Un système de recherche d'information adapté aux données incertaines : adaptation du modèle de langue

Caroline Tambellini

Université Joseph Fourier – Grenoble I

U.F.R. Informatique et Mathématiques Appliquées

THESE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER – GRENOBLE I

Discipline : Informatique

Présentée et soutenue publiquement le 13 décembre 2007 par

Caroline TAMBELLINI

TITRE

Un système de recherche d'information adapté aux données incertaines : adaptation du modèle de langue

Composition du jury

Mme Dominique Rieu	*Présidente du jury*
M. Mohand Boughanem	*Rapporteur*
Mme Sylvie Calabretto	*Rapporteur*
Mme Catherine Berrut	*Directeur de thèse*
M. Christophe Brouard	*Co-directeur de thèse*

Thèse préparée au sein de l'équipe MRIM du laboratoire LIG

(Laboratoire Informatique de Grenoble)

Université Joseph Fourier – Grenoble I

Remerciements

C'est avec grand plaisir que je rédige cette page car elle signifie l'aboutissement de longues années de travail et me permet de remercier toutes les personnes sans qui ce manuscrit ne serait ce qu'il est.

Je souhaite tout d'abord remercier les membres du jury qui ont accepté d'évaluer mon travail. Merci à Mme Dominique Rieu d'avoir présidé mon jury de thèse, à Mme Sylvie Calabretto et M Mohand Boughanem d'avoir été rapporteurs sur mon manuscrit.

Je tiens bien évidemment à remercier mes directeurs de thèse, Mme Catherine Berrut et M Christophe Brouard, pour avoir encadré et dirigé mes recherches, ils ont été d'un soutien indéfectible durant toutes ces années faisant preuve de patience et de disponibilité. Leurs remarques constructives m'ont permises d'avancer et de faire de ce manuscrit ce qu'il est. Un grand merci à Catherine qui a toujours cru en moi, et ceci même lorsque je ne croyais plus en moi !

Je n'oublie pas de remercier tous les membres du laboratoire, et plus particulièrement les membres de l'équipe MRIM. Merci à Leila, qui même partie (trop) loin m'a toujours encouragée, Delphine qui m'a supportée dans le même bureau, Loic avec qui les pauses café ont toujours été un moment de détente et tous les autres Hung, Helga, Rami, Ali, Jean, Stéphane, etc.

Merci à M Laurent Besacier et Mme Brigitte Bigi de l'équipe GETALP pour leur aide et leur contribution.

Merci aussi au personnel administratif : à Valérie qui a toujours eu le mot juste pour m'encourager, à Martine sans qui ce manuscrit n'aurait été imprimé dans les temps et à Bernard qui a toujours été là pour me sortir d'ennuis techniques !

Je tiens aussi à remercier du fond du cœur toute ma famille et plus particulièrement mes parents qui m'ont toujours encouragée pendant toutes mes longues années d'études.

Pour finir, je tiens à remercier Roland pour tout ce qu'il a supporté avec beaucoup de self-control et sans qui, j'en suis certaine, ce manuscrit n'aurait jamais vu le jour !

Pour finir (vraiment !) je tiens à remercier mon fils Mathéo qui du haut de ses 11 mois a su distraire sa maman pour lui permettre de décompresser !

Mathéo, je te dédie cette thèse !

Il y a des longueurs d'onde que les gens ne peuvent pas voir, il y a des bruits que les gens ne peuvent pas entendre, et peut-être que les ordinateurs ont des pensées que les gens ne peuvent pas avoir.

R. Hamming

Résumé

Tout système de recherche d'information développe une méthodologie formelle ou opérationnelle pour affirmer si les termes de chaque document correspondent à ceux de la requête. La plupart de ces systèmes s'appuie sur l'hypothèse que les termes extraits des documents ont été parfaitement reconnus ou identifiés, et de fait leur fonction de correspondance repose sur une capacité à disposer d'une relation d'égalité entre terme du document et terme de la requête.

Notre travail se positionne dans le cas où les données ne s'avèrent pas parfaitement reconnues et donc qualifiées d'incertaines. Dans ce contexte, l'égalité entre termes du document et termes de la requête est remise en cause pour laisser place à la notion de 'presque égalité'. Nous proposons un système de recherche d'informations adapté aux données incertaines et basé sur le modèle de langue. Nous introduisons la notion d'appariement qui mesure la 'presque égalité' entre deux termes par le biais de la concordance et de l'intersection. L'appariement s'intègre à la fonction de correspondance. De plus, la valeur de certitude d'extraction des termes fournie par un système d'interprétation s'insère dans la fonction de pondération. Préalablement à la mise en place d'un tel modèle, nous vérifions l'applicabilité des hypothèses de base de la recherche d'information, à savoir la loi de Zipf et la conjecture de Luhn, à des données issues de l'oral, exemple de données incertaines.

Le modèle proposé est validé expérimentalement et comparé à des systèmes n'intégrant pas la notion d'incertitude. Enfin, nous présentons une application possible utilisant un système de recherche adapté aux données incertaines : un outil d'aide à la réunion téléphonique.

Mots-clés : recherche d'information, gestion de l'incertitude, modèles de langue.

Table des matières

Table des figures

Table des tableaux

Introduction

1. Problématique

Pour trouver l'ensemble des documents répondant à une requête, tout système de recherche d'information développe une méthodologie formelle ou opérationnelle pour affirmer si les termes de chaque document correspondent (plus ou moins partiellement) à ceux de la requête de l'utilisateur. La plupart des systèmes s'appuie sur l'hypothèse que les termes extraits des documents ont été parfaitement reconnus ou identifiés, et de fait leur fonction de correspondance repose sur une capacité à disposer d'une relation d'égalité entre termes du document et termes de la requête.

Dans certains cas les termes extraits sont incertains, c'est-à-dire que leur identification est incertaine. De ce fait, la relation d'égalité entre terme du document et terme de la requête n'est plus vraie. Se pose alors la question du comportement d'un système de recherche d'information dans de telles conditions.

1.1. Rappel sur la recherche d'information classique

La recherche d'information est « *l'opération qui permet à partir d'une expression des besoins en information d'un utilisateur de retrouver l'ensemble des documents contenant l'information recherchée*» [Salton, 1983].

Selon Alan Smeaton [Smeaton, 1989] « *le but d'un système de recherche d'information est de retrouver des documents en réponse à une requête des usagers, de manière à ce que les contenus des documents soient pertinents au besoin initial d'information de l'usager*[1] ».

Un système de recherche d'information est défini par un langage de représentation des documents (qui peut s'appliquer à différents corpus de documents) et des requêtes qui expriment un besoin de l'utilisateur (sous forme de mots-clés par exemple), et une fonction de mise en correspondance du besoin de l'utilisateur et du corpus de documents en vue de fournir comme résultats des documents pertinents pour l'utilisateur, c'est-à-dire répondant à son besoin d'information (cf. Figure 1).

[1] Dans le texte : The aim of an information retrieval system is to retrieve documents in response to a users request in such a way that the content of the documents will be relevant to the user's original information need.

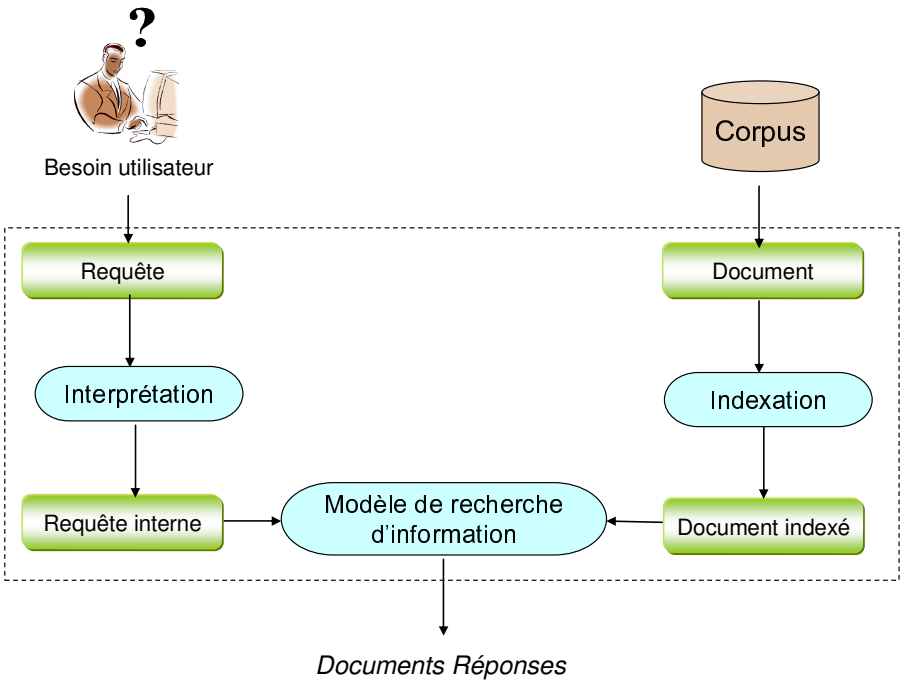


Figure 1. Système de recherche d'information (le système correspond à la zone dans le rectangle pointillé)

Afin de mettre en correspondance la requête et les documents, une interprétation de la requête et une indexation des documents du corpus s'avèrent nécessaires. L'indexation se décompose en trois phases (cf. Figure 2) :

- L'**extraction** des termes du document.
- La **sélection** des termes discriminatifs pour un document.
- La **pondération** des termes.

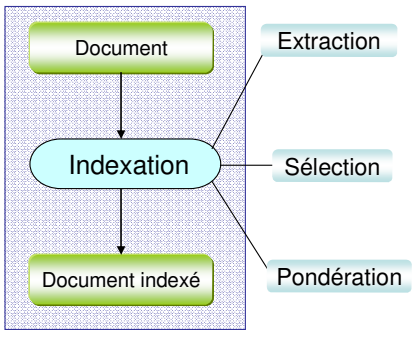


Figure 2. Indexation d'un document dans un processus de recherche d'information

Tout système de recherche d'information s'appuie sur un modèle (plus ou moins) de recherche d'information. Ce modèle de recherche d'information se base sur une fonction

de correspondance qui met en relation les termes d'un document avec ceux d'une requête en établissant une relation d'égalité entre ces termes. Cette relation d'égalité représente la base de la fonction de correspondance et, par la même, du système de recherche d'information. C'est pourquoi nous proposons une représentation alternative d'un système de recherche d'information centrée sur cette égalité (cf. Figure 3). Ainsi, cette vision d'un système de recherche d'information met en avant l'égalité entre termes du document et termes de la requête en montrant que la fonction de correspondance et la représentation du document se basent sur celle-ci.

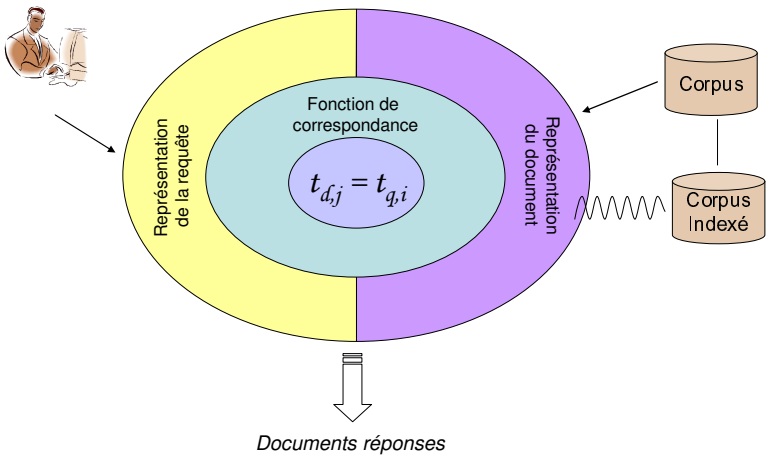


Figure 3. Système de recherche d'information

1.2. Les données incertaines

Lorsque le processus d'interprétation des données introduit de l'incertitude dans les données interprétées par rapport aux données initiales, on parle de données incertaines.

Imaginons deux personnes discutant, si l'une dit « tomate » et l'autre comprend « tarmac », il y a situation de quiproquo. En effet, l'auditeur fait une mauvaise interprétation de la donnée initiale exprimée par l'orateur (cf. Figure 4), la donnée interprétée par l'auditeur est incertaine.

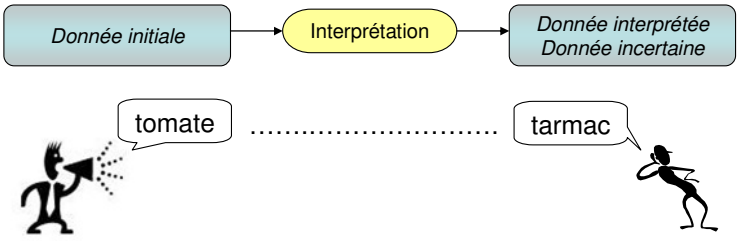


Figure 4. Situation de quiproquo

Certains processus d'interprétation des données commettent des erreurs, on parle de processus d'interprétation incertain. Trois types d'erreurs sont possibles : mauvaise reconnaissance, omission ou insertion de mots.

La notion de données incertaines désigne l'ensemble des données issues d'un processus d'interprétation incertain et associées à un coefficient de certitude. Les données interprétées deviennent alors des données incertaines.

L'exemple de la Figure 4 met en exergue que la donnée initiale diffère de la donnée incertaine dans certains cas. Quel est l'impact de cette différence d'interprétation sur un système de recherche d'information ?

1.3. 'Presque égalité' sur des données incertaines

En cas d'incertitude sur les données prises en compte par le système de recherche d'information, la base du système d'information n'est plus valide (cf. Figure 5).

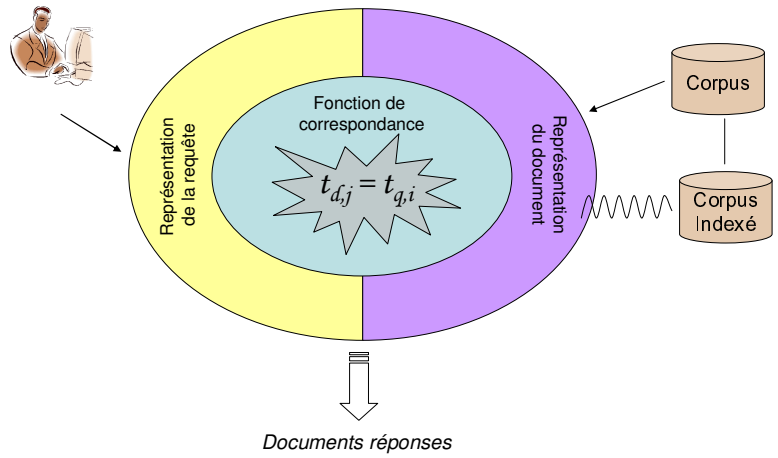


Figure 5. Egalité remise en cause dans un système de recherche d'information

La remise en cause de l'hypothèse de l'égalité entre termes du document et termes de la requête a un impact sur l'ensemble du système de recherche d'information. Ainsi les bases de la fonction de correspondance s'effondrent ainsi que celles des représentations de documents et de requêtes.

A la simple égalité $t_{d,j} = t_{q,i}$ se substitue la presque égalité $t_{d,j} \approx t_{q,i}$. La présence de cette presque égalité s'explique par l'existence d'erreurs dans les données incertaines, erreurs commises lors de l'interprétation des données initiales.

Exemple : système de recherche d'information 'classique' et données certaines

Soient trois documents D1, D2 et D3, contenant respectivement les termes « tomate », « orange » et « tomate » ; et une requête formée du terme « tomate ». Leurs documents

indexés contiennent quant à eux les termes « tomate », « orange » et « tomate » (cf. Figure 6).

Dans un contexte d'extraction de l'information sûre, le système représentera les documents initiaux sous forme de documents indexés contenant les termes des documents initiaux. Au niveau de la fonction de correspondance, les documents dont les représentations sous forme de documents indexés permettent l'égalité $t_{d,j} = t_{q,i}$, seront retournés par le système.

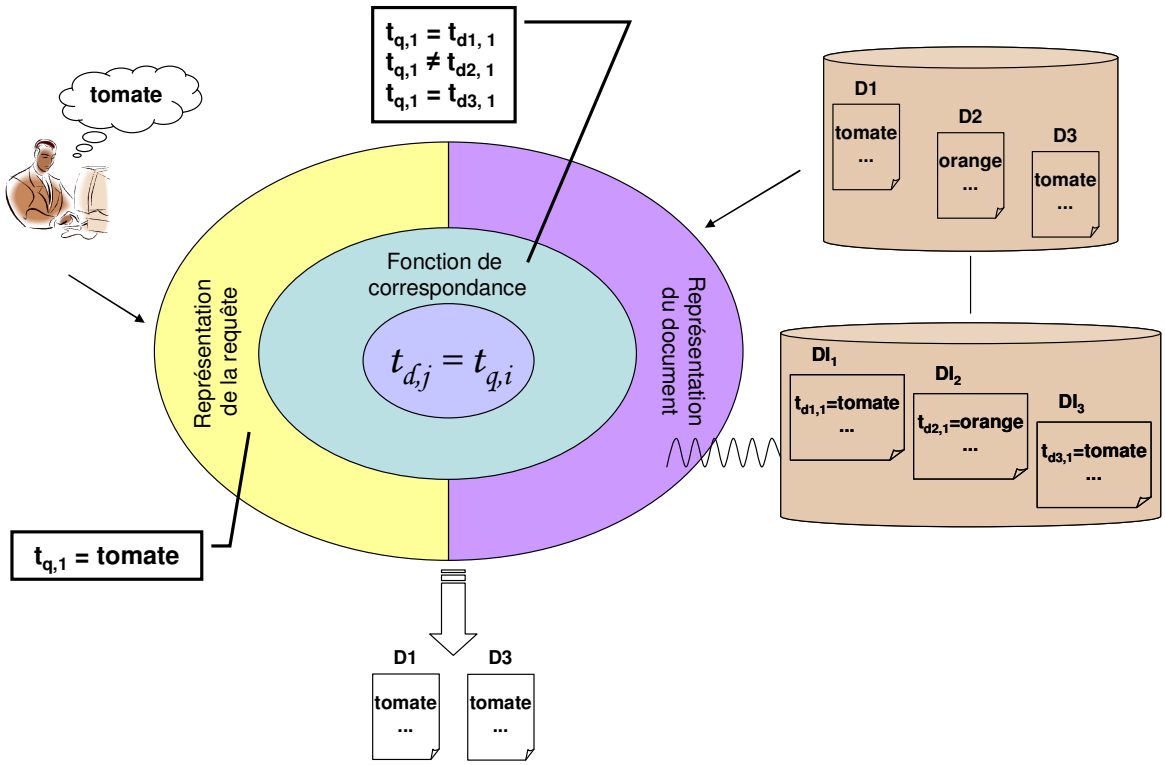


Figure 6. Fonctionnement d'un système de recherche d'information 'classique' avec des données certaines

Exemple : système de recherche d'information 'classique' et données incertaines

Soient trois documents D1, D2 et D3, contenant respectivement les termes « tomate », « orange » et « tomate » ; et une requête formée du terme « tomate ». Leurs documents indexés contiennent quant à eux les termes « tomate », « orange » et « tarmac » (cf. Figure 7).

Lors de la présence de données incertaines dans le document indexé, certains documents ne sont pas retournés par le système car l'égalité n'a pu être établie au sein de la fonction de correspondance. Dans notre exemple, le document D3 n'est pas considéré comme pertinent par le système de recherche d'information à cause du terme « tarmac ».

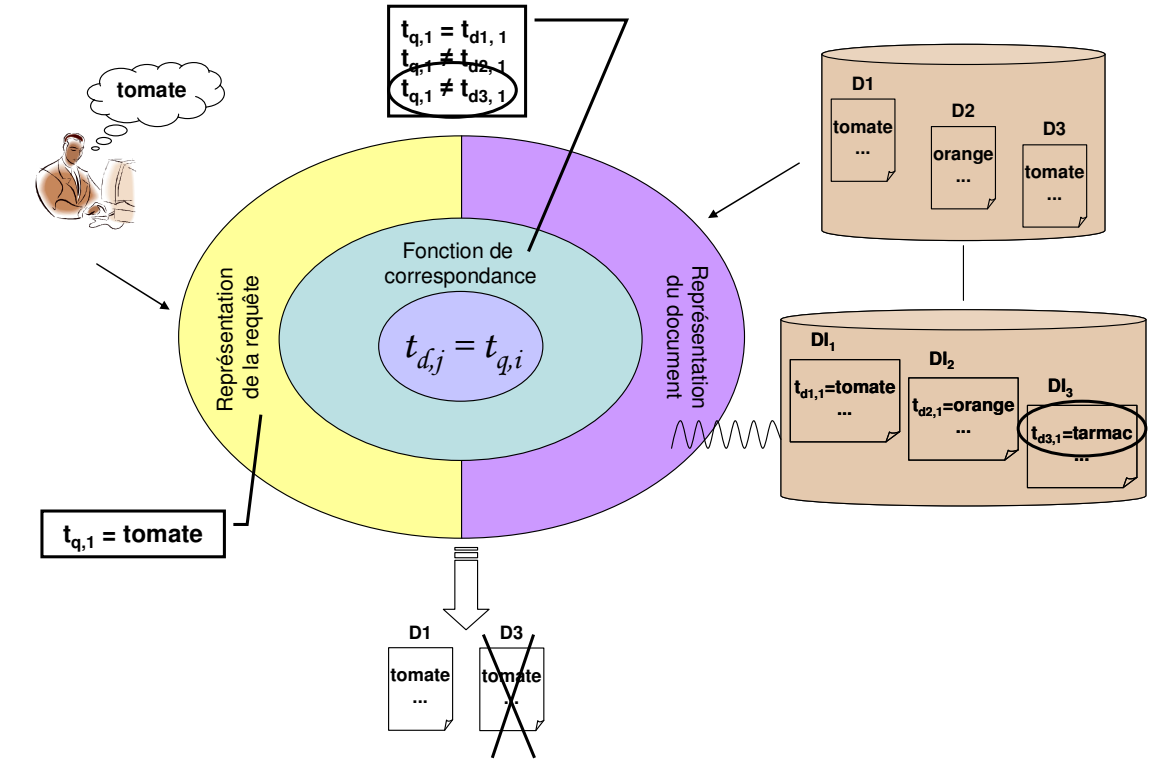


Figure 7. Fonctionnement d'un système de recherche d'information 'classique' avec des données incertaines

Exemple : système de recherche d'information prenant en compte la notion d'incertitude et données incertaines

Soient trois documents D1, D2 et D3, contenant respectivement les termes « tomate », « orange » et « tomate » ; et une requête formée du terme « tomate ». Leurs documents indexés contiennent eux aussi les termes « tomate », « orange » et « tarmac » (cf. Figure 8).

Nous proposons de ne pas considérer uniquement l'égalité ($t_{d,j} = t_{q,i}$) et a fortiori la différence ($t_{d,j} \neq t_{q,i}$) entre termes en intégrant la dimension de 'presque égalité' ($t_{d,j} \approx t_{q,i}$) entre termes.

Dans notre exemple, en considérant le terme « tarmac » comme 'presque égal' au terme « tomate », le système de recherche d'information permet de considérer le document D3 comme pertinent malgré le mot « tarmac » dans le document indexé DI_3.

