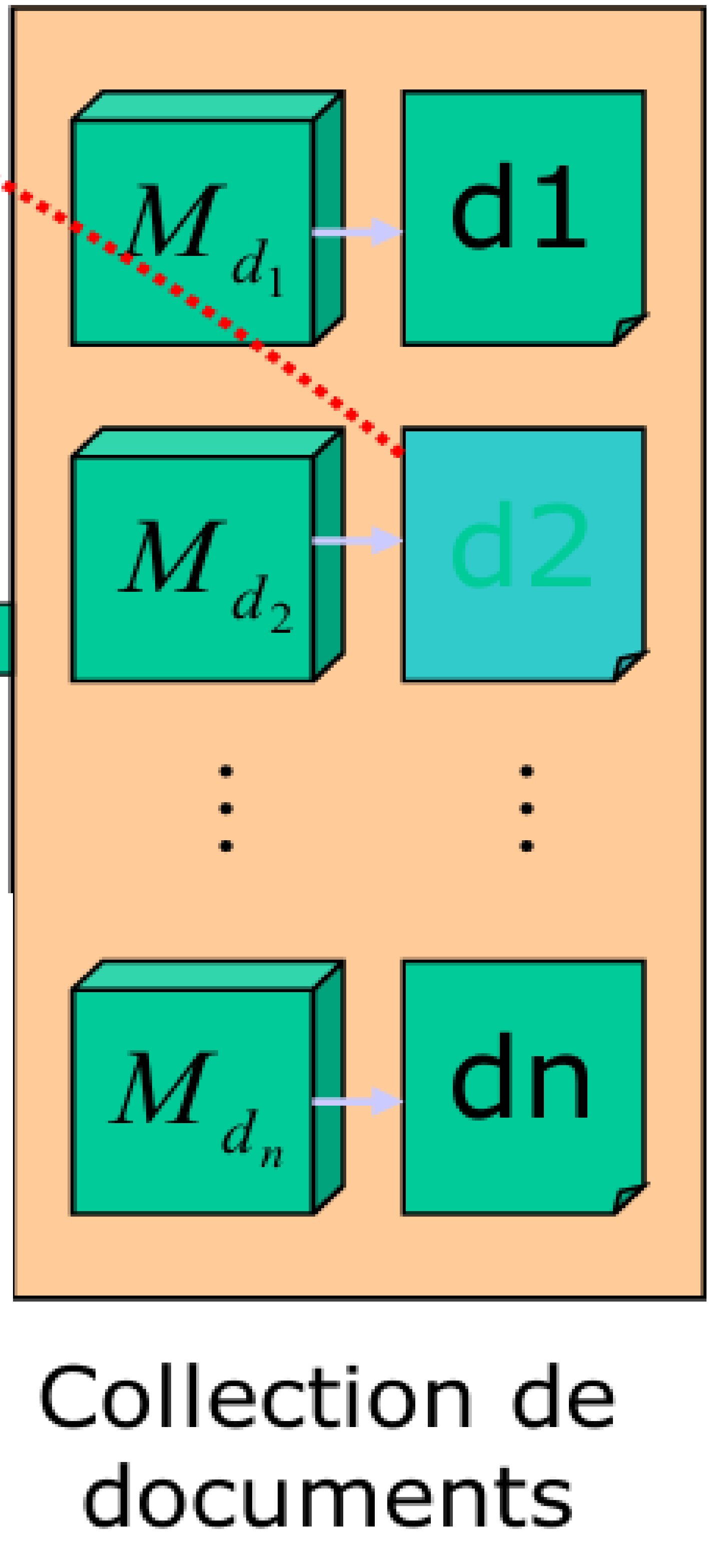
III. MODÈLE DE LANGUE EN Recherche D'Information



- Un document est vu comme une **suite de mots** générée par un **modèle de langage** M_d.
- Chaque document d_i a son propre modèle M_{d_i}.
- La collection complète de documents contient $\{d_1, d_2, \ldots, d_n\}$ avec leurs modèles correspondants $\{M_{d_1}, M_{d_2}, \ldots, M_{d_n}\}$.

III. MODÈLE DE LANGUE EN Recherche D'Information





Objectif : Pour une requête Q, estimer la probabilité qu'elle **ait été générée par le modèle du document** d:

$$P(Q \mid M_d)$$

Plus cette probabilité est élevée, plus le document d est pertinent pour la requête Q.

