# Recherche d'information

## Modèles en Recherche d'Information
## Modèles de langue

Cours Master Recherche Paris 13
Recherche et extraction d'information

Antoine Rozenknop
Source : Romaric Besançon
CEA-LIST/LIC2M

# Modèles de langue pour la RI

- idée
  - ne pas tenter de modéliser directement la pertinence
  - estimer la probabilité P(Q|D) la probabilité d'avoir la requête sachant le document, i.e. estimer la probabilité que la requête soit générée à partir du document
  - repose sur l'idée que l'utilisateur, lorsqu'il formule sa requête, a une idée du document idéal qu'il souhaite retrouver et que sa requête est formulée pour retrouver ce document idéal
    - idée formulée dès les années 60 dans les premiers travaux sur la RI probabiliste (Maron)

# Modèles de langue

- idée des modèles de langue :
  - capter les régularités linguistiques d'une langue
- modèle probabiliste qui assigne une probabilité à toute séquence de mots
  - probabilité de générer une séquence de mots à partir du modèle
  - modèles largement utilisés en traitement du language
    - reconnaissance de la parole
    - désambiguïsation morpho-syntaxique
    - traduction automatique

# Modèles de langue

- séquence $s=m_1 \ldots m_n$

$$P(s)=P(m_1 \ldots . m_n)=\prod_{i=1}^{n} P(m_i|m_1 \ldots m_{i-1})$$

- approximation: dépendance limitée à *k* mots
  - un contexte de (k-1) mots précédent est suffisant pour estimer la probabilité d'un mot

$$P(s)=P(m_1 \ldots . m_n)=\prod_{i=1}^{n} P(m_i|m_{i-k+1} \ldots m_{i-1})$$

# Modèles de langue

- modèles k-grams : les plus utilisés sont pour *k*=1,2,3
  - unigrams

$$P(m_1 \ldots . m_n) = \prod_{i=1}^{n} P(m_i)$$

  - bigrams

$$P(m_1 \ldots . m_n) = \prod_{i=1}^{n} P(m_i | m_{i-1})$$

$$P(m_i | m_{i-1}) = \frac{P(m_{i-1} m_i)}{P(m_{i-1})}$$

  - trigrams

$$P(m_1 \ldots . m_n) = \prod_{i=1}^{n} P(m_i | m_{i-2} m_{i-1})$$

$$P(m_i | m_{i-2} m_{i-1}) = \frac{P(m_{i-2} m_{i-1} m_i)}{P(m_{i-2} m_{i-1})}$$

# Estimation des probabilités

- il faut estimer

  $P(m_i)$ $P(m_{i-1}\, m_i)$ $P(m_{i-2}\, m_{i-1}\, m_i)$

- estimation de la probabilité d'un n-gram $\alpha$ sur un corpus *C*
  - estimateur du maximum de vraisemblance (m*aximum likelihood*)

$$P_{ML}(\alpha|C) = \frac{f(\alpha)}{\sum_{\alpha_i \in C} f(\alpha_i)}$$

*f(* $\alpha$ *)* = fréquence d'occurrence de $\alpha$ dans le corpus

# Lissage

- problèmes de zéros
  - un n-gram qui n'apparaît pas dans le corpus a une probabilité nulle
  - → toute séquence qui le contient a une probabilité nulle
- techniques de *lissage* pour assouplir cette contrainte
- idée générale :
  - au lieu d'attribuer toute la masse de probabilité aux n-grams observés, on en garde une partie qu'on redistribue aux n-grams non observés

# Exemples de Lissage

- Lissage de Laplace (« *ajouter-un* »)
- Lissage de Good-Turing
- Lissage Backoff
- Lissage par interpolation

# Lissage de Laplace

- Lissage de *Laplace*
  - lissage « *ajouter-un* »

$$P(\alpha|C) = \frac{f(\alpha)+1}{\sum_{\alpha_i \in C} f(\alpha_i)+1}$$

  - si le corpus ne contient qu'une petite partie des n-grams possibles (ce qui est souvent le cas), la plus grosse part de la masse de probabilité sera distribuée sur les n-grams non observés.

# Lissage de Good-Turing

- Lissage *Good-Turing*

$$f'(\alpha) = (f(\alpha)+1) \times \frac{n_{f(\alpha)+1}}{n_{f(\alpha)}}$$

$n_x$ = nombre de n-grams apparus $x$ fois dans le corpus

$$P_{GT}(\alpha|C) = \frac{f'(\alpha)}{\sum_{\alpha_i \in C} f(\alpha_i)}$$

- diminution de $f'(\alpha)/f(\alpha)$ du poids de $\alpha$, redistribué sur les n-grams non vus
- l'estimation Good-Turing pour les n-grams de grande fréquence est instable
  - recommandé pour les n-grams de faible fréquence

# Lissage Backoff

- Lissage *Backoff*
  - utiliser des modèles d'ordre inférieur pour les n-grams non observés
  - par exemple, le modèle de Katz combine bigrams et unigrams

$$P_{Katz}(m_i|m_{i-1}) = \begin{cases} P_{GT}(m_i|m_{i-1}) \text{ si } m_{i-1}m_i \in C \\ \alpha(m_{i-1})P_{Katz}(m_{i-1}) \text{ sinon} \end{cases}$$

# Lissage Backoff

$$P_{Katz}(m_i|m_{i-1}) = \begin{cases} P_{GT}(m_i|m_{i-1}) \text{ si } m_{i-1}m_i \in C \\ \alpha(m_{i-1})P_{Katz}(m_{i-1}) \text{ sinon} \end{cases}$$