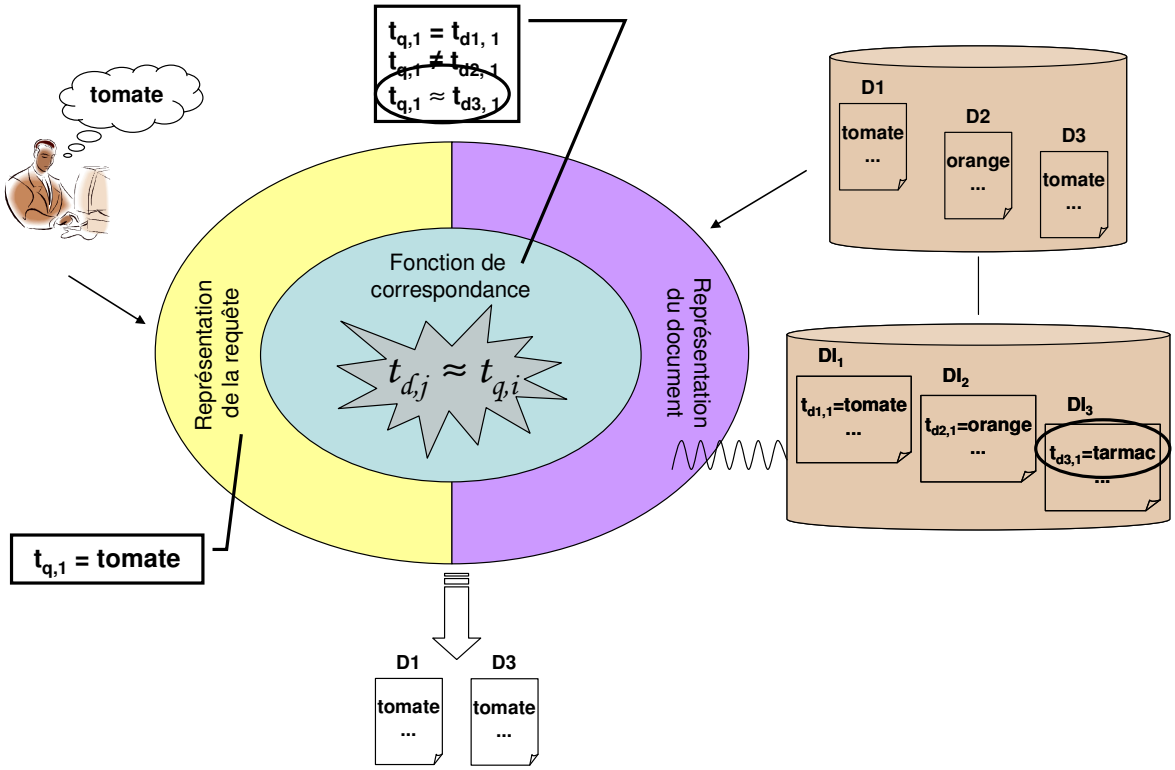


Figure 8. Fonctionnement d'un système de recherche d'information intégrant l'incertitude avec des données incertaines

2. Approche

Ainsi les documents indexés peuvent différer des documents initiaux par l'incertitude liée à l'extraction des termes.

Nous proposons un modèle de recherche d'information pour données incertaines. Pour cela, nous nous posons la question de la définition de la 'presque égalité' entre termes. Par ailleurs, les données incertaines ont un impact aussi bien au niveau de la pondération des termes dans la représentation du document qu'au niveau de la fonction de correspondance.

2.1. 'Presque égalité' entre termes

La fonction de correspondance proposée s'appuie sur la 'presque égalité' entre les termes : $t_{d,j} \approx t_{q,i}$. Cette 'presque égalité' est fortement liée à l'appariement entre un couple de termes, appariement fondé sur la concordance des termes ainsi que leur intersection.

La *concordance* définit le positionnement relatif entre deux termes. L'*intersection* mesure la proximité entre les zones communes de deux termes.

Soient deux termes t1=‘tomate’ et t2=‘tarmac’ (cf. Figure 9). Leur concordance correspond au type ‘égal’. Leur zone d’intersection est respectivement zt1=’tomate’ et zt2=’tarmac’. Leur intersection n’est pas parfaite puisque zt1 ≠ zt2.

Soient deux termes t1=’bonjour’ et t2=’journée’ (cf. Figure 10). Leur concordance correspond au type ‘chevauche’ puisque t2 chevauche le terme t1. Leur zone d’intersection est respectivement zt1=’jour’ et zt2=’jour’. Leur intersection est parfaite puisque zt1 = zt2.

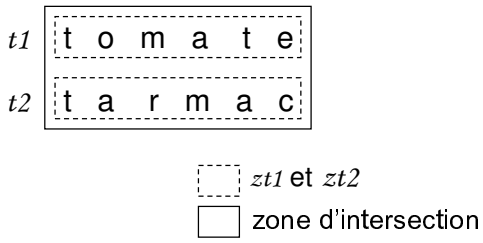


Figure 9. Exemple d'appariement entre les termes 'tomate' et 'tarmac'

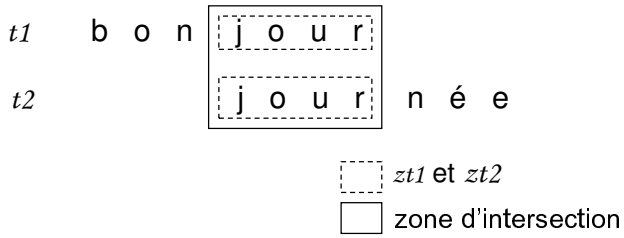


Figure 10. Exemple d'appariement entre 'bonjour' et 'journée'

Ces deux exemples montrent l’existence de diverses manières pour deux termes de concorder entre eux. Ils illustrent également le fait que l’intersection entre deux termes est plus ou moins parfaite.

2.2. Pondération des termes incertains

La notion d’incertitude apparaît également en ce qui concerne la représentativité des termes choisis pour l’indexation d’un document vis-à-vis du contenu de celui-ci. Usuellement, chaque terme d’un document est pondéré en tenant compte de la force locale du terme dans le document et de la force globale du terme dans le corpus. Habituellement, la force locale correspond à la fréquence relative du terme dans le document, le tf (term frequency) et l’inverse de la force globale au rapport du nombre de documents dans le corpus sur le nombre de documents contenant le terme, l’idf (inverse document frequency).

La notion de certitude d’extraction du terme s’ajoute à la fonction de pondération.

2.3. Fonction de correspondance pour données incertaines

Nous proposons un modèle de recherche d'information adapté aux données incertaines basé sur le modèle de langue. Dans le domaine de la recherche d'information, on distingue différents types de modèles dont les modèles booléen, vectoriel, probabiliste et les modèles de langue.

Le modèle booléen ne fournissant pas une liste ordonnée de documents pertinents demeure écarté. Dans l'approche du modèle probabiliste, seules la présence et l'absence des termes pour déterminer la probabilité de pertinence d'un document pour une requête interviennent. Il nous semble délicat d'intégrer la notion d'incertitude dans le modèle probabiliste qui s'avère déjà difficile à implémenter du fait de la nécessité d'un apprentissage préalable de la distribution de termes sur les documents pertinents et non-pertinents.

La dimension 'prise en compte des régularités de la langue' sous-jacente au modèle de langue nous conduit à le privilégier au détriment du modèle vectoriel. Nous souhaitons par la suite intégrer notre modèle adapté aux données incertaines dans un outil d'aide à la réunion. Les documents utilisés correspondent donc à des transcriptions de l'oral. De plus, le modèle de langue offre des perspectives d'évolution du modèle en passant des uni-grammes aux bi-grammes puis aux tri-grammes.

Nous proposons donc un modèle de correspondance adapté aux données incertaines, basé sur le modèle de langue intégrant la notion d'appariement entre termes.

3. Plan du manuscrit

Au vu des erreurs engendrées par certains systèmes d'extraction des données et des répercussions sur le processus d'indexation de documents d'un système de recherche d'information, il est nécessaire de mettre en place un modèle de recherche d'information capable de prendre en compte la dimension 'incertitude'. C'est ce que nous proposons dans ce manuscrit.

Notre manuscrit s'articule selon quatre parties.

La *première partie* traite de l'état de l'art. Nous détaillons les différentes études effectuées dans le domaine de la recherche d'information dans un contexte incertain, à savoir des documents provenant de l'oral ou du domaine de la reconnaissance optique de caractères. Cette étude nous permet de mettre en avant les répercussions des erreurs de transcriptions, aussi bien de la parole que caractères, sur les systèmes de recherche d'information. Dans cette même partie, un chapitre est consacré à la présentation des principaux modèles utilisés en recherche d'information.

La *deuxième partie* présente notre modèle de recherche d'information pour données incertaines. Après quelques définitions et notations préliminaires permettant de poser les bases du modèle, nous présentons notre modèle de langue intégrant la notion d'incertitude. Partant de la constatation que l'extraction d'un terme est plus ou moins certaine, nous intégrons ce concept à notre modèle : la ***certitude*** associée à chaque terme du document.

Pour déterminer la 'presque égalité' entre deux termes, nous introduisons la notion d'***appariement*** entre deux termes. L'appariement se définit par la ***concordance*** et l'***intersection*** existant entre ces deux termes. Nous définissons une typologie des différentes concordances possibles. L'intersection permet de mesurer la zone commune entre deux termes.

Enfin, nous détaillons une instanciation de l'appariement. Dans ce chapitre, nous décrivons nos choix d'implémentations du modèle. Nous montrons notamment l'utilisation d'algorithmes phonétiques pour déterminer l'intersection existant entre termes.

La *troisième partie* présente la validation de notre modèle en détaillant les différentes expérimentations et résultats. Nos expérimentations se divisent en deux parties : validation de la pondération et validation de la fonction de correspondance. Nous développons une étape préliminaire à la validation d'un modèle de recherche d'information adapté au contexte incertain en posant la question de l'applicabilité des grandes hypothèses de la recherche d'information, à savoir la conjecture de Luhn et la loi de Zipf, à des données issues de transcriptions de l'oral, données utilisées dans les expérimentations et dans lesquelles l'incertitude est omniprésente.

Afin de montrer l'utilisation possible d'un tel modèle de recherche d'information adapté au contexte incertain, nous présentons dans la *quatrième partie* une application potentielle : un outil d'aide à la réunion.

Pour finir, nous dressons un bilan de notre travail et nous abordons les perspectives.

Partie 1 : Etat de l'art

La vocation d'un système de recherche d'information est de mettre en correspondance une requête avec les documents d'un corpus en vue de déterminer le document de contenu le plus pertinent en réponse à la requête.

Dans un système de recherche d'information, on distingue trois niveaux (cf. Figure 11):

- Niveau utilisateur (1) : un utilisateur a une représentation mentale d'un besoin d'information dont il souhaite obtenir des documents pertinents, c'est-à-dire capables de répondre à ce besoin.
- Niveau interface du système (2) : à partir de la représentation du besoin d'information de l'utilisateur, c'est-à-dire la requête, le système fournit un certain nombre de documents qu'il juge pertinent vis-à-vis de ce besoin d'information.
- Niveau système (3) : la requête interne représente la requête de l'utilisateur dans un langage de requête. Les documents sont indexés sous une forme représentant leur contenu. Le système interprète la requête afin de pouvoir la mettre en correspondance avec les documents indexés.

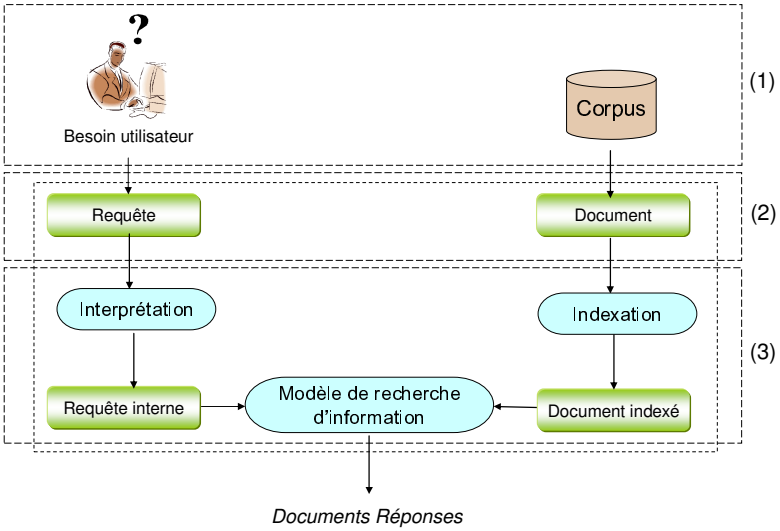


Figure 11. Les 3 niveaux d'un système de recherche d'information

Deux mesures sont généralement utilisées pour évaluer la qualité d'un système de recherche d'information, c'est-à-dire sa performance à renvoyer des documents pertinents en réponse à une requête utilisateur :

- le **rappel** qui mesure la capacité du système à sélectionner tous les documents pertinents.
- la **précision** qui mesure la capacité du système à sélectionner que des documents pertinents.

Pour calculer ces mesures, on confronte le point de vue de l'utilisateur avec celui du système. Pour une requête :

- l'utilisateur fournit les documents qu'ils jugent pertinents pour cette requête

- le système fournit les documents qu'il retrouve pour cette requête.

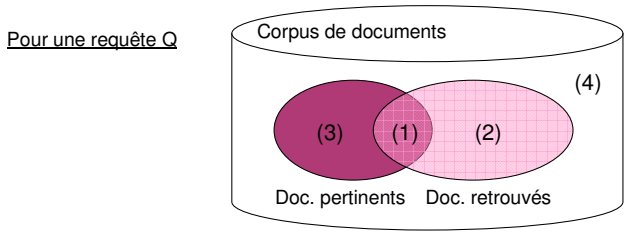


Figure 12. Documents pertinents ou non, retrouvés ou non

Ainsi (cf. Figure 12) on évalue le rappel comme le rapport (1) / [(1) + (3)] et la précision comme le rapport (1) / [(1) + (2)] :

- $rappel = \frac{nombre\ de\ documents\ pertinents\ et\ retrouvés}{nombre\ total\ de\ documents\ pertinents}$
- $précision = \frac{nombre\ de\ documents\ pertinents\ et\ retrouvés}{nombre\ total\ de\ documents\ retrouvés}$.

Trois mesures parmi le rappel, la précision, l'élimination et la généralité permettent une évaluation totale du système.

Pour comparer deux systèmes de recherche d'information, le même corpus de test est utilisé. Pour chaque système, une courbe rappel/précision est construite. Le système dont la courbe dépasse, c'est-à-dire dont la courbe se situe au-dessus de celle d'un autre est considéré comme un meilleur système.

D'autres mesures existent, telle la mesure *ESL* (Expected Search Length), introduite par [Cooper, 1968] qui permet d'évaluer le nombre de documents non pertinents devant être lus avant de lire *n* documents pertinents.

Cette partie, par le biais d'une description des études évaluant les performances des systèmes de recherche d'information sur des données incertaines, met en avant les difficultés des systèmes de recherche d'information pour gérer ce type de données. Les études sont effectuées sur des données provenant de transcriptions automatiques de la parole, c'est-à-dire issues des systèmes ASR[2], et de reconnaissances optiques de caractères, issues des systèmes OCR[3].

Nous étudions ensuite différents modèles de recherche d'information et plus particulièrement deux modèles de recherche d'information basés sur des probabilités : le modèle probabiliste et le modèle de langue.

[2] ASR : Automatic Speech Recognition

[3] OCR : Optical Character Recognition

Chapitre I. La recherche d'information dans un contexte incertain

Le problème de l'application de la recherche d'information à des données incertaines apparaît lorsque des documents « papier » sont scannés et qu'une reconnaissance optique des caractères (on parle d'OCR pour Optical Character Recognition[4]) sujette à des erreurs est réalisée. Le même problème survient lorsque des enregistrements audio sont automatiquement transcrits par des systèmes de reconnaissance de le parole (on parle d'ASR pour Automatique Speech Recognition[5]) eux aussi sujets à des erreurs.

Ce chapitre compare les performances de différents systèmes de recherche d'information sur des données incertaines et sur des données exactes[6] c'est-à-dire dépourvues d'incertitude. Dans un premier temps nous faisons un état de l'art dans le contexte de l'OCR, puis dans un second temps dans celui de l'ASR.

1. Problématique de la recherche d'information dans des documents provenant d'un système de type OCR

Nous présentons ici trois études majeures traitant de la problématique des systèmes de recherche d'information confrontés aux données incertaines générées par les systèmes d'OCR.

La première étude de Taghva et al. [Taghva, 1994a] pose la question de la correction à apporter aux documents provenant d'OCR pour avoir une recherche d'information identique à celle effectuée avec des documents exacts. En effet, malgré une exactitude de reconnaissance des caractères de 99%, un document OCR peut contenir jusqu'à 25 caractères erronés par page, pour une page contenant de 2500 à 3000 caractères.

La seconde étude de Croft et al. [Croft, 1994] porte sur le comportement du modèle booléen et du modèle probabiliste INQUERY en présence de données incertaines.

La troisième étude de Lopresti et al. [Lopresti, 1996] explore le problème de la recherche d'information avec des documents incertains en s'intéressant aux effets de différentes sortes de bruits OCR simulés, sur la performance de plusieurs modèles connus de recherche d'information. Cette étude a été faite suite à l'utilisation grandissante des systèmes OCR et des systèmes d'analyse de documents.

Ces études permettent de déterminer si certains modèles sont relativement robustes en présence de certaines erreurs et si pour d'autres l'impact s'avère plus important.

[4] Reconnaissance optique de caractères

[5] Reconnaissance automatique de la parole

[6] Par exemple issues de transcriptions manuelles

1.1. Etude de Taghva et al.

La première étude que nous décrivons est celle effectuée par Taghva et al. en 1994 [Taghva, 1994a].

1.1.1 Contexte d'évaluation

Un corpus de 204 documents est utilisé pour les expérimentations. Pour ces 204 documents, les documents OCR et les documents ASCII corrects sont à disposition.

Le système de recherche d'information choisi pour les expérimentations est le système BASISplus basé sur le modèle booléen de recherche d'information.

1.1.2 Evaluation & résultats

L'effet d'une variable indépendante simple, à savoir les données d'entrée, sur l'exécution d'un système de recherche d'information basé sur la logique booléenne est évalué dans ces expérimentations.

L'Expérimentation 1 inclut la transcription du texte par un scanner, l'interrogation des requêtes et la comparaison des résultats entre l'ensemble de documents corrects et l'ensemble de documents OCR.

Sur 71 requêtes exécutées, 63 donnent les mêmes résultats pour l'ensemble de documents corrects que pour l'ensemble de documents OCR. Pour ces 71 requêtes, 632 documents sont retrouvés par la base de données correctes et 617 par la base de données OCR, soient 15 documents oubliés dans l'ensemble de documents OCR.

L'Expérimentation 2 suit le même protocole que l'*expérimentation 1* avec une étape supplémentaire où un système de mise de fin de ligne et un système de post-traitement filtrent les documents avant de les enregistrer dans la base OCR.

Pour ces tests, les requêtes restent les mêmes. Les résultats donnent 632 documents retrouvés pour la base de documents corrects contre 624 pour la base de documents OCR. Cette expérimentation montre une amélioration des résultats. Toutefois, cette amélioration ne permet pas de considérer que l'étape supplémentaire ait un réel impact.

Cette étude permet seulement de conclure que la précision n'est pas affectée.

Taghva et al. [Taghva, 1994b] propose deux évaluations similaires de l'impact des documents provenant de systèmes OCR sur les performances de recherche du système de recherche d'information vectoriel SMART et du système de recherche d'information probabiliste INQUERY. Ils concluent que les effets sur les performances de recherche pour des documents longs sont minimes pour les systèmes OCR avec des taux de

reconnaissance élevés. Les erreurs courantes des OCR n'affectent pas significativement la précision et le rappel moyens sur des documents longs.

Cette conclusion non surprenante s'explique par le fait que leurs études sont effectuées avec des documents longs. De ce fait, les erreurs se « lissent » dans la masse d'information contenue dans un document long. L'erreur est compensée par la longueur du document.

1.2. Etude de Croft et al.

Croft et al. [Croft, 1994] utilisent une simulation de modèle OCR pour étudier l'effet des incertitudes sur les systèmes de recherche d'information. Les résultats montrent que l'incertitude a un impact sur les documents courts.

De telles évaluations nécessitent une base de documents tests. De telles bases de documents restent rares et très coûteuses à mettre en place. De ce fait, pour ces expérimentations, la création de documents OCR s'effectue par l'ajout d'erreurs OCR aux documents corrects permettant ainsi une simulation de résultats OCR. Ce système permet de disposer d'une base de documents contenant à la fois les documents corrects et les documents OCR.

Dans un premier temps, ils étudient l'impact des erreurs produites par les systèmes OCR sur un système booléen de recherche d'information. Les sorties produites sur des données exactes ou sur des données provenant d'OCR s'avèrent sensiblement identiques.

Dans un second temps, ils reproduisent ces tests avec un système de recherche d'information fournissant un classement de documents en sortie.

1.2.1 Simulation de OCR

Les données utilisées pour la simulation nécessitent une étude préalable des taux d'erreurs des caractères et des mots pour une gamme de dispositifs et de logiciels OCR. Deux systèmes OCR1 et OCR2, respectivement le plus mauvais et le meilleur système OCR, servent à l'analyse. L'étude s'effectue en utilisant 460 pages de documents provenant de la base de documents test d'un département américain de l'Energie. Les taux d'erreurs sont regroupés par type de page, type de mot, et par longueur de page. Les pages se divisent en groupes de qualité basés sur le nombre d'erreurs OCR contenus (cf. Tableau I). Les résultats obtenus par les deux systèmes OCR1 et OCR2 se trouvent dans les deux colonnes de droite. Une page standard utilisée pour la simulation contient 1778 caractères.

Groupe de qualité de page	Nombre de pages	Nombre de caractères	Exactitude OCR1 (%)	Exactitude OCR2 (%)
1	80	165 110	98.8	99.9
2	77	163 019	96.7	99.0
3	85	162 367	93.1	98.3
4	96	163 176	85.5	96.7
5	122	164 274	62.1	88.3
Total	460	817 946		

Tableau I. Groupes de qualité de pages définis pour simuler des taux d'erreurs OCR dans la performance de recherche de texte

La production des collections tests repose sur plusieurs hypothèses :

- Les statistiques reportées dans cette étude s'appliquent à tous les types de documents dans les collections utilisées.
- Les seuls facteurs de potentielles apparitions d'une erreur OCR dans un groupe particulier correspondent à la longueur et au type des mots (mots de l'anti-dictionnaire ou non).
- Toutes les erreurs OCR entraînent un mot corrompu non indexé par le système de recherche d'information. Or dans les systèmes OCR actuels, certains mots valides se transforment par erreur en d'autres mots valides.
- Toutes les erreurs OCR ont pour conséquence un mot corrompu jeté et non classé. Dans les systèmes OCR actuels, certains mots sont transformés par erreur en d'autres mots valides, par exemple le mot « tarmac » au lieu de « tomate ». Ce type d'erreur se simule difficilement.

Les cinq groupes de pages, représentant les différents niveaux de qualité de pages, se répartissent aléatoirement dans le texte en entrée durant le processus d'indexation. La taille constante des pages se détermine en divisant le nombre total de caractères dans l'ensemble de données par le nombre total de pages.

Un nombre déterminé aléatoirement entre 0 et 1, reflétant la probabilité d'erreur pour un mot en fonction de la longueur et du groupe de page, fournit la simulation des erreurs de mots OCR. Si le nombre se trouve dans la zone d'erreur, le mot n'est pas conservé, autrement le processus se déroule normalement.

1.2.2 Le corpus

L'étude pour la simulation OCR s'effectue sur quatre collections tests. Les collections sélectionnées représentent un ensemble de différentes sources, tailles de documents et de requêtes :

- CACM : des résumés d'informatique constituent cette collection de petite taille. Pendant de nombreuses années, elle sert de repère standard lors des expérimentations.
- NPL : des documents et des requêtes courts forment cette grande collection fréquemment utilisée dans diverses expérimentations de recherche d'information.
- WEST : des longs documents de textes complets, d'informations légales et plus spécifiquement de jurisprudence composent cette troisième collection.
- WSJ : parmi les quatre collections, ce sous-ensemble de la collection TIPSTER représente la plus grande collection contenant des documents de taille modérée et des articles intégraux du journal de Wall Street. Les requêtes de cette collection sont également les plus longues.

Le Tableau II décrit plus en détail les caractéristiques des collections.