III. MODÈLE DE LANGUE EN Recherche D'Information

III.1. Comment estimer P(Q/Md)

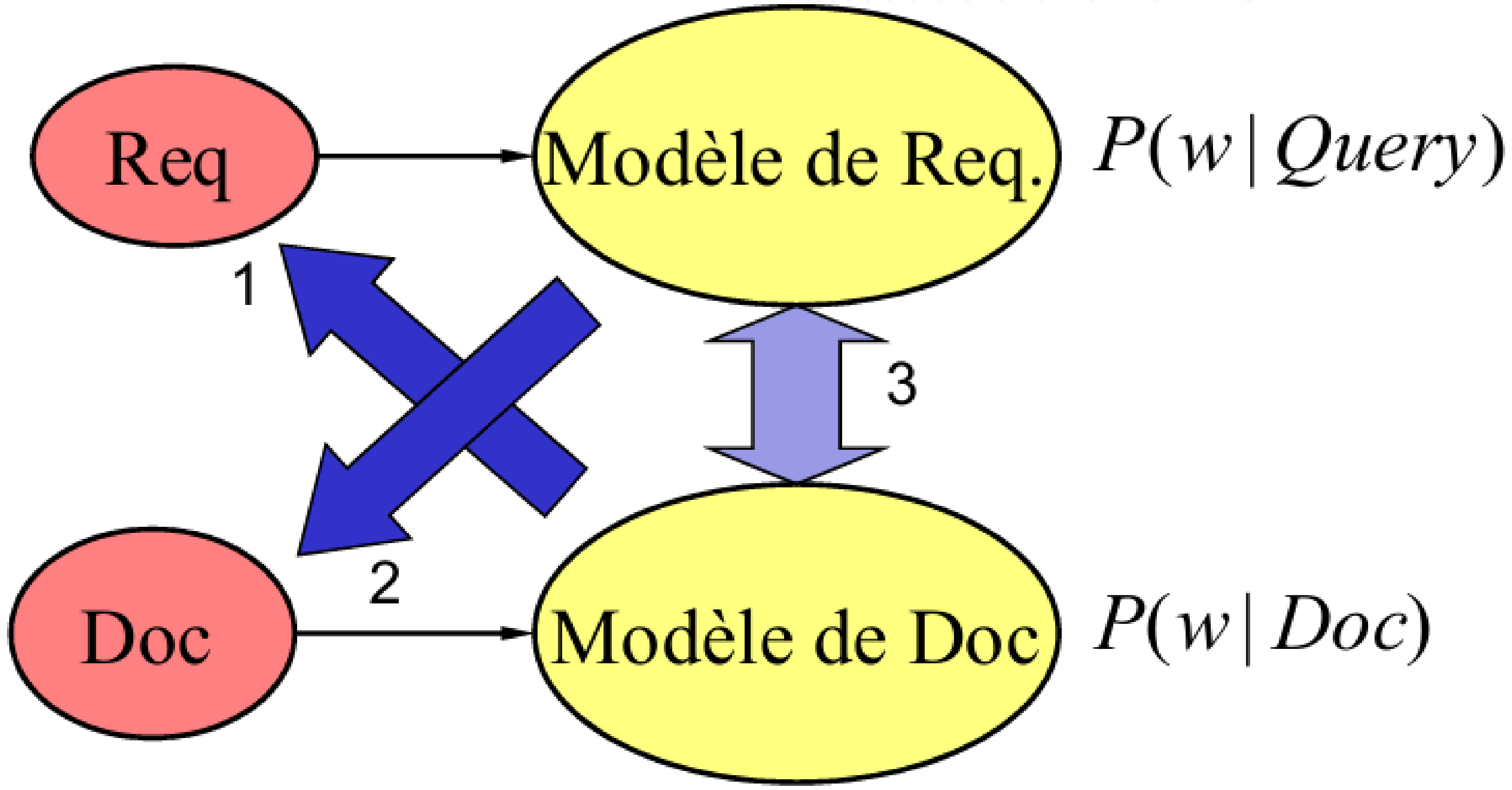
Estimation du modèle de langage du document

- Le modèle M_d est inconnu.
- **Solution** : On dispose d'un **échantillon** : le document lui-même.
- On peut donc **estimer les probabilités des mots** à partir du document, par exemple avec :
  - Maximum de vraisemblance (MLE)
  - Lissage (Laplace, Dirichlet, Jelinek-Mercer…)

III. MODÈLE DE LANGUE EN Recherche D'Information

III.2. ML en RI

On peut adapter les modèles de langage de 3 manières différentes à la RI.



3 Principes

1) P(w | Doc) : *Probabilité de générer la requête à partir du modèle du document.*

2) **P(w | Query)** : Probabilité de générer le document à partir du modèle de la requête.

3) Combinaison / comparaison des deux modèles

III. MODÈLE DE LANGUE EN Recherche D'Information

III.2. ML en RI

Principe 1 — Génération de la requête par le document

C'est le principe standard, le plus utilisé en RI.