- la diminution de fréquence de l'estimation Good-Turing est redistribuée sur le modèle unigrams
- $\alpha(m_{i-1})$ est un paramètre qui détermine la part de cette redistribution à $m_i$

$$\alpha(m_{i-1}) = \frac{1 - \sum_{m_j \text{ t.q. } m_{i-1}m_j \in C} P_{GT}(m_j|m_{i-1})}{1 - \sum_{m_j} P_{ML}(m_j)}$$

# Lissage par interpolation

- lissage par *interpolation*
  - interpoler les modèles d'ordre inférieur, même si le n-gram est observé
  - Lissage de Jelinek -Mercer

$$P_{JM}(m_i|m_{i-1}) = \lambda_{m_{i-1}} P_{ML}(m_i|m_{i-1}) + (1-\lambda_{m_{i-1}}) P_{JM}(m_i)$$

  - $\lambda_{mi-1}$ est un paramètre qui est établi par apprentissage
    - déterminé par processus EM (*Expectation Maximisation*)

# Modèles de langue en RI

- application des modèles de langue pour la recherche d'information
  - estimer la probabilité d'avoir la requête sachant le document
  - estimer $P(Q|D)$ par $P(Q|M_D)$ la probabilité que la requête soit générée à partir du modèle de langue du document *D*
- Les modèles de langue pour la RI utilisent en général des modèles unigrams
- ➔ modèle de Ponte et Croft (98)

# Modèles de langue en RI

- D'autres approches sont aussi possibles
  - créer un modèle de langue pour la requête et calculer la probabilité qu'un document soit généré par ce modèle
    ➥ la pertinence d'un document pour une requête est estimée par $P(D|M_Q)$
  - modèles de Hiemstra (98/99) et Miller (98/99)
  - créer un modèle de langue $M_Q$ pour la requête, un modèle de langue $M_D$ pour le document, et utiliser une mesure de comparaison entre les modèles pour donner un score de pertinence à chaque document
  - modèle d'entropie croisée

# Modèle de Ponte et Croft

- Modèle de Ponte et Croft (98)
  - utilise les mots présents dans la requête et les mots absents de la requête

$$P(Q|M_D) = P(t_1 \ldots t_q) P(\neg t_{q+1} \ldots \neg t_n | M_D)$$

  - hypothèse d'indépendance sur les termes

$$P(Q|M_D) = \prod_{i=1}^{q} P(t_i|M_D) \times \prod_{i=q+1}^{n} 1 - P(t_i|M_D)$$

# Modèle de Ponte et Croft

- estimation de la probabilité $P(t_i|M_D)$ similaire à l'approche *backoff* pour combiner un modèle de langue du document et un modèle de langue du corpus

$$P_{PC}(t_i|M_D) = \begin{cases} P_{ML}(t_i|D)^{1-\hat{R}(t_i,D)} \times P_{avg}(t_i)^{\hat{R}(t_i,D)} & si\ tf(t_i,D) > 0 \\ P_{ML}(t_i|C)\ sinon \end{cases}$$

avec $P_{avg}(t_i)$ la probabilité moyenne de $t_i$ dans les documents qui le contiennent

et $\hat{R}(t_i,D)$ *est une fonction de risque estimée par*

$$\hat{R}(t_i,D) = \frac{1}{1+\text{avg}f(t)} \left( \frac{\text{avg}f(t)}{1+\text{avg}f(t)} \right)^{f(t,D)}$$

# Modèle de Hiemstra

- modèle de Hiemstra (98,99)
  - le score pertinence associé au document est P(D|Q)

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)} \propto P(D)P(t_1 \ldots t_n|D)$$

$$\hookrightarrow \quad s(D,Q) \propto P(D) \prod_{t_i \in Q} P(t_i|D)$$

  - estimation de la probabilité $P(t_i|D)$ de type interpolation

$$P(t_i|D) = \alpha P_{ML}(t_i|D) + (1-\alpha) P_{ML}(t_i|C)$$

# Modèle de Hiemstra

- en log

$$\log \prod_{t_i \in Q} P(t_i|D) = \sum_{t_i \in Q} \log\left(1 + \frac{\alpha P_{ML}(t_i|D)}{(1-\alpha) P_{ML}(t_i|C)}\right) + \underbrace{\sum_{t_i \in Q} \log((1-\alpha) P_{ML}(t_i|C))}_{\text{constante / D}}$$

- les probabilités $P_{ML}(t_i|D)$ *et* $P_{ML}(t_i|C)$ sont estimées par

$$P_{ML}(t_i|D) = \frac{f(t_i, D)}{|D|} \qquad P_{ML}(t_i|C) = \frac{df(t_i)}{\sum_{t_j} df(t_j)}$$

# Modèle de Hiemstra

- en réintégrant P(D) (estimé par |D|/|C|)

  ➥ on obtient une formule finale du score

$$s(D,Q)=\sum_{t_i \in Q} \log\left(1+\frac{\alpha f(t_i,D)\sum_{t_i} df(t_j)}{(1-\alpha)|D|df(t_i)}\right)+\log\frac{|D|}{|C|}$$

# Modèle du ratio de vraisemblance

- proposé par Kenney Ng (99,00)
- La RI comme ratio de vraisemblance
  - utiliser la vraisemblance comme critère de classement
  - si la vraisemblance d'un document augmente après que la requête est soumise, ce document a une plus forte probabilité d'être utile qu'un document dont la vraisemblance reste constante ou décroit
  - utiliser le ratio de vraisemblance défini par

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

  - → très similaire au modèle de Hiemstra