Collection	Taille de la collection	Nombre moyen de mots par document
CACM	1 639 440	512
NPL	3 748 316	327
WEST	297 501 776	24 889
WSJ	279 249 494	2 828

Tableau II. Description des collections tests

1.2.3 Expérimentations

Les expérimentations s'effectuent en utilisant le système de recherche d'information probabiliste INQUERY développé à l'université du Massachusetts. Ce système possède un certain nombre de caractéristiques avancées et a réalisé d'excellents résultats aux évaluations TIPSTER et TREC [Harman, 1993].

De manière générale, on suppose que les erreurs OCR auront un plus gros impact sur les documents courts, puisque les documents longs ont plus d'information redondante. Cette intuition fait l'objet des tests des expérimentations.

Le Tableau III décrit les requêtes associées à ces collections. La caractéristique principale est la longueur importante des requêtes du journal de Wall Street. Les requêtes longues se révèlent une autre forme de redondance pouvant diminuer l'effet des erreurs OCR. De ce point de vue, la plus mauvaise combinaison de caractéristiques revient à la collection NPL formée de requêtes courtes et de documents courts. Nous soulignons, cependant, que le processus de génération d'erreurs s'applique uniquement aux textes des documents et pas aux requêtes.

Collection	Nombre de requêtes	Nombre de mots par requête			Nombre moyen de mots par requête
		Min.	Moy.	Max.	
CACM	50	2	14.24	49	13.0
NPL	93	3	7.26	12	7.1
WEST	34	5	11.05	20	9.6
WSJ	50	13	32.68	118	29.3

Tableau III. Statistiques sur l'ensemble des requêtes standard pour chacune des quatre collections utilisées pour évaluer les erreurs OCR sur la performance de recherche

1.2.4 Résumé des résultats

Le Tableau IV donne les résultats globaux des expérimentations en utilisant la précision moyenne pour tous les niveaux de rappel ainsi que l'écart entre les sorties OCR et les documents originaux.

Collection	Précision moyenne				
	STD	OCR1		OCR2	
CACM	34.9	32.5	(-6.9%)	34.3	(-1.7%)
NPL	25.8	23.2	(-10.1%)	23.5	(-9.1%)
WEST	48.2	46.2	(-4.0%)	48.0	(-0.4%)
WSJ	39.9	38.1	(-4.5%)	39.3	(-1.5%)

Tableau IV. Performance de recherche pour les quatre collections tests standards montrant les effets de deux niveaux de taux d'erreurs OCR simulées.

Les résultats confirment que les documents les plus affectés par les erreurs OCR correspondent aux collections formées de documents courts et de requêtes courtes. NPL correspond à la collection ayant la plus grande dégradation en précision moyenne. Elle demeure également la seule collection où le meilleur système OCR (OCR2) cause une perte significative en précision comparée à la collection originale. La collection CACM, composée de beaucoup de documents courts, connaît la plus grande dégradation après NPL. La collection WEST, possédant des documents très grands, connaît la plus basse dégradation pour les deux systèmes OCR.

Ces résultats permettent de conclure que, généralement, l'utilisation du meilleur système OCR comme entrée d'un système de recherche de texte n'affecte pas significativement la performance de recherche pour une base de données composée de longs documents.

1.2.5 Conclusion

La prise en compte des caractères généralement confondus pas les dispositifs OCR pourrait améliorer les simulations précédemment décrites.

Toutefois, cette étude permet une conclusion essentielle : les documents courts, et a fortiori les requêtes courtes, affectent les performances de recherche d'un système de recherche d'information.

1.3. Etude de Lopresti et al.

A partir des travaux cités ci-dessus et de leurs résultats, Lopresti et al. [Lopresti, 1996] proposent de nouvelles versions de certains modèles pour traiter des données imprécises en combinant un matching de chaîne approximative et la logique floue.

1.3.1 Mode d'évaluation

Dans le but d'évaluer leur modèle, ils utilisent, tout d'abord, un générateur d'erreurs pour simuler des incertitudes dans le document. L'utilisation d'un modèle d'erreurs synthétique permet le contrôle de la distribution des incertitudes. Trois cas d'incertitudes sont mis en place :

- Cas simple : l'incertitude correspond à l'insertion, la suppression ou la substitution d'un caractère simple au hasard dans la base de données.
- Modèle « simple-burst » : le cas simple plus de l'incertitude aléatoire selon un taux de dommage prédéfini.
- Modèle « confusion-matrix » : génération d'erreurs de caractères en proportion d'une distribution d'erreurs de type OCR spécifiée dans une matrice de confusion, matrice de confusion générée en exécutant leur algorithme de classification d'erreurs sur une importante quantité de textes produits par des systèmes OCR du marché.

1.3.2 Mesure de performance

Généralement, les études de l'impact des données incertaines sur le système de recherche d'information utilisent les mesures de rappel et précision. Cependant, mesurer le rappel et la précision nécessite un jugement de pertinence pour toutes les requêtes. Aussi, les auteurs préfèrent utiliser un coefficient de corrélation pour mesurer l'effet des incertitudes sur la performance du système.

Soient $x_1, x_2, \ldots, x_n$ et $y_1, y_2, \ldots, y_n$, deux séquences de nombre réels. La corrélation entre les deux séquences se définit par :

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{x})}{\left[\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2\right]^{1/2}} \text{ avec } \bar{x} = \sum_{i=1}^{n} x_i \text{ et } \bar{y} = \sum_{i=1}^{n} y_i$$

Dans cette étude, la corrélation est calculée entre l'ordonnancement des documents certains et l'ordonnancement des documents incertains.

1.3.3 Données test

1000 articles de journaux d'Internet dont les sujets s'étendent de l'agriculture à l'assurance en passant par le transport constituent les données tests. Une procédure automatique génère la requête : sélection au hasard d'un terme parmi des termes du document ayant une longueur prédéfinie et n'appartenant pas à l'anti-dictionnaire. La requête demeure si elle permet d'obtenir entre 4 et 8 documents avec le système de recherche d'information booléen.

Ainsi, pour chaque groupe de modèle de recherche, on dispose de 400 requêtes.

1.3.4 Expérimentations

Un niveau croissant d'erreurs s'applique à la base de documents afin de créer des données incertaines : on parle de dommage. De fait, le degré de dommage représente la proportion d'incertitudes dans chaque document.

La Figure 13 à la Figure 15 montrent en abscisse le coefficient moyen de corrélation sur l'ordonnancement des premiers 10% des documents pour les systèmes de recherche d'information Booléen, Booléens étendus, vectoriel et de proximité pour des degrés de dommage croissant (en ordonnée). Ainsi, 0,05 correspond à 5% d'erreurs dans les documents.

Les systèmes booléens et d'espace vectoriel montrent tous les deux, une diminution linéaire du coefficient de corrélation entre l'ordonnancement observé sur les données originales, c'est-à-dire certaines, et l'ordonnancement sur les données incertaines à des degrés d'erreurs augmentant.

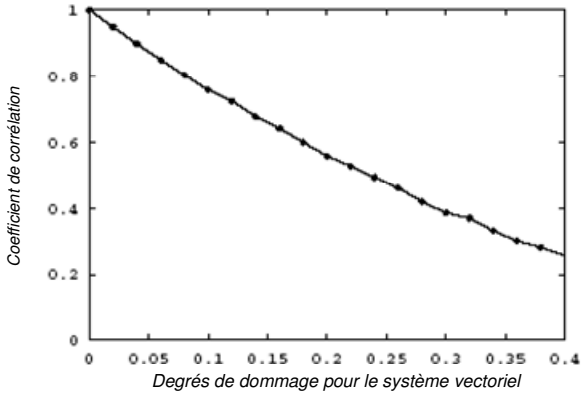


Figure 13. Coefficient de corrélation pour les systèmes de recherche booléens

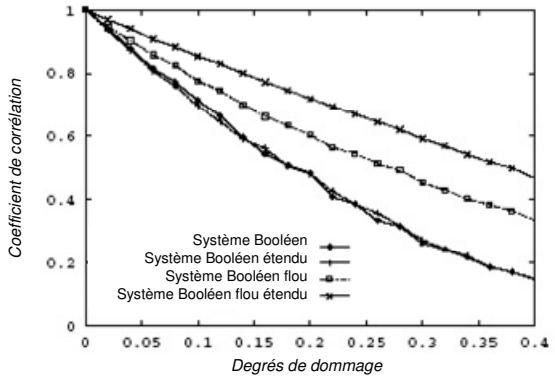


Figure 14. Coefficient de corrélation pour le système de recherche vectoriel

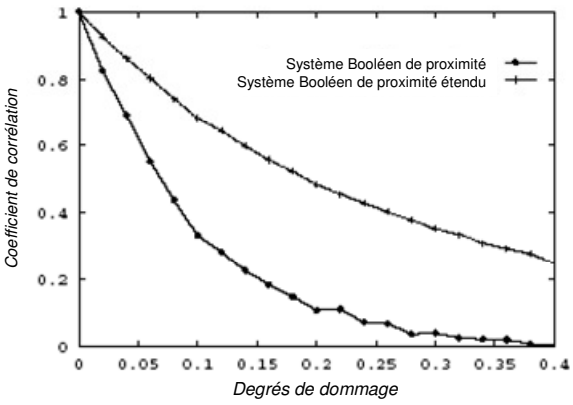


Figure 15. Coefficient de corrélation pour les systèmes de recherche de proximité et les systèmes booléens étendus

Le système booléen de proximité est plus affecté par les erreurs que les modèles booléens et d'espace vectoriel.

Les erreurs dans une base de documents peuvent augmenter le nombre de termes d'indexation de façon importante. En effet, on passe de moins de 20 000 termes d'indexation pour une base de documents sans incertitude à 160 000 pour une base de documents avec un degré de dommage de 0.4 (cf. Figure 16).

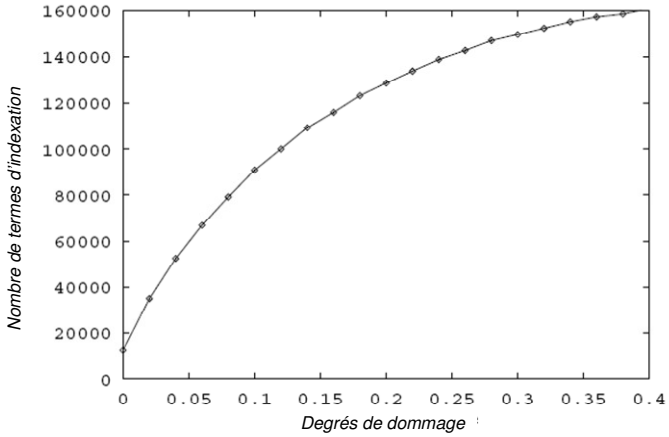


Figure 16. Croissance du nombre d'index dans le modèle d'espace vectoriel en fonction du degré de dommage

Quand une base de documents contient un fort degré de dommage, il est préférable d'avoir un système de recherche qui tolère les erreurs dans le texte.

1.3.5 Conclusions

Ces expérimentations mettent en place trois modèles d'incertitude. Un modèle simple correspond à l'insertion, la suppression ou la substitution de caractères. Un modèle dit « simple-burst » rassemble le cas simple avec du bruit aléatoire ajouté selon un taux de dommage prédéfini. Enfin, un modèle dit « confusion-matrix » représente un distribution d'erreurs de type OCR spécifiée dans une matrice de confusion.

Ces 3 sortes de modèles d'incertitude ont les mêmes conséquences de diminution des performances de recherche des systèmes de recherche d'information booléen, vectoriel, de proximité et booléen étendu.

1.4. Bilan

D'autres études telles que celles de Tsuda al. [Tsuda, 1995] aborde cette problématique. Cette étude pose la question de l'impact des données incertaines sur un système vectoriel de recherche d'information. Toutefois, la conclusion de toutes ces études reste la même, à partir d'un certain taux d'erreurs, tous les systèmes de recherche d'information sont affectés par l'incertitude. De ce fait, il s'avère nécessaire de prendre en compte l'incertitude au sein des systèmes de recherche d'information utilisant des bases de documents incertains.

2. Problématique de la recherche d'information dans les documents provenant d'un système de type ASR

Les documents provenant des transcriptions de systèmes de reconnaissance de la parole apparaissent comme un autre type de données incertaines. Tout comme les documents OCR, les documents provenant d'ASR contiennent des erreurs de type insertion, suppression ou substitution de mots. Partant de ces constatations, Grangier et al. [Grangier, 2003] posent la question de la détermination des conséquences d'une telle incertitude sur les systèmes de recherche d'information. Pour ce faire, ils effectuent des comparaisons entre texte correct et texte incertain sur trois des tâches principales d'un système de recherche d'information : la normalisation, la mesure de comparaison et l'extension de la requête. Nous décrivons cette étude majeure sur la problématique de la recherche d'information sur des données provenant de système ASR effectuée au sein de l'IDIAP[7].

On souligne l'existence d'une conférence appelée TDT[8] (Topic Detection and Tracking) qui propose de faire la recherche d'information à partir d'un signal audio ou plus exactement d'une transcription automatique de signal audio. Toutefois, bien que les systèmes travaillent sur des transcriptions automatiques, la problématique de la gestion de l'incertitude n'est pas abordée.

2.1. Corpus d'évaluation

Le corpus utilisé pour les évaluations correspond à la base de données TDT2 (issue de la conférence TDT) composée de 600h de news en anglais américain. La constitution de corpus s'effectue en enregistrant, chaque jour, un segment audio de longueur fixée pour chacune des sources suivantes : deux chaînes de télévision CNN, ABC, et deux stations de radio PRI et VOA. Les longueurs varient entre 30 et 60 minutes selon les sources. Pour chaque segment audio, deux transcriptions sont disponibles : une manuelle et une faite par

[7] Institut Dalle Molle d'Intelligence Artificielle Perceptive

[8] http://www.nist.gov/speech/tests/tdt/index.htm

un système de reconnaissance de la parole. Ensuite la segmentation manuelle de chaque enregistrement permet l'exécution de la tâche de recherche sans tenir compte des problèmes de segmentation automatique. La base de données fournit approximativement 21 500 documents pour la segmentation manuelle. La distribution de la longueur des documents montrent la distinction entre deux classes (cf. Figure 17) : les documents d'une longueur d'environ 50 mots considérés comme courts, et les documents d'une longueur d'environ 160 mots considérés comme longs.

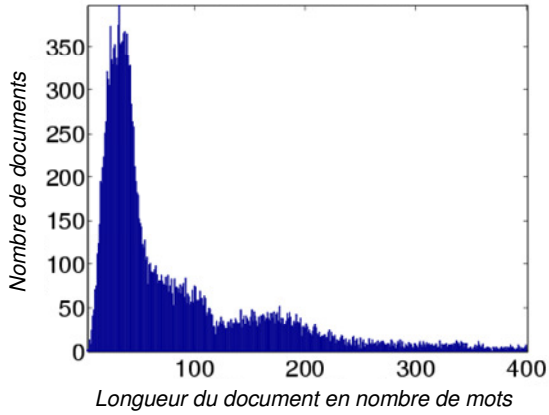


Figure 17. Distribution de la longueur des documents

2.2. La normalisation

La normalisation consiste à lemmatiser les termes du document et à supprimer les termes faisant partie d'un anti-dictionnaire.

Cette tâche s'applique sur les deux corpus : exact et incertain.

La normalisation s'effectue de la même façon pour les deux corpus avec une réduction de l'ordre de 50% après application de l'anti-dictionnaire (cf. Tableau V) et une réduction de l'ordre de 35% pour la lemmatisation (cf. Tableau VI). La seule différence observée réside dans le fait que le vocabulaire plus limité du document provenant du système ASR mène à un plus petit lexique.

Nombre de mots	Avant anti-dictionnaire	Après anti-dictionnaire	Réduction
Corpus exact	3 862 325	1 880 460	51%
Corpus incertain	3 841 053	1 859 893	50%

Tableau V. Réduction du corpus par application de l'anti-dictionnaire

Nombre de mots	Avant lemmatisation	Après lemmatisation	Réduction
Corpus exact	57 141	37 961	34%
Corpus incertain	37 696	23 099	38%

Tableau VI. Réduction du lexique par lemmatisation

2.3. La mesure de correspondance

La mesure de correspondance affecte des poids plus importants aux documents considérés comme pertinents et des poids moins importants aux documents non pertinents.

L'expérimentation compare les performances de la mesure de base *tf.idf* et de la mesure OKAPI sur les deux corpus exact et incertain.

Les résultats obtenus avec le système OKAPI sont inférieurs sur les données incertaines que sur les données exactes : 31.2% vs 35.6% (cf.Tableau VII).

	tf.idf	OKAPI
Précision moyenne[9] - Corpus exact	19.1%	35.6%
Précision moyenne - Corpus incertain	17.8%	31.2%
BEP[10] - Corpus exact	20.6%	35.4%
BEP - Corpus incertain	20.4%	33.1%

Tableau VII. Précision moyenne et BEP en utilisant la mesure de base tf.idf et la mesure OKAPI sur les corpus exact et incertain

2.4. L'extension de la requête

Le principe de l'extension de la requête considère la requête faite par l'utilisateur comme une « tentative » de requête et la change ensuite afin de rendre l'opération de recherche plus efficace. Cette extension de la requête est basée sur le principe de relevance feedback [Rocchio, 1971] (cf. Figure 18).

Les résultats des expérimentations montrent que, aussi bien pour les données exactes que pour les données incertaines, on obtient une amélioration significative des performances de recherche en utilisant les extensions de requête.

[9] Précision moyenne = avgP, voir Annexe 6

[10] BEP : Break-Even Point, voir Annexe 1

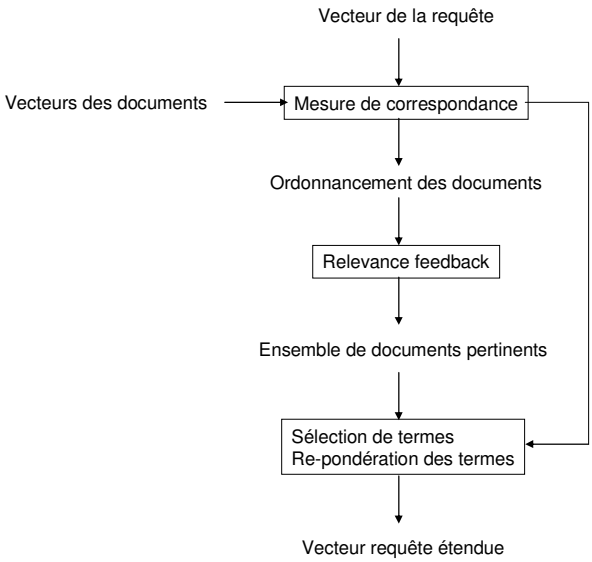


Figure 18. Stucture du processus d'extension de la requête

L'extension de la requête peut se faire de deux façons : à partir du corpus lui-même ou à partir d'un corpus parallèle. Les mots servant à l'extension de la requête proviennent soit du corpus lui-même, soit d'un autre corpus afin d'augmenter les résultats en faisant face notamment au problème rencontré lorsque l'ensemble des documents rapporté lors du premier passage est petit. En effet, dans un petit ensemble, les mots choisis vont apparaître essentiellement dans ces dits documents et de ce fait l'extension de la requête améliore faiblement les performances du système.

		Sans extension de la requête	Extension avec Rocchio	Extension avec LCA	Extension avec OW
Self QE Corpus exact	Précision moyenne	35.6%	42.1%	42.6%	41.1%
	BEP	35.4%	41.3%	40.5%	39.0%
Self QE Corpus incertain	Précision moyenne	31.2%	38.5%	40.2%	37.1%
	BEP	33.1%	38.6%	39.5%	37.7%
Parallel QE Corpus exact	Précision moyenne	35.6%	38.6%	40.1%	41.2%
	BEP	35.4%	38.8%	39.7%	40.9%
Parallel QE Corpus incertain	Précision moyenne	31.2%	34.2%	38.1%	36.8%
	BEP	33.1%	33.6%	37.6%	37.3%

Tableau VIII. Comparaison des effets de l'extension de la requête sur des données exactes et des données incertaines

2.5. Bilan

Les erreurs produites par un système automatique de reconnaissance de la parole affectent les performances des systèmes de recherche d'information aussi bien au niveau de la normalisation (à un plus faible degré), de la mesure de comparaison que de l'extension de la requête.

Cette étude, comme l'étude sur les effets des erreurs produites par les systèmes OCR, met en évidence qu'il est nécessaire de prendre en compte cette ***incertitude*** sur les données que l'on traite au niveau du système de recherche d'information.

3. Conclusion

Cet état de l'art des études a dressé un panorama de l'impact des données incertaines sur les principaux systèmes de recherche d'information.

L'étude de Taghva [Taghva, 1994a] montre que l'impact des données incertaines provenant de système OCR s'avère nul ou très faible sur les performances des systèmes booléens lors de l'utilisation de documents longs ; la longueur des documents induisant un lissage des erreurs.

Croft [Croft, 1994] effectue une étude similaire en utilisant un système probabiliste et il montre que les données incertaines affectent les performances de ce système lorsque les documents sont courts.

Lopresti [Lopresti, 1996] confirme que l'utilisation de documents courts diminuent les performances des systèmes de recherche d'information. Il montre que l'ajout de requêtes courtes affecte encore plus significativement les performances.

Grangier [Grangier, 2003] procède au même type d'études mais sur des données provenant de systèmes ASR. On retrouve les mêmes conclusions que pour les documents provenant de l'OCR.

De manière générale, dans le cas d'utilisation de documents longs, les données incertaines n'affectent pas ou très peu les systèmes de recherche d'information en comparaison des performances obtenues avec des données exactes. A contrario, si les documents et/ou les requêtes sont courts, les performances demeurent significativement affectées par des données incertaines.

Le chapitre suivant présente les principaux systèmes de recherche d'information. Cette étude a pour but de mettre en avant le système le plus adaptable aux données incertaines, afin de palier le problème de la diminution des performances des systèmes de recherche d'information avec des documents contenant des données incertaines.

Chapitre II. Modèles de recherche d'information

On distingue plusieurs familles de modèles de recherche d'information : les modèles basés sur la théorie des ensembles, les modèles basés sur des principes algébriques et les modèles basés sur les probabilités.

Les modèles booléens apparus dans les années 1950 se basent sur la théorie des ensembles. Ainsi, un tel modèle renvoyant un ensemble de documents jugés pertinents sans en proposer un ordonnancement est écarté.

Les modèles vectoriels reposent sur des principes algébriques.

Le premier système vectoriel de recherche d'information apparaît dans les années 1970 avec le système SMART [Salton, 1971].

Dans le modèle vectoreil, des vecteurs de poids représentent document et requête. Chaque poids dans le vecteur désigne l'importance du terme correspondant dans le document ou dans la requête. Pour qu'un vecteur prenne une signification, il faut préalablement définir un espace vectoriel. L'espace vectoriel se définit par l'ensemble de termes que le système a rencontré durant l'indexation, c'est-à-dire l'ensemble des termes de la collection de documents.

Le premier ***modèle probabiliste*** apparaît au début des années 1960 avec Maron et Kuhns [Maron, 1960]. Le principe consiste à présenter les résultats de recherche d'un système de recherche d'information dans un ordre basé sur la probabilité de pertinence d'un document vis-à-vis d'une requête.

Trois paramètres entrent dans le modèle probabiliste : la requête $\mathcal{Q}$, le document $\mathcal{D}$ et la pertinence $\mathcal{R}$. Le modèle classique de Robertson [Robertson, 1976] est fondé sur le ratio de vraisemblance entre $\mathcal{P}(\mathcal{R}=1 \mid \mathcal{D},\mathcal{Q})$ et $\mathcal{P}(\mathcal{R}=0 \mid \mathcal{D},\mathcal{Q})$. Ces deux probabilités signifient respectivement : *si on retrouve le document $\mathcal{D}$, quelle est la probabilité d'obtenir une information pertinente et si on retrouve le document $\mathcal{D}$, quelle est la probabilité d'obtenir une information non pertinente* [Nie, 2007].