- Le document **est représenté par son modèle de langage** $P(w \mid M_D)$
- La requête $Q = (q_1, q_2, \ldots, q_n)$ est une suite de mots.
- On calcule :

$$\text{RSV}(D, Q) = P(Q \mid M_D) = \prod_{i=1}^{n} P(q_i \mid M_D)$$

Plus le modèle du document a une forte probabilité de générer les mots de la requête, plus le document est pertinent.

III. MODÈLE DE LANGUE EN Recherche D'Information

III.2. ML en RI

Principe 2 — Génération du document par la requête

On inverse les rôles du document et de la requête.

- La requête est représentée par son modèle de langage $P\left(w \mid M_Q\right)$
- Le document $D = (d_1, d_2, ., .. d_m)$est une suite de mots.
- On calcule :

$$\text{RSV}(D, Q) = P\left(D \mid M_Q\right) = \prod_{i=1}^{m} P\left(d_i \mid M_Q\right)$$

On mesure si la requête est suffisamment “riche” pour générer le document.

Peu utilisé en pratique : les requêtes sont très courtes, donc leur modèle est peu fiable.

III. MODÈLE DE LANGUE EN Recherche D'Information

III.2. ML en RI

Principe 3 — Ratio de vraisemblance / comparaison de modèles

On compare directement le modèle du document au modèle de la requête.

- Document représenté par : $P(w \mid M_D)$
- Requête représentée par : $P\left(w \mid M_Q\right)$
- On calcule un score basé sur la comparaison des deux modèles :

$$\text{RSV}(Q, D) = f(P(w \mid M_D), P\left(w \mid M_Q\right))$$

Souvent via :

- ratio de vraisemblance,
- distance entre distributions.

On ne génère plus rien : on mesure à quel point **le modèle du document ressemble au modèle de la requête**.

III. MODÈLE DE LANGUE EN Recherche D'Information

III.3. Principe 1 — Génération de la requête par le document

Chaque document d est représenté par un modèle de langage M_d.

- On estime ce modèle à partir des fréquences des mots dans le document.
- On classe les documents selon leur capacité **à générer la requête** Q.

$$P(Q \mid M_d) = \prod_{t \in Q} P\,(t \mid M_d)$$

$$RSV(Q, d) = P(Q \mid M_d)$$

→Les termes de la requête sont supposés **indépendants**. Comme nous avons vu auparavant (Unigram)

III. MODÈLE DE LANGUE EN Recherche D'Information

III.3. Principe 1 — Génération de la requête par le document

La probabilité de générer la requête sachant un modèle de langage du document d , $P(Q \mid M_d)$ avec MLE est:

$$P(t \mid M_d) = \frac{tf(t,d)}{\mid d \mid}$$

- $tf(t,d)$:nombre d'occurrences du terme dans le document
- $\mid d \mid$: nombre total de mots (Nommé N auparavant)

Le score du document devient donc :

$$\text{RSV(Q,d)} = P(Q \mid M_d) = \prod_{t \in Q} P(t \mid M_d) = \prod_{t \in Q} \frac{tf(t,d)}{|d|}$$

III. MODÈLE DE LANGUE EN Recherche D'Information

III.3. Principe 1 — Génération de la requête par le document

Problème du MLE

MLE donne une probabilité **nulle** si un terme de la requête n'apparaît pas dans le document :

$$tf(t,d) = 0 \Rightarrow P(t \mid M_d) = 0$$

→Ce qui fait tomber toute la probabilité du document.

→ **Solution : Méthodes de lissage**.

Vu auparavant : Add-One (Laplace Smoothing) , Add-ε smoothing (Add-delta), Good-Turing Smoothing, Interpolated Smoothing : Lissage Jelinek-Mercer et Lissage Dirichlet

Aujourd'hui, la meilleure performance pratique provient généralement du : **Lissage de Dirichlet** et du **Jelinek-Mercer (JM)**

Exercice

On considère une collection composée de deux documents :

d_1 : *Xerox reports a profit but revenue is down*

d_2 : *Lucent narrows quarter loss but revenue decreases further*

La requête est : **Q = ''revenue down''**

On utilise un modèle de langage unigramme (approche standard), calculer la similarité entre la requête Q et les documents en utilisant :

1- MLE sans lissage,

2- MLE avec lissage Jelinek–Mercer , $\lambda = 0,5$

3- MLE avec lissage de Dirichlet, $\mu = 0,5$

1. MLE (sans lissage)

$$RSV(Q, d) = P(Q \mid d) = \prod_{w \in Q} P_{MLE}(w \mid d)$$

$$P_{MLE}(w \mid d) = \frac{tf(w, d)}{\mid d \mid}$$

2. **Lissage Jelinek–Mercer (JM) ,** λ = 0,5

$$RSV(Q, d) = \prod_{w \in Q} P_{JM}(w \mid d)$$

$$P_{JM}(w \mid d) = \lambda\, P_{MLE}(w \mid d) + (1 - \lambda)\, P_{MLE}(w \mid C)$$