Le principe s'appuie sur la détection de termes à la fois présents dans le document et la requête. Une pondération binaire des termes est utilisée, 0 ou 1, ce qui correspond à l'absence ou la présence d'un terme dans le document ou la requête. Pour une requête donnée, on cherche à déterminer $\mathcal{P}(\mathcal{R}=1 \mid \mathcal{D})$ et $\mathcal{P}(\mathcal{R}=0 \mid \mathcal{D})$. Le calcul de ces probabilités permet le classement des documents entre eux selon leur pertinence par rapport à la requête.

La plupart des modèles probabilistes se basent donc sur le modèle BIR ('Binary Independance Retrieval') introduit par Robertson [Robertson, 1976]. Ce modèle utilise la fonction *odd* qui évalue le rapport entre $\mathcal{P}(\mathcal{R}=1 \mid \mathcal{D})$ et $\mathcal{P}(\mathcal{R}=0 \mid \mathcal{D})$:

$$odd(\mathcal{D}) = \frac{\mathcal{P}(\mathcal{R}=1 \mid \mathcal{D})}{\mathcal{P}(\mathcal{R}=0 \mid \mathcal{D})}$$

Compte tenu de l'hypothèse d'indépendance des termes et grâce au théorème de Bayes, *odd* équivaut à :

$$odd(\mathcal{D}) \propto \frac{\mathcal{P}(\mathcal{D} \mid \mathcal{R}=1)}{\mathcal{P}(\mathcal{D} \mid \mathcal{R}=0)}$$

Cette fonction peut être utilisée à la place de *odd* exacte car elle garde le même rapport entre les scores affectés à chaque document et ainsi conserve l'ordonnancement des documents selon leur pertinence face à une requête donnée.

Le poids d'un terme t_i s'exprime donc selon la formule générale suivante :

$$t_i = log \frac{\frac{nombre\ de\ documents\ pertinents\ contenant\ t_i}{nombre\ de\ documents\ pertinents\ ne\ contenant\ pas\ t_i}}{\frac{nombre\ de\ documents\ non\ pertinents\ contenant\ t_i}{nombre\ de\ documents\ non\ pertinents\ ne\ contenant\ pas\ t_i}}$$

Dans la suite de cette étude, nous nous intéressons à un modèle probabiliste particulier : le modèle de langue.

1. Un modèle probabiliste basé sur la langue

Une autre façon de modéliser les documents sous forme probabiliste réside dans l'utilisation des modèles de langue. Les modèles de langue fournissent une représentation d'un langage. Cette représentation peut être utilisée au sein de modèles de recherche d'information. La partie qui suit présente le principe général des modèles de langue puis leur application au contexte de recherche d'information.

Le principe des modèles de langue consiste à « tenter de capter de manière statistique les régularités d'une langue en observant des phrases dans un corpus d'entraînement » [Boughanem, 2004].

Pendant plus de 25 ans, les modèles de langue jouent un rôle central dans le domaine de la reconnaissance de la parole [Jelinek, 1997]. Basé sur un modèle décrivant une langue, par exemple l'anglais, un système de reconnaissance de la parole s'avère capable de choisir parmi des hypothèses en concurrence celle correspondant à la transcription de ce qui a été dit. Ce modèle de langue attribue une probabilité à chaque expression apparaissant dans le discours anglais. Le système de reconnaissance de la parole compare alors les probabilités affectées à chaque hypothèse en conjonction avec d'autres facteurs (dictionnaire de la langue, etc.) et choisit l'hypothèse de transcription correspondant le mieux au signal sonore en train d'être analysé.

Les modèles de langue jouent également un rôle central dans la traduction automatique statistique [Brown, 1993]. Pour effectuer une traduction du français vers l'anglais, un système de traduction statistique traite le texte en entrée comme le résultat d'une transmission originalement en anglais. La tâche de traduction se conceptualise comme la détermination du texte original en anglais résultant du texte en français reçu en canal d'entrée. Comme pour la reconnaissance de la parole, un entraînement préliminaire du modèle de langue s'avère nécessaire.

Plus récemment, le domaine de la recherche d'information a adapté les modèles de langue à ses besoins. La taille des corpus utilisés en recherche d'information permet la mise en place de modèles de langue.

2. Principe général des modèles de langue

Un modèle de langue se construit en utilisant « une fonction de probabilité $\mathcal{P}$ qui assigne une probabilité $\mathcal{P}(s)$ à un mot ou une séquence de mots s dans une langue » [Boughanem, 1994] (cf. Figure 19). On appelle langue un corpus de documents. Cette fonction permet d'estimer la probabilité d'une séquence quelconque de mots dans la langue modélisée ou de façon plus générale, d'estimer *la probabilité de générer cette séquence de mots à partir du modèle de langue*.

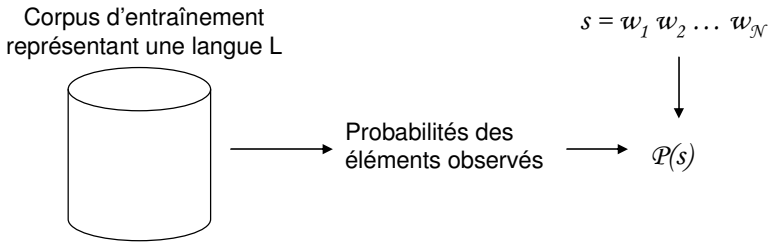


Figure 19. Principe des modèles de langue

Pour estimer la probabilité de la phrase $s = w_1\ w_2 \ldots\ w_n$ d'appartenir à la langue $\mathcal{L}$, on calcule la probabilité de la phrase d'être dans le modèle $\mathcal{ML}$ de la langue $\mathcal{L}$:

$$\mathcal{P}(s) = \prod_{i=1}^{n} \mathcal{P}_{\mathcal{ML}}\left(w_i \middle| w_1 \ldots w_{i-1}\right)$$

Par simplification des calculs, on fait l'hypothèse que seuls les n-1 mots précédents sont importants. Ainsi, on parle de modèle uni-grammes, bi-grammes ou tri-grammes :

- uni-gramme $\mathcal{P}(s) = \prod_{i=1}^{n} \mathcal{P}_{\mathcal{ML}}\left(w_i\right)$
- bi-gramme $\mathcal{P}(s) = \prod_{i=1}^{n} \mathcal{P}_{\mathcal{ML}}\left(w_i \middle| w_{i-1}\right) = \prod_{i=1}^{n} \frac{\mathcal{P}_{\mathcal{ML}}\left(w_{i-1} w_i\right)}{\mathcal{P}_{\mathcal{ML}}\left(w_{i-1}\right)}$
- tri-gramme $\mathcal{P}(s) = \prod_{i=1}^{n} \mathcal{P}_{\mathcal{ML}}\left(w_i \middle| w_{i-2} w_{i-1}\right) = \prod_{i=1}^{n} \frac{\mathcal{P}_{\mathcal{ML}}\left(w_{i-2} w_{i-1} w_i\right)}{\mathcal{P}_{\mathcal{ML}}\left(w_{i-2} w_{i-1}\right)}$.

Pour estimer les probabilités, on utilise un corpus de documents représentatifs de la langue à modéliser. Si le corpus est suffisamment grand, il permet de faire l'hypothèse qu'il reflète la langue en général et ainsi le modèle de langue correspond approximativement au modèle de langue pour le corpus.

La calcul de la probabilité d'un mot w dans un corpus C se base sur l'estimation de vraisemblance maximale du terme w :

$$\mathcal{P}_{\mathcal{ML}}(w) = \frac{|w|}{\sum_{w_j \in C} w_j} = \frac{\textit{nombre d' occurrences du terme } w \textit{ dans le corpus } C}{\textit{le nombre de } n \textit{ - grammes de } C}$$

Par simplification d'écriture, on note $\mathcal{P}(t)$ au lieu de $\mathcal{P}_{\mathcal{ML}}(t)$ en donnant au préalable la langue modélisée.

Le principe général des modèles de langue se résume donc en deux grandes étapes : l'apprentissage du modèle de langue et l'évaluation de la probabilité d'appartenance d'un document à ce modèle (cf. Figure 20). Ainsi, les paramètres du modèle de langue M_{Lx} s'estiment (2) en se basant sur les caractéristiques statistiques de langue extraites du corpus d'entraînement (1). A partir de ce modèle de langue M_{Lx}, la probabilité du document D d'appartenir à la langue X s'évalue par $P(D \mid M_{Lx})$ (3).

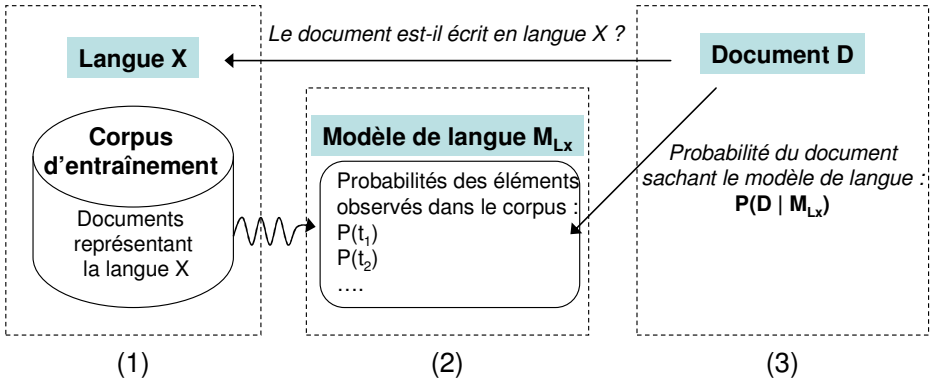


Figure 20. Principe général des modèles de langue

3. Les modèles de langue en recherche d'information

Le principe des modèles de langue en recherche d'information consiste à déterminer la probabilité de génération de la requête Q à partir du document $\mathcal{D}$: $\mathcal{P}(Q|\mathcal{D})$ alors que dans les modèles probabilistes classiques, on évalue $\mathcal{P}(\textit{pertinent} \mid \mathcal{D}, Q)$.

3.1. Principe général des modèles de langue en recherche d'information

Dans les approches de recherche d'information basées sur les modèles de langue, on considère que la pertinence d'un document face à une requête est en rapport avec la probabilité que la requête puisse être générée par le modèle de langue du document. Le principe de ces systèmes se base lui sur la détermination de la probabilité qu'une requête soit générée par le modèle de langue d'un document.

Pour évaluer cette probabilité, trois approches principales sont envisageables :

- Calculer la probabilité que la requête soit générée par le modèle de langue du document : $P(Q \mid M_D)$
- Calculer la probabilité que le document soit généré par le modèle de langue de la requête : $P(D \mid M_Q)$
- Comparer les modèles de langue de la requête et du document : M_Q et M_D

3.1.1 *Probabilité que la requête soit générée par le modèle de langue du document : $P(Q \mid M_D)$*

Un document D est vu comme la représentation d'un langage auquel correspond un modèle de langue M_D. Le score du document face à une requête Q se caractérise par la probabilité que son modèle génère la requête :

$$Score(Q,D) = P(Q \mid M_D)$$

Une requête est une suite de mots : $Q = t_1 t_2 \ldots t_n$ ce qui permet d'avoir $Score(Q,D) = P(t_1 t_2 \ldots t_n \mid M_D)$.

Ce principe est celui utilisé par la plupart des systèmes de recherche d'information basée sur les modèles de langue.

3.1.2 *Probabilité que le document soit généré par le modèle de langue de la requête: $P(D \mid M_Q)$*

Une autre approche possible consiste à créer un modèle de langue pour la requête et à déterminer le score d'un document $D = t_1 t_2 \ldots t_m$ par la probabilité que le document puisse être généré par ce modèle :

$$Score(Q,D) = P(t_1 t_2 \ldots t_m \mid M_Q)$$

Cette formule demeure peu utilisée car elle crée un déséquilibre entre les documents longs et courts puisque l'on détermine la probabilité que la requête soit dans le modèle de document ; ainsi les documents longs sont avantagés. De ce fait, on utilise l'alternative suivante :

$P(M_Q \mid D) = \frac{P(D \mid M_Q)}{P(D \mid C)}$, c'est-à-dire chercher à déterminer la probabilité d'obtenir le modèle de langue de la requête sachant le document.

3.1.3 Comparaison des modèles de langue de la requête et du document : $\mathcal{M}_Q$ et $\mathcal{M}_D$

Une autre possibilité consiste à construire un modèle de langue pour le document $\mathcal{P}(. \mid \mathcal{M}_D)$ et un autre pour la requête $\mathcal{P}(. \mid \mathcal{M}_Q)$. Le score d'un document face à la requête se détermine par une comparaison entre les deux modèles.

3.2. Modélisation des systèmes de recherche d'information basé sur les modèles de langue

3.2.1 Modèle de document

a. Cas unigramme

Soit $\mathcal{D}1$ = l'ensemble des sous séquences contiguës de $\mathcal{D}$ de taille maximale 1.

$\mathcal{D}1 = \{(t_{d,1}), (t_{d,2}), \ldots, (t_{d,i}), \ldots, (t_{d,\mathcal{N}dI})\}$

On appelle $\mathcal{M}_{D,1}$ le modèle uni-gramme du document $\mathcal{D}$. On notera $\mathcal{M}_{D,1}$ par $\mathcal{M}_D$ en donnant préalablement $n = 1$.

$\mathcal{M}_D = \{(t_{d,i}, \mathcal{P}(t_{d,i})), \forall i \in [1 \ldots \mathcal{N}_{dI}]\}$ avec $t_{d,i} \in \mathcal{V}$ et $\mathcal{N}_{dI} \leq \mathcal{N}_d$

$\mathcal{M}_D = \{(t_{d,1}, \mathcal{P}(t_{d,1})), (t_{d,2}, \mathcal{P}(t_{d,2})), \ldots, (t_{d,i}, \mathcal{P}(t_{d,i})), \ldots, (t_{d,L}, \mathcal{P}(t_{d,\mathcal{N}dI}))\}$

avec $t_{d,i} \in \mathcal{V}$ et $\forall i, \forall j \; i \neq j, t_{d,i} \neq t_{d,j}$

$\mathcal{P}(t_{d,i}) = \| t_{d,i} \|$

$\| t_{d,i} \|$ = estimation de la vraisemblance maximale, c'est à dire la fréquence du terme dans le document $\mathcal{D}$

$\| t_{d,i} \| = tf_{td,i}$

$$\| t_{d,i} \| = \frac{|t_{d,i}|}{|\mathcal{D}|} = \frac{|t_{d,i}|}{\mathcal{N}_{\mathcal{D}}}$$

b. Cas n-gramme

Soit $\mathcal{D}_n$ = l'ensemble des sous séquences contiguës de $\mathcal{D}$ de taille maximale n.

$\mathcal{D}_n = \{(t_{d,1}), \ldots, (t_{d,1}, \ldots, t_{d,n}), (t_{d,2}, \ldots, t_{d,n+1}), \ldots, (t_{d,n-\mathcal{N}dI+1}, \ldots, t_{d,\mathcal{N}dI})\}$

On appelle $\mathcal{M}_{D,n}$ le modèle n-gramme du document $\mathcal{D}$. On notera $\mathcal{M}_{D,n}$ par $\mathcal{M}_D$ en donnant préalablement n.

$\mathcal{M}_D = \{([t_{d,i-n+1}, \ldots, t_{d,i}], \mathcal{P}([t_{d,i-n+1}, \ldots, t_{d,i}])), \forall i \in [1 \ldots \mathcal{N}_{dI}]\}$ avec $t_{d,i} \in \mathcal{V}$ et $\mathcal{N}_{dI} \leq \mathcal{N}_d$

$\mathcal{M}_D = \{([t_{d,1}, \ldots, t_{d,1+n}], \mathcal{P}([t_{d,1}, \ldots, t_{d,n+1}])), ([t_{d,2}, \ldots, t_{d,n+2}], \mathcal{P}([t_{d,2}, \ldots, t_{d,n+2}])), \ldots, ([t_{d,i-n+1}, \ldots, t_{d,i}], \mathcal{P}([t_{d,i-n+1}, \ldots, t_{d,i}])), \ldots, ([t_{d,\mathcal{N}dI-n+1}, \ldots, t_{d,\mathcal{N}dI}], \mathcal{P}([t_{d,\mathcal{N}dI-n+1}, \ldots, t_{d,\mathcal{N}dI}]))\}$

$\mathcal{P}([t_{d,i-n+1}, \ldots, t_{d,i}]) = \left\| [t_{d,i-n+1}, \ldots, t_{d,i}] \right\|$

$\left\| [t_{d,i-n+1}, \ldots, t_{d,i}] \right\|$ = *fréquence d' apparitions du n - gramme*

avec $\left\| [t_{d,i-n+1}, \ldots, t_{d,i}] \right\| = tf_{[t_{d,i-n+1}, \ldots, t_{d,i}]}$

$$\left\| [t_{d,i-n+1}, \ldots, t_{d,i}] \right\| = \frac{\left| t_{d,i-n+1}, \ldots, t_{d,i} \right|}{\left| \mathcal{D} \right|} = \frac{\left| t_{d,i-n+1}, \ldots, t_{d,i} \right|}{\mathcal{N}_{\mathcal{D}}}$$

3.2.2 Principe de correspondance

La fonction de correspondance se base sur le principe de génération de la requête par le modèle de document. Ainsi, le calcul de la probabilité $\mathcal{P}(Q \mid \mathcal{M}_{\mathcal{D}})$ que la requête Q soit générée par $\mathcal{M}_{\mathcal{D}}$ s'effectue pour chaque document $\mathcal{D}$:

$$\mathcal{P}(Q \mid \mathcal{M}_{\mathcal{D}}) = \mathcal{P}(t_{q,1}\, t_{q,2} \ldots t_{q,\mathcal{N}q} \mid \mathcal{M}_{\mathcal{D}})$$

3.2.3 Fonction de correspondance

a. Cas unigramme

Soit le modèle unigramme : $\mathcal{M}_{\mathcal{D}} = \{(t_{d,1}, \mathcal{P}(t_{d,1})), (t_{d,2}, \mathcal{P}(t_{d,2})), \ldots, (t_{d,i}, \mathcal{P}(t_{d,i})), \ldots, (t_{d,\mathcal{N}dI}, \mathcal{P}(t_{d,\mathcal{N}dI}))\}$

On suppose l'indépendance des termes de la requête.

Dans le cas d'un modèle uni-gramme, la fonction de correspondance se généralise par :

Pour chaque $\mathcal{D} \in C$: $\mathcal{P}(Q \mid \mathcal{M}_{\mathcal{D}}) = \prod_{i=1}^{\mathcal{N}_q} \mathcal{P}(t_{q,i})$

$\mathcal{P}(Q \mid \mathcal{M}_{\mathcal{D}}) = \mathcal{P}(t_{q,1}) * \mathcal{P}(t_{q,2}) * \ldots * \mathcal{P}(t_{q,i}) * \ldots * \mathcal{P}(t_{q,\mathcal{N}q})$

$\mathcal{P}(t_{q,i}) = \mathcal{P}(t_{q,i} \mid \mathcal{M}_{\mathcal{D}})$ *par abus d'écriture*

La probabilité se base sur le poids du terme de la requête dans le modèle de document. Il existe une relation d'égalité entre terme du document et terme de la requête. Si un terme de la requête n'apparaît pas dans le document, une fonction de lissage est utilisée pour évaluer $\mathcal{P}(t_{q,i} \mid \mathcal{M}_{\mathcal{D}})$ et ainsi éviter une probabilité nulle pour l'ensemble de la requête.

Dans le cas de modèle uni-gramme et avec un lissage par interpolation, le calcul de la probabilité d'un terme de la requête sachant un modèle de document s'exprime donc ainsi :

- Pour tout $t_{q,i} \in Q$, tel que $\exists\ t_{d,j} \in \mathcal{M}_{\mathcal{D}}$, $t_{d,j} = t_{q,i}$: $\mathcal{P}(t_{q,i} \mid \mathcal{M}_{\mathcal{D}}) = \mathcal{P}_{\mathcal{MD}}(t_{d,j}) + \mathcal{P}_{\mathcal{MC}}(t_{d,j})$

- Pour tout $t_{q,i} \in Q$, tel que $\forall\ t_{d,j} \in \mathcal{M}_{\mathcal{D}}$, $t_{d,j} \neq t_{q,i}$: $\mathcal{P}(t_{q,i} \mid \mathcal{M}_{\mathcal{D}}) = \mathcal{P}_{\mathcal{MC}}(t_{d,j})$

Deux dimensions entre donc en compte dans l'estimation de la probabilité qu'une requête soit générée par un document : la probabilité pour chaque terme d'appartenir au modèle de

document $\mathcal{M}_D$ et la probabilité pour ces mêmes termes d'appartenir au modèle de corpus $\mathcal{M}_C$.

b. Cas n-gramme

Soit le modèle n-gramme : $\mathcal{M}_D = \{([t_{d,1}, \ldots, t_{d,1+n}], \mathcal{P}([t_{d,1}, \ldots, t_{d,1+n}])), ([t_{d,2}, \ldots, t_{d,n+2}], \mathcal{P}([t_{d,2}, \ldots, t_{d,n+2}])), \ldots, ([t_{d,i-n+1}, \ldots, t_{d,i}], \mathcal{P}([t_{d,i-n+1}, \ldots, t_{d,i}])), \ldots, ([t_{d,\mathcal{N}dI-n+1}, \ldots, t_{d,\mathcal{N}dI}], \mathcal{P}([t_{d,\mathcal{N}dI-n+1}, \ldots, t_{d,\mathcal{N}dI}]))\}$

$$\text{Pour chaque } D \in C : \mathcal{P}(Q \mid \mathcal{M}_D) = \prod_{i=1}^{\mathcal{N}_q} \mathcal{P}\left(t_{q,i} \middle| t_{q,i-n+1} \ldots t_{q,i-1}\right)$$

$\mathcal{P}(Q \mid \mathcal{M}_D) = \mathcal{P}(t_{q,1}) * \ldots * \mathcal{P}(t_{q,i} \mid t_{q,i-n+1} \ldots t_{q,i-1}) * \ldots * \mathcal{P}(t_{q,\mathcal{N}q} \mid t_{q,\mathcal{N}q-n+1} \ldots t_{q,\mathcal{N}q-1})$

$$\mathcal{P}\left(t_{q,i} \middle| t_{q,i-n+1} \ldots t_{q,i-1}\right) = \frac{\mathcal{P}\left(t_{q,i-n+1}\, t_{q,i-n+2} \ldots t_{q,i}\right)}{p\left(t_{q,i-n+1} \ldots t_{q,i-1}\right)} = \frac{\mathcal{P}\left(t_{d,i-n+1}\, t_{d,i-n+2} \ldots t_{d,i}\right)}{p\left(t_{d,i-n+1} \ldots t_{d,i-1}\right)}$$

Le problème de termes inconnus présent pour les modèles uni-gramme se rencontre également au niveau des modèles n-grammes. Afin d'éviter les probabilités nulle, une fonction de lissage est également introduite dans la fonction de correspondance.

Dans la pratique, en recherche d'information, les modèles uni-grammes restent les plus utilisés compte tenu de leur performance. Les modèles bi-grammes s'avérant coûteux, peu de systèmes de recherche d'information les utilisent actuellement.

3.2.4 Lissage

Plus le document utilisé est grand, plus la probabilité que l'ensemble des termes de la requête soit présent est grande. Toutefois, certains documents ne couvrent pas l'ensemble des termes de la requête et malgré leur pertinence ont un score nul en réponse à la requête puisque le score de la requête est basé sur un produit des probabilités que les termes de la requête appartiennent au modèle de document.

En effet, la probabilité se base sur le poids du terme de la requête dans le modèle de document. Il existe une relation d'égalité entre terme du document et terme de la requête. Comme certains termes de la requête peuvent ne pas apparaître dans le document, une fonction de lissage permet d'évaluer $\mathcal{P}(t_{q,i} \mid \mathcal{M}_D)$ et ainsi éviter une probabilité nulle pour l'ensemble de la requête.

Le principe du lissage s'exprime comme suit : une partie des probabilités des n-grammes du document est enlevée et affectée aux n-grammes absents du document, évitant une probabilité nulle pour les n-grammes absents du document.

Soit le modèle de langue $\mathcal{M}_D$ du document D composé de $\mathcal{N}$ n-grammes :

$\mathcal{M}_D = \{(n\text{-gramme1}, \mathcal{P}(n\text{-gramme1})), \ldots, (n\text{-grammeN}, \mathcal{P}(n\text{-grammeN}))\}$

- $\sum_{i=1}^{N} P(n-gramme\_i) = 1$ *si pas de lissage*
- $\sum_{i=1}^{N} P(n-gramme\_i) = 1-\alpha < 1$ *si lissage et* α *correspond à la masse de probabilité redistribuée pour les n-grammes inconnus.*

Les méthodes de lissage les plus classiques sont : le lissage de Laplace, le lissage Good-Turing, le lissage Backoff et le lissage par interpolation.

Le lissage de Laplace des probabilités consiste à ajouter une quantité faible identique à toutes les valeurs (généralement 1 dans le cas des modèles de langue pour la recherche d'information), si bien qu'aucune valeur n'est nulle.

Le lissage Good-Turing redéfinit le nombre d'apparitions des termes. Ainsi, un terme apparu r fois voit son nombre d'apparitions convertit en r^*, tel que :

$r^* = (r+1)\frac{n_{r+1}}{n_r}$ avec n_r = le nombre de termes apparaissant r fois.

Le lissage Backoff est une méthode dite de repli sur des modèles n-grammes d'ordre inférieur. Ainsi, si la probabilité d'un mot ne peut s'exprimer avec ses deux prédécesseurs, c'est-à-dire avec des tri-grammes alors on utilise une prédiction basée sur un seul prédécesseur, c'est-à-dire les bi-grammes et ainsi de suite sur les uni-grammes jusqu'à obtenir une prédiction.

Sur le même principe que le lissage Backoff, Jelinek propose un lissage par interpolation combinant systématiquement des modèles n-grammes de niveaux inférieurs.

3.2.5 Estimation de $\mathcal{P}(Q \mid \mathcal{M}_D)$

En recherche d'information, les modèles de langue ont été introduits en 1998 par Ponte et Croft [Ponte, 1998] qui voient le score d'un document face à une requête comme la probabilité que la requête soit générée par le modèle du document. Ainsi,

$Score(\mathcal{D}, Q) = \mathcal{P}(Q \mid \mathcal{M}_D)$.

Plusieurs façons d'exprimer ce score demeurent envisageables.

Ponte et Croft [Ponte, 1998] combine un modèle de langue du document et un modèle de langue du corpus. Dans cette approche, non seulement la probabilité des mots d'appartenir à la requête est prise en compte mais également la probabilité de ne pas rencontrer les mots n'appartenant pas à la requête entre en jeu. Ceci permet de différencier un document contenant beaucoup de sujets de la requête d'un document en contenant peu par la prise en compte des mots absents de la requête. Toutefois, cette approche s'avère coûteuse en calcul si le nombre de termes absents de la requête est important, ce qui est généralement le cas.

Hiemstra [Hiemstra, 1998] propose d'évaluer le score du document face à une requête en utilisant une approche par interpolation :

$$Score(D,Q) = \prod_{t \in Q} (\alpha \ P_{ML}(t_i|D) + (1-\alpha) P_{ML}(t_i|C))$$

Miller [Miller, 1998] [Miller, 1999] reformule le modèle de Hiemstra à l'aide d'un modèle de Markov caché à deux états (cf. Figure 21) :

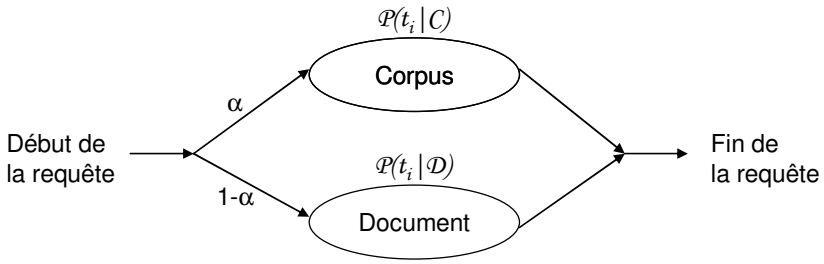


Figure 21. Modèle de Markov caché à deux états [Boughanem, 2004]

Enfin, Ng [Ng, 1999] offre une variante du modèle de la langue basée sur le ratio de vraisemblance :

$$Score(D,Q) = \sum_{t \in Q} log \frac{\alpha \ P_{ML}(t|D) + (1-\alpha) P_{GT}(t|C)}{P_{GT}(t|C)}$$

$P_{GT}(t|C)$ correspond à un lissage de type Good-Turing.

4. Conclusion

4.1. Bilan des modèles présentés

Ce chapitre établit un panorama des différents modèles classiquement utilisés en recherche d'information.

Le premier modèle apparu dans les années 1950 correspond au modèle booléen basé sur la théorie des ensembles. Ce modèle représente un document comme un ensemble de termes et une requête comme une expression logique de termes. La réponse à une requête booléenne s'exprime soit par 0, le document n'est pas pertinent pour la requête, soit par 1, le document est pertinent pour la requête. Ce type de modèle présente deux inconvénients principaux. Tout d'abord, tous les termes du document et de la requête sont pondérés de la même manière : 0 (terme absent) ou 1 (terme présent). La pondération ne tient pas compte de la fréquence des termes dans les documents et requêtes. Ensuite, une réponse de type binaire ne permet pas un ordonnancement des documents selon leur pertinence face à la requête. Ceci a le désavantage de fournir un ensemble non ordonnée de documents

pertinents à l'utilisateur. L'utilisateur ayant exprimé un besoin d'information doit encore fouiller dans cet ensemble de documents pour trouver les documents qui l'intéressent.

Le modèle vectoriel proposé par Salton dans les années 1970 demeure incontournable en recherche d'information. Dans ce modèle, des vecteurs de poids représentent document et requête. Chaque poids dans le vecteur désigne l'importance d'un terme correspondant dans le document ou dans la requête. Contrairement au modèle booléen, le modèle vectoriel permet une pondération des termes et un ordonnancement des documents selon leur pertinence vis-à-vis de la requête.

La recherche d'information s'avérant un processus incertain et imprécis, une incertitude dans la représentation des documents et dans l'expression des besoins de l'utilisateur existe. Ainsi les modèles probabilistes tentent d'estimer la probabilité qu'un document donné soit pertinent pour une requête donnée.

Une autre façon de modéliser les documents sous forme probabiliste apparaît avec les modèles de langue. Ces modèles, initialement utilisés dans les domaines de la reconnaissance de la parole et de la traduction automatique de texte, fournissent une représentation du langage. Le principe général des modèles de langue consiste à représenter un document par un modèle de langue en considérant qu'un document peut être vu comme une 'langue' particulière. On évalue ensuite la probabilité que la requête soit générée par ce modèle. On obtient la probabilité que la requête soit générée par chacun des documents du corpus. Cette probabilité permet d'établir un ordonnancement des documents selon leur pertinence face à la requête.

Les modèles de langue uni-gramme ont montré leur efficacité sur les tâches de recherche d'information. Ces modèles tiennent compte de la probabilité des termes de la requête dans l'ensemble du corpus, on parle de lissage. Ce lissage permet d'éviter les probabilités nulles pour des documents dans le cas où un mot de la requête s'avère absent du document mais que tous les autres mots demeurent présents. La fonction de correspondance utilisée dans le modèle de Hiemstra se base sur une approche par interpolation :

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

Ainsi la probabilité d'un terme de la requête d'appartenir au document s'exprime par la probabilité du terme dans le modèle du document plus la probabilité du terme dans le modèle du corpus. Nous proposons que la probabilité du terme de la requête soit exprimée par les deux probabilités citées précédemment mais également par la probabilité que le terme soit 'presque' présent dans le document. C'est ce que nous montrons dans notre modèle.

4.2. Modèle de langue et données incertaines

Le modèle de langue adapté à la recherche d'information se base sur la probabilité que la requête puisse être générée par le modèle de langue du document. De plus, de manière indirecte, les modèles de langue tiennent compte à la fois du fait qu'un terme peut être présent ou absent dans un document par le biais des fonctions de lissage. En effet, le poids d'un terme de la requête dans un document s'exprime par la combinaison de la probabilité du terme dans le modèle du document et une fonction de lissage pour éviter les scores nuls

lorsque le terme est absent du modèle de document. Compte tenu de ces deux aspects absence et présence du terme de la requête dans le document, le choix du modèle de langue nous parait judicieux car en y ajoutant la notion de 'presque égalité' du terme, les trois dimensions présence, absence et approximation du terme sont intégrées.

Nous proposons donc d'ajouter la notion de 'presque égalité' au modèle de langue afin qu'il prenne en compte les données incertaines. Pour cela, nous choisissons d'enrichir l'approche par interpolation de Hiemstra qui tient compte de la probabilité du terme de la requête dans le document $\mathcal{P}_{ML}(t_i|\mathcal{D})$ et dans le corpus $\mathcal{P}_{ML}(t_i C)$, en ajoutant la prise en compte de la probabilité des 'presque égalités' du terme dans le document.

Partie 2 : Proposition d'un modèle de recherche d'information adapté aux données incertaines

La recherche d'information se base sur la relation d'égalité qu'il peut exister entre termes de la requête et termes du document. La présence d'incertitude dans les données remet en cause cette égalité.

Dans cette partie nous remettons donc en cause la véracité de cette égalité de base dans certains contextes : les données incertaines. Dans un contexte incertain, des termes s'avèrent mal reconnus ou identifiés impliquant de mauvaises performances des systèmes de recherche d'information. Nous avons vu dans l'état de l'art que les performances de systèmes de recherche d'information diminuaient en présence d'incertitude. Il est donc nécessaire de repenser un système de recherche d'information dans le but de l'adapter à ce type de données. Afin de palier ce problème, nous proposons de substituer la 'presque égalité' à l'égalité habituellement utilisée (cf. Figure 22).

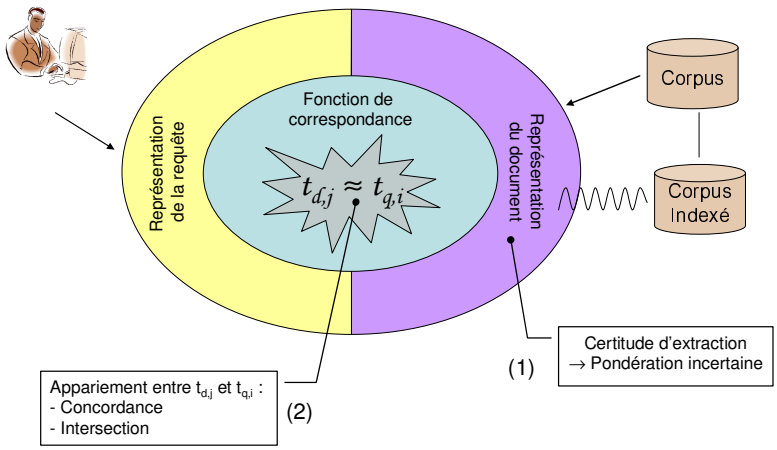


Figure 22. Remise en cause de l'égalité de base au centre d'un système d'information

Nous suggérons la prise en compte de l'incertitude dans le système de recherche d'information à deux niveaux : dans la représentation des documents et dans la fonction de correspondance. Nous introduisons donc la notion de certitude d'extraction de chaque terme au sein de la fonction de pondération (1) et le concept d'appariement au sein de la fonction de correspondance (2). L'appariement permet de mesurer la 'presque égalité' entre deux termes par le biais de la concordance et l'intersection.

Le chapitre III fixe les définitions et notations des concepts nécessaire à la mise en place d'un système de recherche d'information adapté aux données incertaines, à savoir la 'presque égalité', la certitude, l'incertitude et l'appariement.

Le chapitre IV décrit notre proposition de système de recherche d'information adapté aux données incertaines. Pour cela, nous montrons l'intégration de l'incertitude dans la fonction de pondération utilisée pour la représentation des documents et de l'appariement dans la fonction de correspondance.

La chapitre V termine cette partie en illustrant une instanciation de l'appariement avec des données issues de l'oral.

Chapitre III. Préliminaires : définitions et notations

La définition formelle d'un modèle de recherche d'information passe par la description de différents objets, à savoir un corpus de documents, un vocabulaire d'indexation, un modèle de documents, une représentation de la requête et une fonction de correspondance.

Le début de ce chapitre décrit notre proposition de modèle de recherche d'information et pose les notations utilisées dans le Chapitre IV. Pour la modélisation que nous proposons des données incertaines, la suite du chapitre fixe les notions nécessaires à la mise en place du modèle de recherche d'information adapté aux données incertaines : incertitude, certitude, appariement. Nous définissons le concept de 'presque égalité' et la certitude associée à chaque terme. Nous décrivons ensuite l'appariement entre deux termes. Pour cela, nous caractérisons la concordance et l'intersection entre deux termes.

1. Un modèle de recherche d'information

1.1. Corpus de documents

Soit un corpus C composé de documents $\mathcal{D}_i$.

La cardinalité du corpus C est $\mathcal{N}_C$.

$C = \{\mathcal{D}_i, 1 < i \leq \mathcal{N}_C\}$

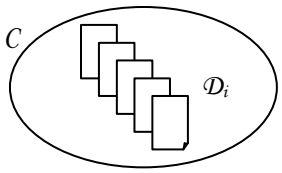


Figure 23. Corpus C de documents $\mathcal{D}_i$

Exemple :

Soit le corpus de documents $C = \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_5\}$ et $\mathcal{N}_C = 5$

1.2. Vocabulaire

$\mathcal{V}$ est un ensemble fermé de termes :

$\mathcal{V} = \{t_1, t_2, \ldots, t_{\mathcal{N}v}\}$

$\| \mathcal{V} \| = \mathcal{N}_{\mathcal{V}}$

1.3. Document et document indexé

1.3.1 Document

Un document $\mathcal{D}$ est une séquence de $\mathcal{N}_d$ termes :

$\mathcal{D} = [t_{d,1}, t_{d,2}, t_{d,3}, \ldots, t_{d,\mathcal{N}d}]$ avec $t_{d,i} \in \mathcal{V}$

1.3.2 Document indexé

Soit un vocabulaire d'indexation $\mathcal{V}_I$ tel que :

$\| \mathcal{V}_I \| = \mathcal{N}_{\mathcal{V}I}$

$\mathcal{V}_I = \{t_{v1}, t_{v2}, \ldots, t_{v\mathcal{N}\mathcal{V}I}\}$

Chaque document $\mathcal{D}_i$ du corpus $\mathcal{C}$ est associé à sa représentation $\mathcal{M}_{\mathcal{D}}$ basée sur $\mathcal{V}_I$:

$\mathcal{M}_{\mathcal{D}} = \{t_{vi}, 1 < i < \mathcal{N}_{\mathcal{V}I}\}$

Nous notons $\mathcal{C}I$ le corpus des documents indexés.

1.4. Requête et représentation de la requête

1.4.1 Requête

Une requête Q est une séquence de $\mathcal{N}_q$ termes quelconques :

$Q = [t_{q,1}\ t_{q,2}\ \ldots\ t_{q,\mathcal{N}q}]$ avec $\mathcal{N}_q \geq 1$.

1.4.2 Représentation de la requête

Soit un vocabulaire de représentation de la requête $\mathcal{V}_q$:

$\| \mathcal{V}_q \| = \mathcal{N}_{\mathcal{V}q}$

Nota bene : dans beaucoup de systèmes, $\mathcal{V}_q$ est égal au vocabulaire $\mathcal{V}_I$.

dans le cas où les termes de la requête sont associés à une valeur de certitude, on a :

$\mathcal{V}_q = \mathcal{V} \times \mathcal{C}e$ avec $\mathcal{C}e = \{c_i, c_i \in \Re^+\}$

Nous noterons $\mathcal{C}Q$ l'ensemble des requêtes possibles.

1.5. Pertinence entre document et requête

La fonction de correspondance met en relation un document indexé avec une requête et permet d'établir soit :

- si un document est pertinent pour une requête par une réponse booléenne
- combien un document est pertinent pour une requête en ordonnançant les documents selon leur valeur de pertinence.

Nous nous intéressons au 2ème cas.

On définit la fonction $\mathcal{P}ertinence$ suivante :

$$\begin{aligned} \mathcal{P}ertinence : \quad & CQ \times CI \quad \rightarrow \quad \Re^+ \\ & Q \times \mathcal{M}_{\mathcal{D}} \quad \rightarrow \quad n \end{aligned}$$

2. 'Presque égalité' entre termes

2.1. Introduction

Dans le cas où le processus d'extraction de l'information fournit des données incertaines, l'égalité $t_{d,j} = t_{q,i}$ entre les termes du document et ceux de la requête ne s'avère pas toujours possible. Imaginons un document $\mathcal{D}$ initial, sa représentation $\mathcal{D}_i$ (fournie par le processus d'extraction) et une requête Q :

$\mathcal{D}$ = « Paul est à l'honneur »
$\mathcal{D}_i$ = « Paul est talonneur » d'où $\mathcal{D}_i = \{t_{d,1} = Paul, t_{d,2} = est, t_{d,3} = talonneur\}$
Q = « honneur » d'où $Q = \{t_{q,1} = honneur\}$

Dans cet exemple, l'égalité $t_{d,j} = t_{q,i}$ ne se vérifie pas alors que le terme de la requête « honneur » s'avère bien présent dans le document initial $\mathcal{D}$.

De ce fait, se pose la question du devenir de l'égalité $\mathcal{P}(t_{q,i} | \mathcal{M}_{\mathcal{D}}) = \mathcal{P}(t_{d,j})$ dans un contexte incertain.

Plusieurs cas deviennent envisageables :

(i) **Le terme de la requête est présent** dans le document, d'où $t_{d,j} = t_{q,i}$

$\mathcal{D}$ = « Paul est à l'honneur »
$\mathcal{D}_i$ = « Paul est à l'honneur »
d'où $\mathcal{D}_i = \{t_{d,1} = Paul, t_{d,2} = est, t_{d,3} = à, t_{d,3} = le , t_{d,4} = honneur \}$
Q = « honneur » d'où $Q = \{t_{q,1} = honneur\}$

Ici $t_{d,4} = t_{q,1}$

(ii) **Le terme de la requête est « presque » présent** dans le document, on parle de « presque égalité », on a $t_{d,j} \approx t_{q,i}$

$\mathcal{D}$ = « *Paul est à l'honneur* »
$\mathcal{D}_i$ = « *Paul est talonneur* » *d'où* $\mathcal{D}_i = \{t_{d,1} = Paul, t_{d,2} = est, t_{d,3} = talonneur\}$
Q = « *honneur* » *d'où* $Q = \{t_{q,1} = honneur\}$

Ici $t_{d,3} \approx t_{q,1}$

(iii) Le terme de la requête est à la fois présent et « presque présent » dans le document, on a $t_{d,j} = t_{q,i}$ ET $t_{d,j} \approx t_{q,k}$:

$\mathcal{D}$ = « *Paul est à l'honneur. Cet honneur …* »
$\mathcal{D}_i$ = « *Paul est talonneur. Cet honneur …* »
d'où $\mathcal{D}_i = \{t_{d,1} = Paul, t_{d,2} = est, t_{d,3} = talonneur, t_{d,4} = cet , t_{d,5} = honneur \}$
Q = « *honneur* » *d'où* $Q = \{t_{q,1} = honneur\}$

Ici $t_{d,5} = t_{q,1}$ *et* $t_{d,3} \approx t_{q,1}$

2.2. Notations de la presque égalité et de $\sim t_i$

On note la 'presque égalité' entre deux termes par l'opérateur ≈.

Ainsi si $t_{d'}$ est 'presque égal' à t_q, on a $t_{d'} \approx t_q$.

$$\boxed{\forall t_i \in \mathcal{V},\ \exists \left\{ t_{d'} \in \mathcal{V} \,\middle|\, t_i \approx t_{d'} \right\}}$$

L'ensemble des 'presque égalité' de t_i se note : $\sim t_i$, ainsi $\boxed{\sim t_i = \left\{ t_{d'} \in \mathcal{V} \,\middle|\, t_i \approx t_{d'} \right\}}$.

3. Certitude du terme

Soit une valeur de certitude c associée à chaque terme $x \in \mathcal{V}$.

Nous représentons cette valeur par la fonction *Cert* :

Définition 1 : $\boxed{\begin{array}{l} Cert : \mathcal{V} \to \Re^+ \\ \qquad\quad x \to c \end{array}}$

Prenons comme exemple la phrase suivante dite à l'oral :

$\mathcal{P}$ = « *Paul est à l'honneur. Cet honneur …* »

Soit un processus d'extraction des données tel qu'un système de reconnaissance automatique de la parole qui fournit en sortie la phrase :

P_{sortie} = « *Paul est talonneur. Cet honneur … »*

Avec les indications suivantes :

Cert (Paul) = 0.6

Cert(est) = 0.8

Cert(talonneur) = 0.2

Cert(cet) = 0.73

Cert(honneur) = 0.45

O.6, 0.8, 0.2, 0.73, 0.45, correspondent respectivement aux valeurs de certitude associées aux termes Paul, est, talonneur, cet, honneur.

4. Appariement entre deux termes

4.1. Introduction

Soient deux termes x, y de longueur nx et ny sur lesquels seule leur chaîne de caractère est connue.

Ces deux termes s'*apparient* :

- selon leur *concordance*, c'est-à-dire leur positionnement relatif. Nous notons *Conc(x, y)* la concordance entre deux termes x et y.
- selon leur *intersection*, c'est-à-dire les zones communes aux deux termes. Nous notons *Inter(x, y)* l'intersection entre deux termes x et y.

4.2. Exemples

Soient deux termes x, y de longueur respective nx et ny.

<u>*1er cas*</u> : x et y sont alignés et de même longueur (cf. Figure 24)

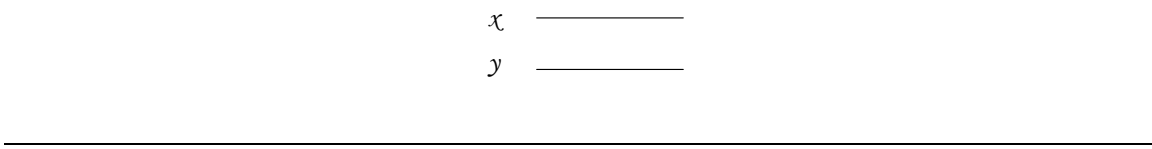


Figure 24. Soient deux termes x et y de longueur nx = ny

Cette situation correspond à celle classiquement utilisée en recherche d'information. On sait qu'il y a appariement entre x et y, si et seulement si $x = y$.

X concorde avec y et partagent x et y comme zone commune, ce que nous notons *Conc(x, y) = concorde* et *Inter(x, y) = (x, y)*.

L'intersection entre deux termes peut être imparfaite, il s'agit donc d'une zone de forte ressemblance entre x et y.

<u>2ème cas</u> : x et y sont alignés et $nx < ny$ (cf. Figure 25), on a alors :

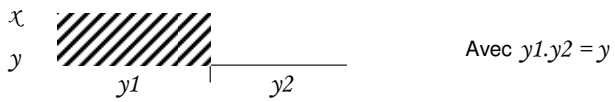


Figure 25. Soient deux termes x et y alignés de longueur nx < ny

On mesure l'appariement de x et y en tenant compte de leur intersection (zone hachurée) c'est-à-dire x pour l'un et *y1* pour l'autre.

Dans ce cas, x débute y et les termes partagent comme zone commune x et *y1* respectivement, ce que nous notons *Conc(x, y) = débute* et *Inter(x, y) = (x, y1)*.