3. Lissage de Dirichlet, μ = 0,5

$$RSV(Q, d) = \prod_{w \in Q} P_{Dir}(w \mid d)$$

$$P_{Dir}(w \mid d) = \frac{tf(w, d) + \mu\, P_{MLE}(w \mid C)}{\mid d \mid + \mu}$$

Solution

1- MLE sans lissage,

$$RSV(Q, d) = P(Q \mid d) = \prod_{w \in Q} P_{MLE}(w \mid d), P_{MLE}(w \mid d) = \frac{tf(w, d)}{\mid d \mid}$$

Calcul pour d_1:

•$tf(revenue, d_1) = 1 \rightarrow P_{MLE}(revenue \mid d_1) = 1/8$

•$tf(down, d_1) = 1 \rightarrow P_{MLE}(down \mid d_1) = 1/8$

$$\boldsymbol{RSV(Q, d_1) = \frac{1}{8} \times \frac{1}{8} = \frac{1}{64} \approx 0.0156}$$

Calcul pour d_2:

•$tf(revenue, d_2) = 1 \rightarrow P_{MLE}(revenue \mid d_2) = 1/8$

•$tf(down, d_2) = 0 \rightarrow P_{MLE}(down \mid d_2) = 0$

$$\boldsymbol{RSV(Q, d_2) = \frac{1}{8} \times 0 = 0}$$

Solution

2- Lissage Jelinek–Mercer (JM), $\lambda = 0.5$

$$RSV(Q, d) = \prod_{w \in Q} P_{JM}(w \mid d), P_{JM}(w \mid d) = \lambda P_{MLE}(w \mid d) + (1 - \lambda) P_{MLE}(w \mid C)$$

Probabilité dans la collection C**:**

Total mots dans les deux documents $\mid C \mid = 16$

$tf(revenue, C) = 2 \rightarrow P_{MLE}(revenue \mid C) = 2/16 = 0.125$

$tf(down, C) = 1 \rightarrow P_{MLE}(down \mid C) = 1/16 = 0.0625$

Pour d_1**:**

$$P_{JM}(revenue \mid d_1) = 0.5 \cdot \frac{1}{8} + 0.5 \cdot 0.125 = 0.0625 + 0.0625 = 0.125$$

$$P_{JM}(down \mid d_1) = 0.5 \cdot \frac{1}{8} + 0.5 \cdot 0.0625 = 0.0625 + 0.03125 = 0.09375$$

$$RSV(Q, d_1) = 0.125 \times 0.09375 \approx 0.0117$$

Solution

2- Lissage Jelinek–Mercer (JM), $\lambda = 0.5$

Pour d_2**:**

$$P_{JM}(revenue \mid d_2) = 0.5 \cdot \frac{1}{8} + 0.5 \cdot 0.125 = 0.125$$

$$P_{JM}(down \mid d_2) = 0.5 \cdot 0 + 0.5 \cdot 0.0625 = 0.03125$$

$$RSV(Q, d_2) = 0.125 \times 0.03125 \approx 0.00391$$

Solution

3- Lissage de Dirichlet, $\mu = 0.5$

$$RSV(Q, d) = \prod_{w \in Q} P_{Dir}(w \mid d) \;\; et \; P_{Dir}(w \mid d) = \frac{tf(w, d) + \mu \, P_{MLE}(w \mid C)}{\mid d \mid + \mu}$$

Pour d_1**:**

$$P_{Dir}(revenue \mid d_1) = \frac{1 + 0.5 \cdot 0.125}{8 + 0.5} = \frac{1 + 0.0625}{8.5} = \frac{1.0625}{8.5} \approx 0.125$$

$$P_{Dir}(down \mid d_1) = \frac{1 + 0.5 \cdot 0.0625}{8.5} = \frac{1.03125}{8.5} \approx 0.1213$$

$$RSV(Q, d_1) \approx 0.125 \times 0.1213 \approx 0.0152$$

Pour d_2**:**

$$P_{Dir}(revenue \mid d_2) = \frac{1 + 0.5 \cdot 0.125}{8.5} \approx 0.125$$

$$P_{Dir}(down \mid d_2) = \frac{0 + 0.5 \cdot 0.0625}{8.5} = \frac{0.03125}{8.5} \approx 0.00368$$

$$RSV(Q, d_2) \approx 0.125 \times 0.00368 \approx 0.00046$$