<u>3ème cas</u> : x et y ne sont pas alignés et $nx < ny$ (cf. Figure 26), on peut par exemple avoir :

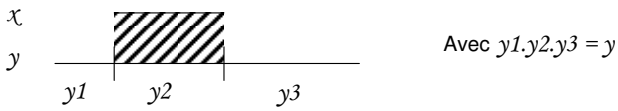


Figure 26. Soient deux termes x et y non alignés de longueur nx < ny

On mesure l'appariement de x et y en tenant compte de leur intersection (zone hachurée) et de la nature de leur concordance (ici x contient y).

Dans ce cas, x est pendant y et les termes partagent comme zone commune x et y2 respectivement, ce que nous notons *Conc(x, y) = pendant* et *Inter(x, y) = (x, y2)*.

4.3. Notation de l'appariement

Cette partie a pour objectif de présenter de façon générale l'appariement de tout terme x avec tout terme y compte tenu :

- de leur concordance
- de leur intersection.
-

L'appariement entre tout terme x et y est défini par la fonction App :

Définition 2 :

$$\begin{array}{lccc} App : & \mathcal{V}^2 & \rightarrow & \mathcal{A}\times C^2 \\ & (x, y) & \rightarrow & \big(Conc(x, y), Inter(x, y)\big) \end{array}$$

La valeur associée à l'appariement se définit ainsi :

Définition 3 :

$$\begin{array}{lccc} ValApp : & \mathcal{A}\times C^2 & \rightarrow & [0,1] \\ & \big(Conc(x, y), Inter(x, y)\big) & \rightarrow & ValConc(x, y)\times ValInter(x', y') \end{array}$$

Afin de décrire plus précisément le type d'égalité qu'il peut exister entre deux termes, une typologie des relations existant entre deux termes se montre nécessaire. Notre typologie de concordance entre deux termes s'inspire des relations de Allen. Les relations de Allen permettent de définir la *position temporelle* de tous les objets d'un document en plaçant des relations temporelles entre les objets.

4.4. Les relations de Allen

Le modèle de Allen se présente sous la forme d'un graphe complet où les arcs expriment des relations et les nœuds des intervalles [Allen, 1981]. Ce modèle définit les treize relations possibles entre deux intervalles. Il distingue des relations de connexité (précède, rencontre, chevauche), d'inclusion (débute, pendant, termine), de simultanéité (égal) et leurs inverses. Cette typologie est représentée dans le schéma ci-après (cf. Figure 27).

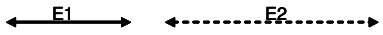
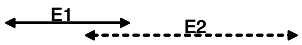
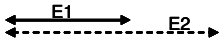
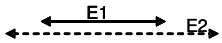
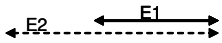
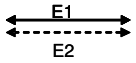
Schéma de relations	Signification	Symétrie	Notation
	E1 précède E2	E2 est_précédé_par E1	*E1 < E2, E2 > E1*
	E1 rencontre E2	E2 est_rencontré_par E1	*E1 m E2, E2 mt E1*
	E1 chevauche E2	E2 est_chevauché_par E1	*E1 o E2, E2 ot E1*
	E1 débute E2	E2 est_débuté_par E1	*E1 s E2, E2 st E1*
	E1 pendant E2	E2 contient E1	*E1 d E2, E2 dt E1*
	E1 termine E2	E2 est_terminé_par E1	*E1 e E2, E2 et E1*
	E1 égal E2		*E1 = E2*

Figure 27. Typologie des relations de Allen

Toutes les relations de Allen ne nous intéressent pas dans notre contexte de travail. Il est inutile de différencier les deux premières relations (précède et rencontre) dans notre contexte, elles se regroupent dans une même relation.

4.5. Concordance entre deux termes

4.5.1 Introduction

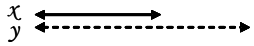
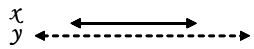
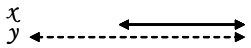
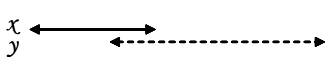
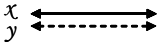
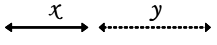
Schéma de relations	Signification	Symétrie	Notation
	*x* débute *y*	*y* est_débuté_par *x*	*Conc(x,y) = debute*
	*x* pendant *y*	*y* est_pendant *x*	*Conc(x,y) = pendant*
	*x* termine *y*	*y* est_termine_par *x*	*Conc(x,y) = termine*
	*x* chevauche *y*	*y* est_chevauché_par *x*	*Conc(x,y) = chevauche*
	*x* concorde *y*	*y* concorde *x*	*Conc(x,y) = concorde*
	*x* ne_concorde_pas *y*	*y* ne_concorde_pas *x*	*Conc(x,y) = ne_concorde_pas*

Figure 28. Typologie de Allen adaptée à la concordance

Cette typologie (cf. Figure 28) s'obtient en reprenant les relations de Allen qui ont un sens dans notre contexte. On ne parle plus de relations temporelles mais de ***concordance*** entre deux termes.

4.5.2 Nature de la concordance entre deux termes

L'ensemble $\mathcal{A}$ des concordances possibles entre deux termes x et y se définit par $\mathcal{A} = \{ debute,\ pendant,\ termine,\ chevauche,\ concorde,\ ne\_concorde\_pas \}$.

Nous définissons une fonction $Conc(x,y)$ qui détermine la nature de la relation de concordance entre deux termes x et y.

Définition 4 :

$$\begin{array}{|lccc|} \hline Conc : & \mathcal{V}^2 & \rightarrow & \mathcal{A} \\ & (x,y) & \rightarrow & a \\ \hline \end{array}$$

4.5.3 Notation de la valeur de la concordance

A chaque type de relation correspond une valeur $\alpha \in \Re^+$. Ainsi,

Définition 5 :

$$\begin{array}{|lccc|} \hline ValConc : & \mathcal{A} & \rightarrow & \Re^+ \\ & a & \rightarrow & \alpha \\ \hline \end{array}$$

Avec $ValConc(a) = \begin{cases} \alpha_1 & si\ \ a = débute \\ \alpha_2 & si\ \ a = pendant \\ \alpha_3 & si\ \ a = termine \\ \alpha_4 & si\ \ a = chevauche \\ \alpha_5 & si\ \ a = concorde \\ 0 & si\ \ a = ne\_concorde\_pas \end{cases}$

4.6. Intersection

4.6.1 Introduction

Deux termes x et y partagent une intersection. La Figure 29 montre les intersections de chaque concordance (telle que précédemment définie).

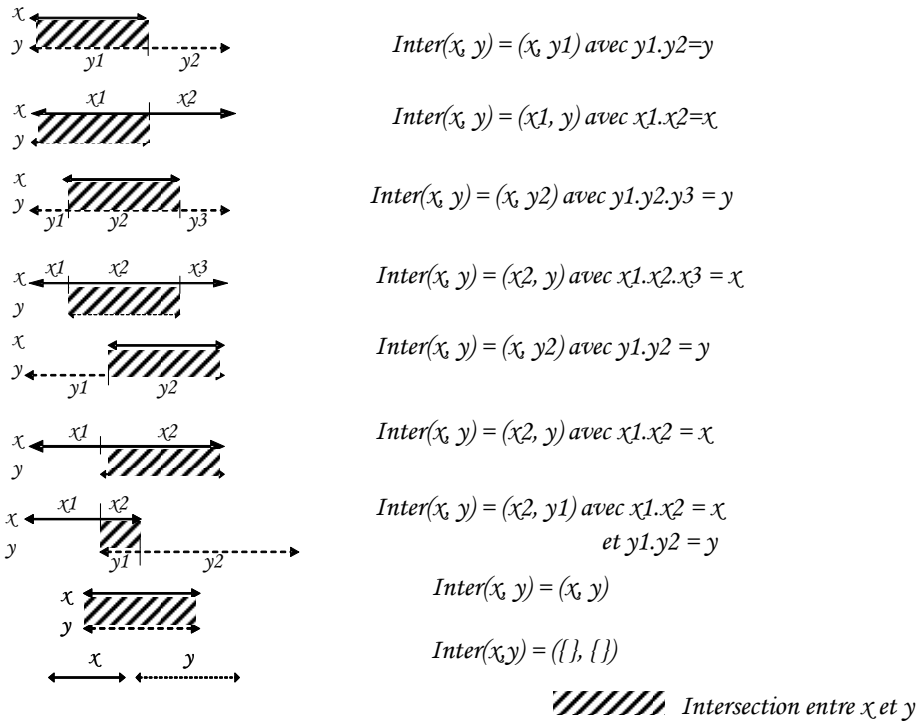


Figure 29. Intersection pour chaque concordance entre deux termes x et y

4.6.2 Notation de l'intersection

La nature d'une intersection entre deux termes x et y se définit comme suit :

Définition 6 :

$$\begin{array}{lccc} Inter : & C^2 & \rightarrow & C^2 \\ & (x, y) & \rightarrow & (x', y') \end{array}$$

Où x' et y' sont les zones communes de x et y (voir Figure 29).

4.6.3 Notation de la valeur de l'intersection

On définit la valeur de l'intersection entre deux termes $(x, y) \in \mathcal{V}^2$ liés par une concordance par la fonction *ValInter* :

Définition 7 :

$$\begin{array}{lccc} ValInter : & \mathcal{V}^2 & \rightarrow & [0,1] \\ & (x, y) & \rightarrow & \beta \end{array}$$

$$ValInter(x, y) = \begin{cases} 1 & si \quad x = y \\ 0 \leq ValInter(x, y) < 1 & sinon \end{cases}$$

La problématique de l'incertitude des données pose une question soulevée dans différents domaines. On peut citer notamment les systèmes qui permettent de déterminer si deux termes sont phonétiquement identiques (algorithme du Soundex). Les systèmes de correcteur orthographique se basent également sur la problématique de l'incertitude des données en cherchant à rapprocher 2 termes ayant des lettres en commun.

Ces différents algorithmes sont utilisables pour déterminer *ValInter(x, y)*.

4.7. Exemples

Deux exemples illustrent les différentes fonctions décrites précédemment.

Soient deux termes *w1* et *w2* avec *Cert(w1)* et *Cert(w2)* comme valeurs respectives de certitude d'extraction.

4.7.1 1er exemple : w1 = 'exemple' et w2 = 'exemple'

Ce premier exemple correspond à l'appariement entre deux termes égaux (cf. Figure 30).

Leur appariement se représente ainsi :

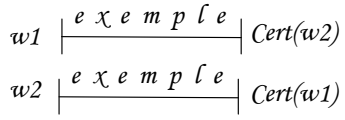


Figure 30. Appariement entre w1 et w2

a. Concordance :

Nature de la concordance :

Conc(w1, w2) = Conc('exemple', 'exemple') = concorde

Valeur de la concordance :

ValConc (Conc(w1, w2)) = ValConc (Conc('exemple', 'exemple')) = ValConc(concorde) = α_5

b. Intersection :

Nature de l'intersection :

Inter(w1, w2) = Inter('exemple', 'exemple') = ('exemple', 'exemple')

Valeur de l'intersection :

ValInter(Inter(w1, w2)) = ValInter(Inter('exemple', 'exemple') =

ValInter('exemple', 'exemple') = 1

4.7.2 2ème exemple : w1 = 'honneur' et w2 = 'talonneur'

Leur appariement se représente ainsi :

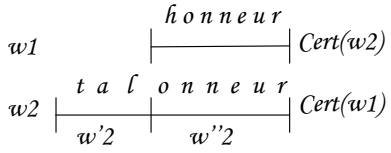


Figure 31. Appariement entre w1 et w2

a. Concordance :

Nature de la concordance :

Conc(w1, w2) = Conc('honneur', 'talonneur') = termine

Valeur de la concordance :

ValConc (Conc(w1, w2)) = ValConc(termine) = α_3

b. Intersection :

Nature de l'intersection :

Inter(w1, w2) = Inter('honneur', 'talonneur') = ('honneur', 'onneur')

Valeur de l'intersection :

ValInter(Inter(w1, w2)) = ValInter('honneur', 'onneur') < 1

4.7.3 3ème exemple : w1 = 'bon' et w2 =' bonheur'

Leur appariement se représente ainsi :

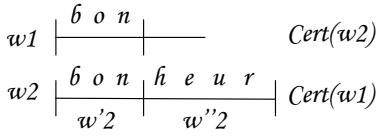


Figure 32. Appariement entre w1 et w2

a. Concordance :

Nature de la concordance :

Conc(w1, w2) = Conc('bon', 'bonheur') = débute

Valeur de la concordance :

ValConc (Conc(w1, w2)) = ValConc(débute) = α1

b. Intersection :

Nature de l'intersection :

Inter(w1, w2) = Inter('bon', 'bonheur') = ('bon', 'bon')

Valeur de l'intersection :

ValInter(Inter(w1, w2)) = ValInter('bon', 'bon') = 1

5. Conclusion

Ce chapitre fixe les définitions et notations nécessaires à notre proposition de modèle de recherche d'information adapté aux données incertaines décrit dans le chapitre suivant. Nous venons donc de définir les notions de presque égalité, de certitude des termes et d'appariement entre termes.

Chapitre IV. Modèle de recherche d'information adapté aux données incertaines

Ce chapitre décrit notre modèle de recherche d'information adapté aux données incertaines. Ce modèle s'appuie sur les spécificités liées à ce type de données. Ces caractéristiques entraînent la définition d'une fonction de correspondance utilisant la 'presque égalité' $t_{d,j} \approx t_{q,i}$.

Un système de recherche d'information repose sur trois parties essentielles : la représentation de la requête, la représentation du document sous forme de document indexé et une fonction permettant la mise en correspondance des termes de la requête avec ceux du document. Nous montrons dans cette partie de quelle manière la notion d'incertitude intervient au niveau de ces trois éléments constitutifs de tout système de recherche d'information.

Ainsi, après avoir décrit la représentation du document et de la requête, nous développons la fonction de correspondance utilisée dans le modèle de recherche d'information adapté aux données incertaines (cf. Figure 33). Nous montrons que la certitude due au processus d'extraction des données et associée à chaque terme entre dans le processus d'indexation des documents (3). De plus, dans ce contexte, la correspondance (1) entre termes de la requête et termes du document nécessite l'introduction de la notion d'appariement entre termes (2). L'appariement, par le biais de la concordance et l'intersection, détermine la 'presque égalité' existant entre deux termes. Nous finissons ce chapitre en montrant que dans le cas de données certaines, nous retrouvons une fonction de correspondance classiquement utilisée en recherche d'information.

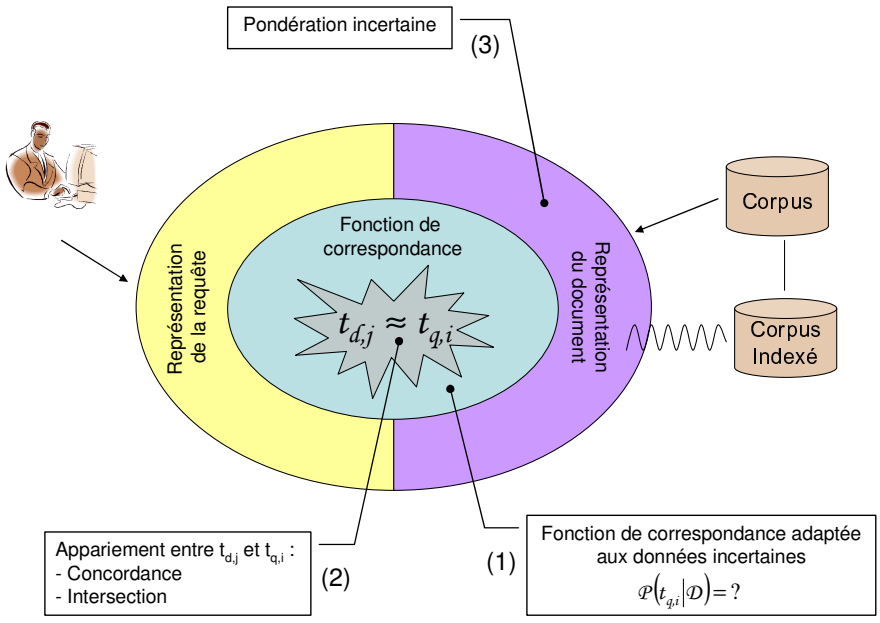


Figure 33. Plan du chapitre

1. Représentation du document

1.1. Document

Un document $\mathcal{D}$ est une séquence de longueur $\mathcal{N}_d$ telle que :

$\mathcal{D} = [(\text{mot}_1, c1), (\text{mot}_2, c2), (\text{mot}_3, c3), \ldots, (\text{mot}_{\mathcal{N}d}, c\mathcal{N}d)]$ avec $\text{mot}_i \in \mathcal{V}$ et $ci \in \Re^+$

avec $ci = \mathcal{C}ert(\text{mot}_i)$.

1.2. Modèle de document

Soit $\mathcal{D}I$ = l'ensemble des sous séquences contiguës de $\mathcal{D}$ de taille maximale 1 c'est-à-dire l'ensemble des termes de $\mathcal{D}$.

$\|\mathcal{D}I\| = \mathcal{N}_{dI}$ avec $\mathcal{N}_{dI} \leq \mathcal{N}_d$

$\mathcal{D}I = \{(t_{d,1}), (t_{d,2}), \ldots, (t_{d,i}), \ldots, (t_{d,\mathcal{N}dI})\}$

On appelle $\mathcal{M}_{\mathcal{D},1}$ le modèle **uni**-gramme du document $\mathcal{D}$. On simplifiera par la suite cette notation par $\mathcal{M}_{\mathcal{D}}$ en donnant préalablement la valeur de 1. Ce modèle uni-gramme représente l'ensemble des probabilités pour chaque uni-gramme du document.

$\mathcal{M}_{\mathcal{D},1} = \mathcal{M}_{\mathcal{D}} = \{(t_{d,i}, \mathcal{P}(t_{d,i})), \forall i \in [1...\mathcal{N}_{dI}]\}$ avec $t_{d,i} \in \mathcal{V}$ et $\mathcal{N}_{dI} \leq \mathcal{N}_d$

Rappel : on note $\mathcal{C}I$ l'ensemble des documents indexés.

1.3. Pondération des termes dans le modèle de document

Dans un contexte incertain, à chaque terme est associée une valeur de certitude. Les termes de la requête ne doivent pas être pondérés uniquement par leur importance, comme en recherche d'information classique, mais également par leur valeur de certitude.

Usuellement dans un modèle de langue, un comptage du nombre d'apparitions du terme dans le document $\mathcal{D}$ sert au calcul de la probabilité d'un terme $t_{d,i}$ dans le modèle de document, d'où :

$\mathcal{P}(t_{d,i}) = \| t_{d,i} \|$ avec

$\| t_{d,i} \| = \textit{estimation de la vraisemblance maximale}$

$$\| t_{d,i} \| = \frac{|t_{d,i}|}{|\mathcal{D}|} = \frac{|t_{d,i}|}{\mathcal{N}_d}$$

Dans un contexte où l'on n'a plus uniquement des termes mais des termes associés à des valeurs de certitude, le comptage ne s'effectue plus uniquement sur le nombre

d'apparitions du terme mais sur le nombre d'apparitions du terme tout en tenant compte de la certitude des termes. En résumé, on aura :

$$\mathcal{P}(t_{d,i}) = \frac{\sum_{t_{d,i}} Cert(t_{d,i})}{\mathcal{N}_d} \leq \frac{|t_{d,i}|}{\mathcal{N}_d}$$

On remarque que si $Cert(t_{d,i}) = 1$, on retrouve $\mathcal{P}(t_{d,i}) = \frac{|t_{d,i}|}{\mathcal{N}_d}$.

2. Représentation de la requête

Une requête Q est une séquence de longueur $\mathcal{N}_q$:

$Q = [(t_{q,1}, c_1), (t_{q,2}, c_2), (t_{q,3}, c_3), \ldots, (t_{q,\mathcal{N}q}, c_{\mathcal{N}q})]$ avec $t_{q,i} \in \mathcal{V}$ et $c_i \in \Re^+$

avec $c_i = Cert\ (t_{q,i})$.

3. Principe de correspondance

3.1. Caractéristiques

La fonction de correspondance doit tenir compte du fait que non seulement un terme $t_{q,i}$ est mis en relation avec les termes $t_{d,j}$ du document $\mathcal{D}$ tel que $t_{q,i} = t_{d,j}$ mais aussi avec les termes $t_{d,k}$ du document $\mathcal{D}$ approximant le terme $t_{q,i}$ tel que $t_{q,i} \approx t_{d,k}$.

3.2. Liens entre termes du document et termes de la requête

Comme dans la plupart des modèles de langue utilisés en recherche d'information, on suppose la simplification suivante: *les mots d'une requête sont indépendants*.

La fonction de correspondance associée au modèle de langue traite à la fois le cas où le terme de la requête est présent dans le document et le cas où il est absent de ce document par le biais des fonctions de lissage. Dans notre contexte de travail, nous prenons en compte trois dimensions présence du terme dans le document, absence du terme dans le document et approximation du terme dans le document. La pondération d'un terme d'une requête dans un document est donc de la forme :

$P(t_q|\mathcal{M}_\mathcal{D})$ = prise en compte de la présence du terme dans le document + prise en compte de l'absence du terme dans le document + prise en compte des approximations du terme dans le document.

De ce fait, en prenant en compte l'incertitude des termes, pour un terme $t_{q,i}$ de la requête Q, on définit, à partir d'un terme $x \in \mathcal{V}$ et d'un document $\mathcal{M}_D \in \mathcal{CD}$, les fonctions suivantes :

3.2.1 Le terme de la requête est présent dans le document :

Nous définissons une fonction $Présence(\mathcal{M}_D, x)$:

Définition 8 :

$$\begin{aligned} Présence : CI \times \mathcal{V} &\rightarrow \mathcal{V} \\ (\mathcal{M}_D, x) &\rightarrow t_j \in \mathcal{M}_D \mid Conc(x, t_j) = Concorde \; ET \; ValInter(Inter(x, t_j)) = 1 \end{aligned}$$

Cette fonction détermine s'il existe un terme t_j appartenant au modèle de document $\mathcal{M}_D$ tel que $t_j = x$.

Si le terme de la requête n'est pas présent dans le document, $Présence(\mathcal{M}_D, t_{q,i}) = w0$ avec $w0$ le terme vide.

3.2.2 Le terme de la requête est absent du document :

Nous définissons une fonction $Absence(\mathcal{M}_D, x)$:

Définition 9 :

$$\begin{aligned} &Absence : CI \times \mathcal{V} \rightarrow \{0,1\} \\ &(\mathcal{M}_D, x) \rightarrow Absence(\mathcal{M}_D, x) = \begin{cases} 1 \; si \; \forall t_j \in \mathcal{M}_D, Conc(x, t_j) = ne\_concorde\_pas \\ 0 \; sinon \end{cases} \end{aligned}$$

Si tous les termes du modèle de document $\mathcal{M}_D$ ne concorde pas avec x alors le terme est absent du document.

Certains termes de la requête peuvent être absents du document. Pour éviter d'avoir un score nul pour la requête entière, une fonction de lissage devient nécessaire. Il existe différentes fonctions de lissage. Nous choisissons d'utiliser par la suite une fonction de lissage tenant compte de la présence du terme dans le corpus.

3.2.3 Des approximations du terme de la requête sont présentes dans le document :

Nous définissons la fonction d'approximation d'un terme $x \in \mathcal{V}$ avec un document $\mathcal{M}_D \in CI$ par $Approximation(\mathcal{M}_D, x)$.

Cette fonction $Approximation(\mathcal{M}_D, x)$ détermine l'ensemble des termes t_j du document $\mathcal{M}_D$ s'appariant avec x et tel que $x \neq t_j$.

Définition 10 :
$$\begin{array}{l} Approximation : \quad CI \times \mathcal{V} \rightarrow \mathcal{V}^* \\ (\mathcal{M}_D, x) \rightarrow \left\{ t_j \in \mathcal{M}_D \,\middle|\, Absence(\mathcal{M}_D, x) = 0 \text{ et } Présence(\mathcal{M}_D, x) \neq t_j \right\} \end{array}$$

L'ensemble des 'presque égalité' de t_i se note : $\sim t_i$, ainsi $\sim t_i = \left\{ t_{d'} \in \mathcal{V} \,\middle|\, t_i \approx t_{d'} \right\}$. D'où :

Définition 11 : $\sim t_i = Approximation(\mathcal{M}_D, t_i)$.

3.3. Fonction de correspondance

Nous utilisons le principe de génération de la requête par le document pour évaluer le score d'un document en réponse à une requête. Ainsi, en prenant une requête Q et un document $\mathcal{D}$ et compte tenu de l'hypothèse d'indépendance des termes, on obtient :

Définition 12 : $\mathcal{P}(Q|\mathcal{D}) = \prod_{t_{q,i} \in Q} \mathcal{P}(t_{q,i}|\mathcal{D})$

3.4. Principe de correspondance au niveau du terme : $\mathcal{P}(t_{q,i}|\mathcal{D})$

La probabilité $\mathcal{P}(t_{q,i}|\mathcal{D})$ prend en compte que :

- le terme $t_{q,i}$ est associé à une valeur de certitude : $Cert(t_{q,i})$
- le terme $t_{q,i}$ peut être présent, absent, approximé dans le document $\mathcal{D}$.

Afin de tenir compte des trois dimensions : *présence* du terme dans le document (1), *absence* du terme dans le document (2) et *approximations* du terme dans le document (3), nous utilisons l'approche suivante :

Définition 13 :
$$\mathcal{P}(t_{q,i}|\mathcal{D}) = Cert(t_{q,i}) \times \left[\underbrace{\mu\lambda \mathcal{P}(t_{q,i}|\mathcal{M}_D)}_{(1)} + \underbrace{\mu(1-\lambda)\mathcal{P}(t_{q,i}|\mathcal{M}_C)}_{(2)} + \underbrace{(1-\mu)\mathcal{P}(\sim t_{q,i}|\mathcal{M}_D)}_{(3)} \right]$$

On rappelle que Définition 11 : $\sim t_{q,i} = Approximation(\mathcal{M}_D, t_{q,i})$, c'est-à-dire l'ensemble des approximations de $t_{q,i}$.

La Définition 13 se décompose en trois parties.

3.4.1 Présence

La présence d'un terme correspond à la partie (1) de la Définition 8 du principe de correspondance d'un terme. On définit la probabilité associée au fait que le terme de la requête $t_{q,i}$ apparaît dans le modèle de document $\mathcal{M}_D$ par :

Définition 14 : $$\mathcal{P}(t_{q,i}|\mathcal{M}_D) = \mathcal{P}\left(t_{d,j} \in \mathcal{M}_D \,\middle|\, \begin{array}{c} Conc(t_{q,i}, t_{d,j}) = Concorde \\ ET\ ValInter(Inter(t_{q,i}, t_{d,j})) = 1 \end{array} \right)$$

$\mathcal{P}(t_{q,i}|\mathcal{M}_D) = \mathcal{P}(t_{d,j})$ tel que $t_{q,i} \in Q$, $t_{d,j} \in D$, $t_{q,i} = t_{d,j}$

$$\mathcal{P}(t_{d,j}) = \frac{\sum_{t_{d,j}} Cert(t_{d,j})}{\mathcal{N}_d}$$

On rappelle que $\sum_{t_{d,j}} Cert(t_{d,j})$ correspond à la somme des certitudes associées à chaque occurrence du terme $t_{d,j}$.

3.4.2 Approximations

La prise en compte des approximations d'un terme correspond à la partie (3) de la Définition 8 du principe de correspondance d'un terme.

La Définition 10 de l'approximation caractérise l'ensemble des termes approximant un terme comme les termes à la fois non absents et non présents :

$$\begin{array}{l} Approximation : \quad CI \times \mathcal{V} \rightarrow \mathcal{V}^* \\ (\mathcal{M}_D, \chi) \rightarrow \left\{ t_j \in \mathcal{M}_D \,\middle|\, Absence(\mathcal{M}_D, \chi) = 0 \text{ et } Présence(\mathcal{M}_D, \chi) \neq t_j \right\} \end{array}$$

Ces termes sont définis par leur appariement avec le terme de la requête. La valeur d'appariement dépendant de la valeur de concordance *ValConc* et de la valeur d'intersection *ValInter*, nous utilisons une fréquence relative des approximations pour prendre en compte le fait que le terme $t_{q,i}$ de la requête peut avoir des approximations dans $\mathcal{M}_D$, ainsi :

Définition 15 : $$\mathcal{P}(\sim t_{q,i}|\mathcal{M}_D) = \frac{\sum_{t_{d,j} \in Approximation(t_{q,i})} ValConc(t_{q,i}, t_{d,j}) \times ValInter(t_{q,i}, t_{d,j})}{|D|}$$

Par la suite, nous simplifions l'écriture de $\mathcal{P}(\sim t_{q,i}|\mathcal{M}_D)$, en donnant :

$\alpha = ValConc(t_{q,i}, t_{d,j})$ et $\beta = ValInter(t_{q,i}, t_{d,j})$

ce qui permet d'écrire :

$$\mathcal{P}(\sim t_{q,i}|\mathcal{M}_D) = \frac{\sum_{t_{d,j} \in Approximation(t_{q,i})} \alpha \times \beta}{|D|}$$

3.4.3 Absence

La prise en compte de l'absence d'un terme correspond à la partie (2) de la Définition 8 du principe de correspondance d'un terme.

Bon nombre de fonctions de lissage existent pour les modèles de langue nous choisissons d'utiliser un lissage par interpolation proposé par [Hiemstra98]. Ce lissage tient compte de la probabilité des termes dans le corpus de documents :

Définition 16 : $\mathcal{P}(t_{q,i}|\mathcal{M}_C)$

Nous rappelons que C représente le corpus de documents.

$$\mathcal{P}(t_{q,i}|\mathcal{M}_C)=\frac{df(t_{q,i})}{\sum_{t_j \in \mathcal{VI}} df(t_j)}$$

avec $df(t_{q,i})$ *le nombre de documents de* C *contenant* $t_{q,i}$ *et* $\mathcal{VI}$ *le vocabulaire d'indexation de l'ensemble des documents de* C

3.4.4 Paramètres λ et μ

Le paramètre λ permet de définir l'importance donnée aux approximations des termes par rapport aux termes eux-mêmes. Quant au paramètre μ, il permet de moduler l'importance de la fonction de lissage.

3.5. Fonction de correspondance globale

La fonction de correspondance s'écrit sous la forme suivante :

Définition 17 : $$\mathcal{P}(Q|\mathcal{M}_\mathcal{D})=\prod_{i=1}^{\mathcal{N}_q} Cert(t_{q,i})\times\left[\mu\lambda\,\mathcal{P}(t_{q,i}|\mathcal{M}_\mathcal{D})+\mu(1-\lambda)\mathcal{P}(t_{q,i}|\mathcal{M}_C)+(1-\mu)\mathcal{P}(\sim t_{q,i}|\mathcal{M}_\mathcal{D})\right]$$

Après développement de chacune des fonctions, on obtient :

$$\mathcal{P}(Q|\mathcal{M}_\mathcal{D})=\prod_{i=1}^{\mathcal{N}_q} Cert(t_{q,i})\times\mu\left(\lambda\|t_{q,i}\|+(1-\lambda)\frac{\sum_{t_{d,k}\in Approximation(t_{q,i})}\alpha\times\beta}{|\mathcal{D}|}\right)+(1-\mu)\frac{df(t_{q,i})}{\sum_{t_j\in\mathcal{VI}}df(t_j)}$$

Avec $\alpha=ValConc(t_{q,i},t_{d,j})$ *et* $\beta=ValInter(t_{q,i},t_{d,j})$ et pour tout $t_{q,i}\in Q$, $\exists\ t_{d,j}\in\mathcal{D}$ tel que $t_{d,j}=t_{q,i}$, $\|t_{q,i}\|=\frac{|t_{d,j}|}{\mathcal{N}_\mathcal{D}}$.

3.6. Conclusion

Les données incertaines remettent en cause la base de la fonction de correspondance classiquement utilisée en recherche d'information textuelle qui s'appuie sur une capacité à disposer d'une relation d'égalité entre termes du document et termes de la requête : $t_{d,j} = t_{q,i}$.

La remise en cause de cette égalité a des conséquences sur la fonction de correspondance et sur la représentation des documents, impliquant la mise en place d'un système d'information adapté. Nous proposons (cf. Figure 34) une fonction de correspondance (1) basée sur l'appariement entre termes (2). Cette notion, par le biais de la concordance et de l'intersection, détermine la 'presque égalité' existant entre deux termes. De plus, la valeur de certitude associée à chaque terme extrait s'intègre à la fonction de pondération des documents (3).

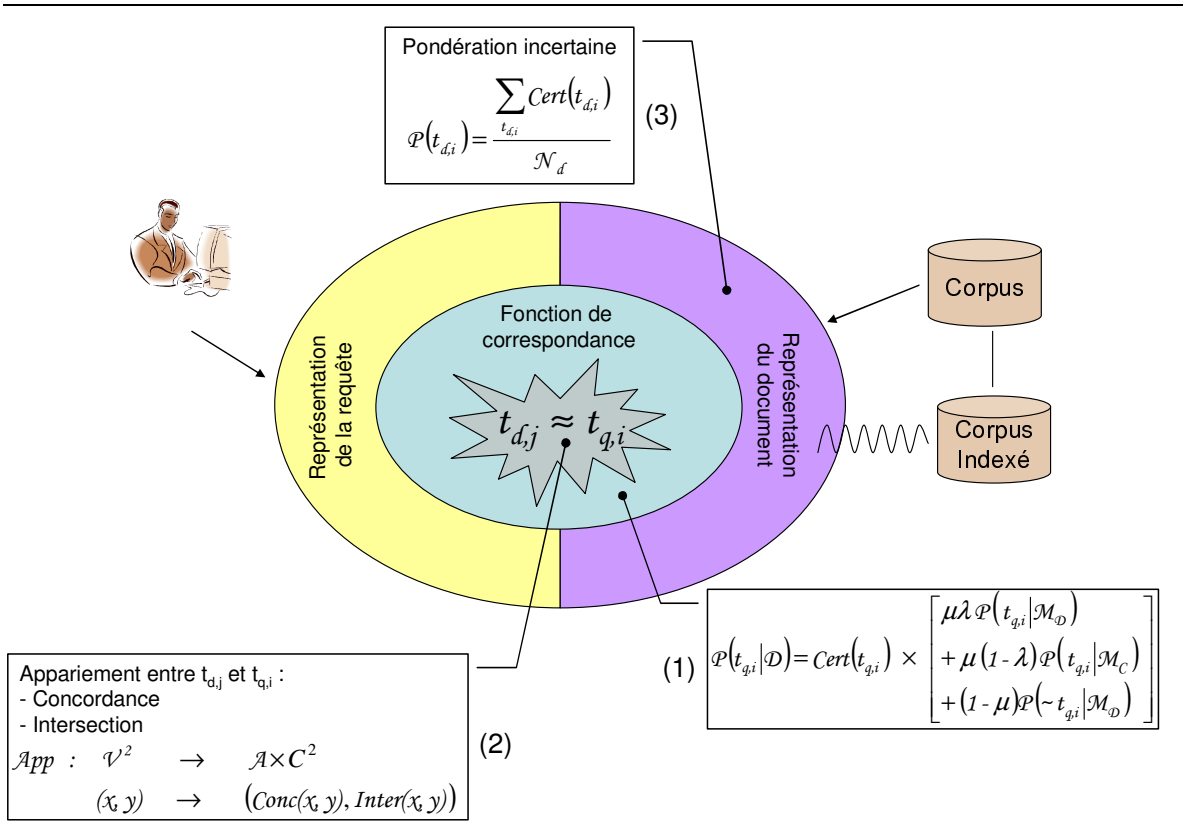


Figure 34. Schéma bilan du modèle de recherche d'information adapté aux données incertaines

4. Une fonction de correspondance générique

La fonction de correspondance ainsi proposée permet de retrouver la fonction de correspondance classiquement utilisée en modèle de langue en recherche d'information, c'est-à-dire n'intégrant pas l'incertitude.

Afin de montrer le parallèle possible entre les deux fonctions de correspondance, nous reprenons chaque point du modèle proposé et l'appliquons à des données certaines.

4.1. Certitude des termes

Avec un processus d'extraction des termes certain, on a :

$Cert : \mathcal{V} \rightarrow \{1\}$, $x \rightarrow 1$ De ce fait : $\forall x \in \mathcal{D},\ c = 1$.

4.2. Appariement entre deux termes

Deux possibilités d'appariement entre un terme x de la requête et un terme y d'un document s'envisagent :

- les termes sont égaux

$Conc(x, y) = concorde$ et $ValInter(Inter(x, y)) = 1$

- les termes sont différents

$Conc(x, y) = ne\_concorde\_pas$

4.3. Pondération d'un terme

La pondération s'exprimant en fonction de la certitude d'un terme, devient une pondération de forme 'classique' dans un contexte certain.

En effet, on otient $\sum_{t_{d,i}} Cert(t_{d,i}) = |t_{d,i}|$ puisque $\forall t_{d,i} \in \mathcal{D}, Cert(t_{d,i}) = 1$

D'où $\mathcal{P}(t_{d,i}) = \frac{\sum_{t_{d,i}} Cert(t_{d,i})}{\mathcal{N}_d} = \frac{|t_{d,i}|}{\mathcal{N}_d}$

On retrouve la pondération classiquement utilisée en recherche d'information avec les modèles de langue.

4.4. Fonction de correspondance

Pour chaque terme du document, deux possibilités existent :

- le terme est présent
- le terme est absent.

Si on considère que l'on est sûr des termes (et donc du processus d'extraction) et que seuls les termes de la requête « exactement » présents dans le document doivent être pris en compte, le cas des approximations de termes n'a plus lieu d'être.

Si on reprend la fonction de correspondance définie précédemment :

$$\mathcal{P}(Q|\mathcal{M}_D) = \prod_{i=1}^{\mathcal{N}_q} Cert(t_{q,i}) \times \left[\mu\lambda \mathcal{P}(t_{q,i}|\mathcal{M}_D) + \mu(1-\lambda)\mathcal{P}(t_{q,i}|\mathcal{M}_C) + (1-\mu)\mathcal{P}(\sim t_{q,i}|\mathcal{M}_D)\right]$$

(1) *(2)* *(3)* *(4)*

(1) $Cert(t_{q,i}) = 1$ puisque les termes de la requête sont certains

(2) Pour $\mathcal{P}(t_{q,i}|\mathcal{M}_D)$, on a $\mathcal{P}(t_{q,i}|\mathcal{M}_D) = \mathcal{P}(t_{d',j})$ *tel que* $\exists t_{d',j} \in \mathcal{M}_D$, $t_{d',j} = t_{q,i}$

et $\mathcal{P}(t_{d',i}) = \dfrac{\sum_{t_{d',i}} Cert(t_{d',i})}{\mathcal{N}_{d'}} = \dfrac{|t_{d',i}|}{\mathcal{N}_{d'}}$.

On retrouve l'estimation de vraisemblance maximale du terme pour évaluer sa probabilité.

(3) Cette partie disparaît puisque les approximations de termes ne sont pas prises en compte : $\mu = 1$

(4) Le lissage demeure pour les mots inconnus (i.e. non présents dans le document)

Compte tenu de ces constatations, la fonction de correspondance devient :

$$\mathcal{P}(Q|\mathcal{M}_D) = \prod_{i=1}^{\mathcal{N}_q} \left(\mu\mathcal{P}(t_{q,i}|\mathcal{M}_D) + (1-\mu)\mathcal{P}(t_{q,i}|\mathcal{M}_C)\right)$$

On retombe sur la fonction de correspondance utilisant les modèles de langue, classiquement utilisée en recherche d'information :

$$\mathcal{P}(Q|\mathcal{M}_D) = \prod_{i=1}^{\mathcal{N}_q} \left(\mu\,\mathcal{P}(t_{q,i}|\mathcal{M}_D) + (1-\mu)\,lissage(t_{q,i})\right)$$

5. Bilan

Nous venons de présenter dans ce chapitre notre système de recherche d'information adapté aux données incertaines. Nous proposons de prendre en compte l'incertitude au sein de la pondération et de la fonction de correspondance.

Le principe de correspondance de notre modèle ne tient pas uniquement compte de la présence du terme de la requête dans le document et de l'absence du terme de la requête dans le document comme le font les modèles de langue par interpolation mais également des approximations du terme de la requête présents dans le document. Ainsi par le biais de l'appariement nous tenons compte du concept de 'presque égalité' entre termes.

Nous proposons donc la fonction de correspondance suivante (Définition 17) :

$$\mathcal{P}(Q|\mathcal{M}_D) = \prod_{i=1}^{\mathcal{N}_q} Cert(t_{q,i}) \times \left[\mu\lambda\,\mathcal{P}(t_{q,i}|\mathcal{M}_D) + \mu(1-\lambda)\mathcal{P}(t_{q,i}|\mathcal{M}_C) + (1-\mu)\mathcal{P}(\sim t_{q,i}|\mathcal{M}_D)\right]$$

Le chapitre suivant présente un exemple d'instanciation de l'appariement au sein du modèle avec des données issues de l'oral.

Chapitre V. Instanciation de l'appariement

Nous venons de décrire notre proposition de modèle de recherche d'information appliqué aux données incertaines. Ce modèle se base sur une fonction de pondération tenant compte de la certitude des termes et sur une fonction de correspondance utilisant l'appariement entre termes par le biais de la concordance et de l'intersection.

Afin de donner vie à ce modèle, nous donnons un exemple de mise en œuvre de l'appariement en proposant une instanciation de celui-ci avec des données issues de l'oral pour une mise en application possible.

Pour ce faire, nous développons l'instanciation de la concordance et de la valeur de la concordance. Nous faisons de même pour l'intersection. Nous montrons que dans le cas de données issues de l'oral, plusieurs appariements peuvent exister pour un même couple de termes, on parle de multi-appariement. Nous finissons ce chapitre par un exemple de détermination de la valeur d'appariement entre deux termes définis.

1. Concordance

On décrit, dans un premier temps, les instanciations de chaque nature de concordance. Ensuite, on définit les valeurs de concordance attribuées à chaque nature de concordance.

1.1. Instanciation de la concordance

Soient deux termes x et y, de longueur respective nx et ny.

Chaque type de concordance est défini en fonction des longueurs nx et ny des termes x et y et/ou des caractères formant les termes x et y.

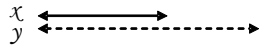
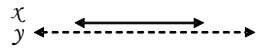
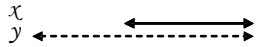
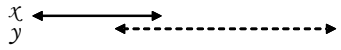
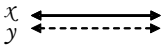
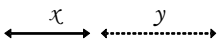
Schéma de relations	Notation de concordance	Instanciation de concordance
	*Conc(x,y) = debute*	*nx < ny et* *1er caractère de x = 1er caractère de y*
	*Conc(x,y) = pendant*	*nx < ny et* *1er caractère de x = ième caractère de y* *avec i≠1 et i≤ nx - ny* *et nxème caractère de x ≠ nyème caractère de y*
	*Conc(x,y) = termine*	*nx < ny et* *nxème caractère de x = nyème caractère de y* *et 1er caractère de x = ième caractère de y* *avec i≠ 1 et i ≤ nx - ny*
	*Conc(x,y) = chevauche*	*1er caractère de y = ième caractère de x* *et nxème caractère de x = jème caractère de y* *tel que i ≠ nx, j ≠ny et nx -1= j-1*
	*Conc(x,y) = concorde*	*nx = ny*
	*Conc(x,y) = ne_concorde_pas*	*Aucune concordance n'a été identifiée*

Figure 35. Instanciation de la concordance

1.2. Instanciation de la valeur de la concordance

On rappelle qu'une valeur de concordance (Définition 5) est définie comme suit :

$$\begin{array}{lccc} ValConc : & \mathcal{A} & \rightarrow & \Re^+ \\ & a & \rightarrow & \alpha \end{array}$$

Les valeurs de concordance α se déterminent en fonction de l'importance attribuée à chaque type de concordance. On établit une relation d'ordre entre les concordances. Ainsi, la concordance *concorde* apparaît comme la concordance la plus « forte » entre deux termes ; les concordances *débute* et *termine* comme équivalentes entre elles ; la concordance *pendant* comme moins importante et *chevauche* encore plus faiblement importante ; enfin, la concordance *ne_concorde_pas* comme nulle. La fonction *ValConc* est une fonction monotone croissante (cf. Figure 36).

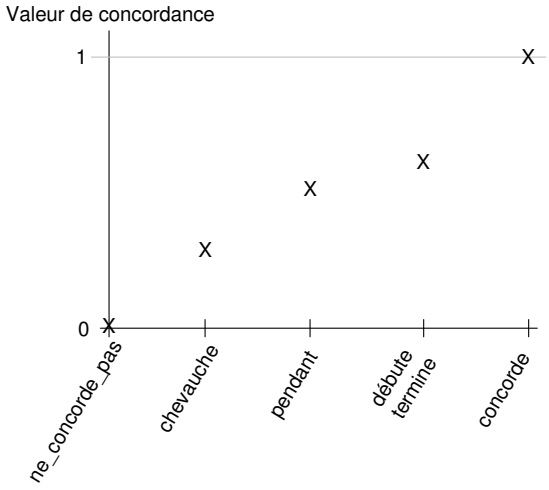


Figure 36. Fonction de la valeur de concordance

La modularité de ces valeurs permet de donner plus ou moins d'importance à une nature de concordance.

Dans le cadre de nos expérimentations, nous utilisons les valeurs suivantes :

$$ValConc(a) = \begin{cases} 0,8 \quad si \quad a = débute \\ 0,6 \quad si \quad a = pendant \\ 0,8 \quad si \quad a = termine \\ 0,2 \quad si \quad a = chevauche \\ 1 \quad si \quad a = concorde \\ 0 \quad si \quad a = ne\_concorde\_pas \end{cases}$$

2. Intersection

L'intersection correspond à la zone partagée par les deux mots au niveau de leur concordance.

Prenons deux exemples :

- Soient les mots x = « bonjour » et y = « journée »

On constate que ces deux mots se *chevauchent* (nature de leur concordance) ainsi : bon***jour*** et ***jour***née.

Leur zone commune est respectivement zx = *jour* et zy = *jour*. Leur intersection s'avère 'parfaite' puisque *jour* = *jour*, c'est-à-dire zx = zy.

- Soient les mots x = « talonneur » et y = « honneur »

Le mot « honneur » termine (nature de leur concordance) le mot « talonneur » ainsi : ***honneur*** et tal***onneur***.

Leur zone commune correspond respectivement à x = *honneur* et y = *onneur*. Dans ce cas là, *honneur* ≠ *onneur*, c'est-à-dire $zx \neq zy$. Toutefois, ces deux zones (ou groupes de caractères) se prononcent de façon identique : *oneur*.

Leurs intersections sont phonétiquement égales.

Nous devons définir un algorithme permettant de définir la valeur de l'intersection entre deux mots.

Nous décrivons le principe des algorithmes phonétiques nécessaires à la définition des valeurs de l'intersection entre deux termes.

2.1. Les algorithmes phonétiques

2.1.1 Historique et principe général

a. Historique

Les algorithmes phonétiques les plus connus correspondent aux algorithmes de type Soundex. Le Soundex correspond au premier algorithme de ce type inventé par Margaret O'Dell et Robert C. Russell en 1918. Depuis, ce terme regroupe une famille d'algorithmes de codage phonétique.

b. Principe général

Le principe de base des algorithmes phonétiques de type Soundex consiste à codifier les mots en éliminant les lettres en doubles, les lettres muettes (telles que H) et en effectuant un rapprochement entre lettres ayant le même son. Cette codification permet une comparaison phonétique entre deux termes. Les algorithmes phonétiques se basent donc sur la consonance et non sur les termes eux-mêmes. Ils sont beaucoup utilisés dans les bases de données afin de palier les problèmes de fautes d'orthographe ou mauvaise écriture de noms. Ainsi, « DUPOND » et « DUPONT » sont considérés comme identiques.

Ces algorithmes ont été tout d'abord développés pour la langue anglaise. Leur extension à la langue française apparaît dans les années 1990, notamment avec l'algorithme de Frédéric Brouard [Brouard, 1999].

Une description des algorithmes phonétiques les plus fréquemment utilisés, à savoir, Soundex, Soundex2 et Phonex, se trouve en annexe 2 de ce manuscrit.

2.1.2 Notre choix d'algorithme phonétique

Nous définissons une adaptation d'un des algorithmes phonétiques précédemment cités.

Les algorithmes phonétiques permettent de déterminer si deux termes sont phonétiquement identiques. L'utilisation que nous faisons des algorithmes phonétiques s'avère différente puisque les termes dont nous disposons sont incertains et nous cherchons comment les termes auraient pu être retranscrits.

Nous avons choisi d'adapter l'algorithme du Soundex2.

Exemples de codage Soundex de termes :

bonjour -> BNJR

journée -> JRN

Le codage phonétique des termes 'bonjour' et 'journée' met en avant que les deux termes contiennent la chaîne phonétique 'JR', en l'occurrence 'jour'.

Dupont -> DPND

Dupond -> DPNT

Ce deuxième exemple montre que les deux chaînes de caractères ont un partie commune 'DPN', à savoir 'dupon'.

2.2. Instanciation de l'intersection

2.2.1 Introduction

La valeur de l'intersection entre deux termes se détermine par la comparaison des codages phonétiques des deux parties de termes correspondant à la zone d'intersection. Plus les codages phonétiques sont semblables puis la valeur de leur intersection s'avère élevée.

Les codages phonétiques se font sur 4 caractères, aussi est-il possible d'avoir des caractères « 0 » pour compléter un codage plus court.

2.2.2 Notations

Soient deux termes x et y partageant une intersection dont les zones d'intersection sont respectivement zx et zy. On définit cx et cy comme le codage phonétique de chacune de ces zones d'intersection sur 4 caractères (Figure 37). Dans le cas de l'intersection, seules les zones d'intersection et leur codage phonétique ont une importance, les termes en eux-mêmes n'interviennent pas.

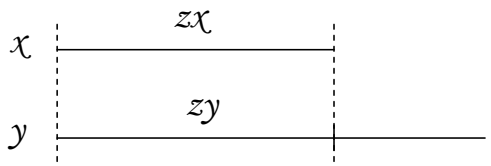


Figure 37. Deux termes x et y de partageant une zone commune respectivement zx et zy

Egalité de deux zones d'intersection :

Deux zones de termes zx et zy sont égales si $zx = zy$.

Egalité phonétique de deux zones d'intersection :

Deux zones de termes zx et zy sont phonétiquement égales si $cx = cy$.

Nombre de caractères communs à deux zones d'intersection :

Le nombre de caractères communs aux zones zx et zy est défini par $nbCC(zx, zy)$.

Taille d'une zone d'intersection

La taille d'une zone d'intersection zx est définie par $|zx|$ tel que :

$$|zx| = \begin{cases} \text{nombre de caractères de } zx & \text{si nombre de caractères} \leq 4 \\ 0 & \text{sinon} \end{cases}$$

Taille du codage phonétique d'une zone d'intersection

La taille du codage phonétique cx d'une zone d'intersection zx est définie par $|cx|$ tel que :

$|cx|$ = *nombre de caractères non nuls, c'est-à-dire* $\neq 0$.

2.2.3 Mise en œuvre de l'instanciation de l'intersection

L'affectation des différentes valeurs de $ValInter(x, y)$ peut s'exprimer à l'aide des notations définies précédemment et de la distance de Hamming [Hamming, 1950].

a. Définition de la distance de Hamming

On rappelle que la distance de Hamming, définie par Richard Hamming, s'utilise en informatique, en traitement du signal et dans les télécommunications. Elle permet de quantifier la différence entre deux séquences de symboles.

La distance de Hamming est une distance au sens mathématique du terme. À deux suites de symboles de même longueur, elle associe l'entier désignant le cardinal de l'ensemble des symboles de la première suite qui différent de la deuxième.

Le poids de Hamming correspond au nombre d'éléments différents de zéro dans une chaîne d'éléments d'un corps fini. On définit la distance de Hamming entre deux éléments a et b par la fonction *d(a, b)*.

Formellement, $\forall a,b \in F \quad a=(a_i)_{i\in[0,n-1]} \text{ et } b=(b_i)_{i\in[0,n-1]}, \quad d(a,b)=\#\{i : a_i \neq b_i\}$

avec *A* un alphabet et *F* l'ensemble des suites de longueur *n* à valeur dans *A*. La notation #*E* désigne le cardinal de l'ensemble *E*.

Nous décrivons plus en détail cette distance en annexe 3.

b. Exemple de distance de Hamming

Soient deux suites binaires suivantes :

a = (0 0 0 1 1 1 1) et *b* = (1 1 0 1 0 1 1)

alors *d(a,b)* = 1 + 1 + 0 + 0 + 1 + 0 + 0 = 3

la distance de Hamming entre *a* et *b* est égale à 3.

c. Instanciation de la valeur de l'intersection avec la distance de Hamming

On peut donc définir *ValInter(x, y)* ainsi :

$$ValInter(x,y)=\begin{cases} 0 & si \quad d(cx,cy)=4 \\ 0,1 & si \quad d(cx,cy)=3 \\ 0,2 & si \quad d(cx,cy)=2 \\ 0,3 & si \quad d(cx,cy)=1 \\ 0,2\times|cx| & si \quad d(cx,cy)=0 \ \ avec\ |cx| = nombre\ de\ caractères\ de\ codage\ non\ nuls \\ 0,25\times|x| & si \quad x=y \ \ avec\ |x| = nombre\ de\ caractères\ de\ x \quad si\ nombre\ de\ caractères \leq 4 \\ & \qquad\qquad\qquad\quad = 4 \qquad\qquad\qquad\quad sinon \end{cases}$$

2.3. Illustration de l'instanciation de l'intersection

Nous prenons un exemple afin d'illustrer l'instanciation de l'intersection.

Soient les termes x = monde et y = mandat

Soient *nx* et *ny*, leur longueur respective.

Le terme *x débute* le terme *y* puisque :

- $nx < ny$
- 1er caractère de *x* = 1er caractère de *y*

Leurs zones communes, respectivement *zx* et *zy* (cf. Figure 38), valent :

- *zx* = 'monde'
- *zy* = 'manda'

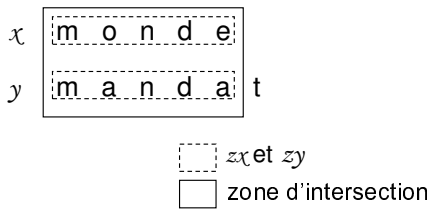


Figure 38. Schématisation de la zone d'intersection entre les termes 'monde' et mandat'

Les codages phonétiques de *zx* et *zy* correspondent respectivement à :

- *cx* = MND0
- *cy* = MND0

Pour déterminer la valeur de l'intersection entre *x* = monde et *y* = mandat, on procède comme suit :

- test de l'égalité entre les zones d'intersection *zx* et *zy* :
- $zx \neq zy$ (monde ≠ manda)
- test de l'égalité entre les codages phonétiques des zones d'intersection *cx* et *cy* :
- *cx* = *cy* (MND0 = MND0)
- détermination de la taille de *cx*
- |*cx*| = 3
- détermination de la valeur de l'intersection entre les 2 termes *ValInter(zx, zy)* :
- *ValInter(zx, zy)* = 0,2 * 3 = 0,6

3. Appariement

3.1. Définition de l'appariement

3.1.1 Rappel de l'appariement

L'appariement entre deux termes x et y se compose du couple concordance et intersection entre ces deux termes.

$$\begin{array}{lccl} App : & \mathcal{V}^2 & \rightarrow & \mathcal{A} \times C^2 \\ & (x, y) & \rightarrow & (Conc(x, y), Inter(x, y)) \end{array}$$

La valeur associée à l'appariement se définit ainsi :

$$\begin{array}{lccl} ValApp : & \mathcal{A} \times C^2 & \rightarrow & [0,1] \\ & (\alpha, (x', y')) & \rightarrow & (ValConc(x, y), ValInter(x', y')) \end{array}$$

3.1.2 Multi-appariement

Deux termes peuvent être liés entre eux par plusieurs concordances. Dans ce cas, l'appariement choisi entre les deux termes correspond à celui ayant la plus grande valeur *ValApp*, c'est-à-dire la plus grande valeur *ValConc* * *ValInter*.

Ainsi, pour tout couple de termes x et y, on a un ensemble de n appariements possibles :

$\{App(x, y)_n\}$

Nous définissions l'appariement entre deux termes x et y comme l'appariement maximum de l'ensemble des n appariements possibles :

$$App(x, y) = \max_n \{App_n(x, y)\}.$$

3.2. Illustration du multi-appariement

Nous détaillons un exemple de détermination de l'appariement entre deux termes x et y parmi les appariements possibles.

Soient les termes x = maman et y = ma

y débute x ***ma***man mais également *y pendant x* ma***ma***n.

1er cas : y débute x

Conc(x, y) = débute

ValConc(x, y) = 0,8

Inter(x, y) = ('ma', 'ma')

*ValInter('ma', 'ma') = 0,25 * 2 = 0,5*

D'où $ValApp_{débute}(x, y) = 0,8 * 0,5 = 0,4$

2ème cas : y pendant x

Conc(x, y) = pendant

ValConc(x, y) = 0,6

Inter(x, y) = ('ma', 'ma')

*ValInter('ma', 'ma') = 0,25 * 2 = 0,5*

D'où $ValApp_{pendant}(x, y) = 0,6 * 0,5 = 0,3$

Conclusion :

On a pour *x* et *y* l'ensemble des appariements $\{App_n\} = \{App_{pendant}, App_{débute}\}$ et $App_{débute}(x, y) > App_{pendant}(x, y)$, l'appariement entre les termes *x* et *y* correspond donc à $App(x, y) = App_{débute}(x, y) = 0,4$.

4. Conclusion

Ce chapitre vient de détailler une instanciation de l'appariement nécessaire à notre modèle de recherche d'information adapté aux données incertaines. Pour cela, nous revenons sur les différents composants de l'appariement : la concordance et l'intersection. L'instanciation de l'appariement présentée correspond à celle utilisée pour les expérimentations. Nous avons montré que la fonction *ValConc(Conc (x, y))* associant une valeur à chaque type de concordance *Conc(x, y)* est croissante. Il en est de même pour la fonction *ValInter(Inter(x, y))* qui attribue une valeur en fonction de l'intersection *Inter(x, y)*. Enfin, la fonction *ValApp(Conc(x, y), Inter(x, y))* définit la valeur de l'appariement entre deux termes *x* et *y*, elle se base sur la valeur maximale d'appariement entre les termes.

Cette instanciation possible du modèle permet la mise en place du protocole expérimental décrit dans la partie suivante.

Partie 3 : Validation du modèle

Tout système de recherche d'information repose sur un certain nombre d'hypothèses basé sur la distribution et le pouvoir d'expression des termes au sein des documents et corpus.

Dans nos expérimentations, nous utilisons des transcriptions de l'oral, cadre d'étude dans lequel l'incertitude est omniprésente. Avant toute chose, on se pose la question de la possibilité de faire de la recherche d'information avec des données issues de transcriptions de l'oral. Pour répondre à cette interrogation, dans le chapitre III, nous effectuons une étude de l'applicabilité des hypothèses de la recherche d'information aux données incertaines en vérifiant que les données orales respectent la loi de Zipf et la conjecture de Luhn.

La suite de cette partie 3 Partie 3 : décrit les expérimentations effectuées pour valider notre proposition de système de recherche d'information adapté aux donnes incertaines. Notre système propose :

- Une représentation des documents et requêtes basée sur une pondération intégrant la valeur de certitude associée au processus d'extraction des termes.
- Une fonction de correspondance fondée sur l'appariement entre deux termes, l'appariement permettant d'évaluer la 'presque égalité' entre deux termes par le biais de la concordance et de l'intersection.

Afin de pouvoir tester notre modèle, nous devons disposer d'un corpus formé de cinq composants, à savoir des documents audio, les transcriptions manuelles et automatiques correspondant aux documents audio, des valeurs de certitude associées à chaque terme extrait et des requêtes résolues.

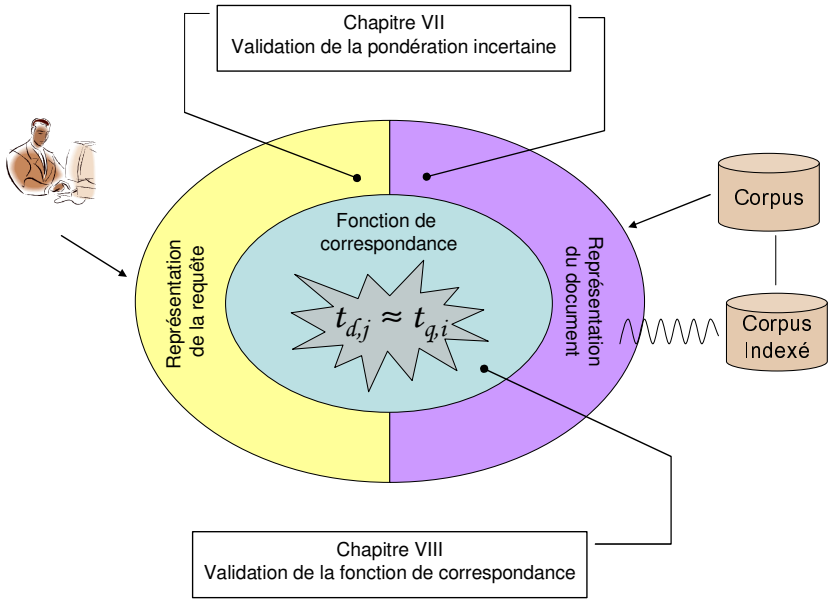


Figure 39. Validation du système d'information adapté aux données incertaines

Nous disposons de deux corpus : données de CLEF-2004 et données de la campagne ESTER. Le corpus de CLEF-2004 se compose de documents avec des valeurs de certitude associées à chaque terme du document ainsi que de requêtes résolues. Ce corpus permet de

tester l'effet de la prise en compte des valeurs de certitude dans la représentation des documents sur les performances d'un système de recherche de type vectoriel.

Le corpus de la campagne ESTER se compose de documents audio ainsi que les transcriptions automatiques et manuelles de ces documents audio. L'évaluation de l'impact de la prise en compte de l'incertitude par le biais de la 'presque égalité' et de l'appariement dans la fonction de correspondance sur les performances de recherche d'un système de recherche d'information de type modèle de langue.

De ce fait, n'ayant pas à disposition un corpus complet, c'est-à-dire composé de documents audio, de transcriptions automatiques et manuelles correspondantes et de requêtes résolues ; nos expérimentations se divisent en deux parties : premièrement, une validation de la fonction de pondération en utilisant un corpus de données issues de la campagne CLEF-2004, ensuite une validation de la fonction de correspondance avec un corpus provenant de la campagne ESTER.

Ainsi comme le montre la Figure 39, le chapitre VII présente les expérimentations de la fonction de pondération et le chapitre VIII celles de la fonction de correspondance.

Ces deux chapitres décrivent les corpus utilisés, les protocoles expérimentaux, les résultats et dressent un bilan des expérimentations.

Chapitre VI. Une étude préalable à la recherche d'informations orales

Nous effectuons une étude préalable à la validation d'un modèle de recherche d'information adapté à l'oral afin d'évaluer le bien fondé de l'utilisation des méthodes classiquement utilisées en recherche d'information textuelle pour des données orales.

En effet, en recherche d'information, des hypothèses sur la distribution des termes dans un corpus de documents textuels servent de base à la mise en place d'un système de recherche d'information. Ces hypothèses, établies sur des corpus de documents textuels, s'avèrent nécessaires pour l'extraction des mots pour la représentation des documents sous forme de documents indexés.

Dans ce chapitre, nous revisitons certaines de ces hypothèses de la recherche d'information en étudiant la spécificité et l'applicabilité de ces hypothèses à un corpus de documents issus de transcriptions de l'oral. Ces hypothèses portant sur la distributivité et le pouvoir expressif des mots dans un document reposent sur la loi de Zipf et la conjecture de Luhn. Ce chapitre (cf. Figure 40) commence par la présentation de ces principes (1), puis se poursuit par la description des corpus (2) utilisés pour la vérification de l'applicabilité de celles-ci. Nous choisissons, pour notre étude, deux corpus de transcriptions de l'oral : des transcriptions de conversations téléphoniques et des transcriptions d'émissions radiophoniques. Enfin, nous montrons l'applicabilité des hypothèses présentées (3) sur ces données incertaines.

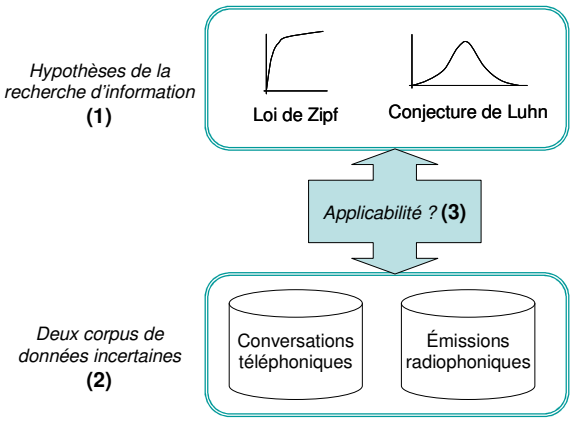


Figure 40. Plan du chapitre

1. Rappel des hypothèses de la recherche d'information

1.1. Loi de Zipf

Le vocabulaire des documents classiquement utilisés en recherche d'information suit la loi de Zipf. Le classement par ordre de fréquence décroissante de l'ensemble des mots différents d'un texte quelconque montre que la fréquence d'un mot est inversement proportionnelle à son rang dans la liste. Autrement dit, le produit de la fréquence de n'importe quel mot par son rang reste constant, ce que traduit la formule :

*Rang du terme * (fréquence du terme / nombre de termes) = constante*

Ainsi, l'analyse statistique des documents textuels en anglais montre que les mots les 20% les plus fréquents représentent 70% du vocabulaire des documents écrits [Salton, 1975] (cf. *Figure 41*).

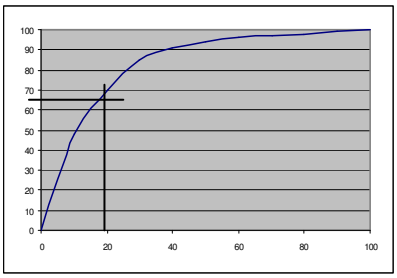


Figure 41. Usage des mots dans les documents de recherche d'information

1.2. Conjecture de Luhn

En recherche d'information, la conjecture de Luhn [Luhn, 1957] indique le pouvoir d'expression des mots dans un texte en fonction de leur fréquence. Ceci se représente par la courbe de la Figure 42.

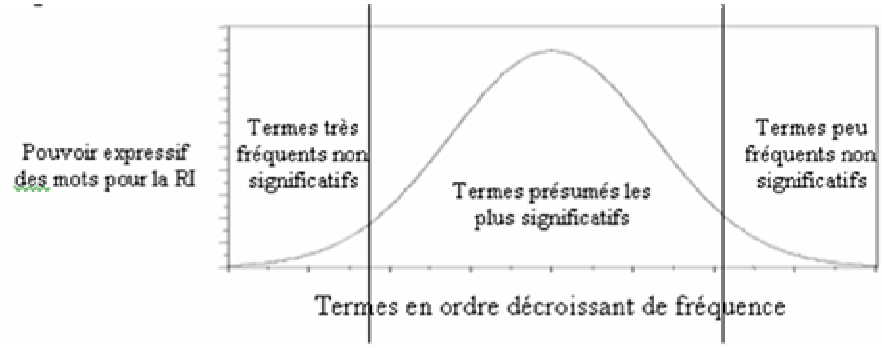


Figure 42. Conjecture de Luhn : pouvoir d'expression des mots

Sur cette courbe, en abscisse, les mots sont ordonnés du plus fréquent au moins fréquent. En ordonnée, cette courbe montre que les extrêmes (mots peu ou trop utilisés) offrent un pouvoir expressif limité, contrairement aux mots d'utilisation moyenne. Le pouvoir expressif des mots se base ici sur le calcul de leur fréquence.

2. Description des corpus

Pour notre étude, nous avons choisi deux corpus de transcriptions de l'oral : les conversations téléphoniques correspondant à du discours spontané et les émissions radiophoniques étant un discours moins spontané puisque certains orateurs lisent un texte.

2.1. Le corpus 'conversation téléphoniques'

2.1.1 Construction du corpus

La construction d'un tel corpus de conversations, grâce à une collaboration avec l'équipe GEOD (Groupe d'Etude sur l'Oral et le Dialogue) (GEOD) du CLIPS et la société CALISTEL (CALISTEL), se fait en deux temps.

Dans un premier temps, les conversations audio ayant eu lieu lors de réunions téléphoniques sont recueillies. Dans un second temps, un professionnel des sciences du langage retranscrit ces conversations afin de constituer notre corpus de conversations retranscrites. Les transcriptions comprennent beaucoup d'annotations : ton, hésitations, confusions, répétitions, etc.… (Voir annexe 5). Notre étude ne porte pas sur le signal de la conversation mais bien sur une transcription « ASCII » de la conversation. On élude volontairement la problématique liée au traitement du signal dans une première approche.

2.1.2 Description du corpus audio

Le corpus dont nous disposons comporte 13 transcriptions manuelles de réunions téléphoniques. Chaque réunion dure en moyenne 45 minutes. Trois grandes catégories regroupent ces conversations : réunion de projet, brainstorming et entretien d'embauche.

Chacun des trois types de réunions a ses particularités que nous décrivons ci-dessous.

La réunion de projet se déroule en général en début et tout au long de la mise en œuvre d'un projet. Cette réunion regroupe en général un responsable de la réunion et un cercle d'intervenants qui sont tour à tour actifs ou passifs. Généralement, ce type de réunion débute par un tour de table où chaque participant fait le point sur l'état d'avancement personnel au sein du projet et expose parfois ses soucis. Ensuite, l'équipe débat quelques points et enfin fait le point sur les futures tâches à accomplir.

La réunion de type brainstorming a généralement pour but la récolte de nombreuses idées en un minimum de temps. On pourrait assimiler la réunion de type brainstorming à un outil de « résolution de problème ». Dans ce type de réunion, tous les intervenants participent en même temps, chacun étant sur un pied d'égalité vis-à-vis des autres participants.

Enfin, la réunion de type entretien ne correspond pas seulement à l'entretien d'embauche mais à tout type d'entretien. Dans ce type de réunion, on trouve un jury (composé au minimum d'une personne) face à la personne en entretien. Généralement, au sein du jury se trouve un président de jury assimilé au leader du groupe. Souvent, celui-ci dirige l'entretien. Ensuite, plusieurs hypothèses possibles : soit seul le président du jury pose les questions et les autres membres du jury (dans le cas d'un jury composé de n personnes) écoutent et se contentent de juger sans poser de questions, soit chaque membre du jury participe à tour de rôle en posant des questions au candidat.

Pour chaque transcription de réunions, nous effectuons un comptage du nombre de mots, du nombre de tours de parole, du nombre de mots par tour de parole. Le détail de ces statistiques se trouve en annexe 5.

2.2. Le corpus 'émissions radiophoniques'

Ce corpus issu de la campagne d'évaluation ESTER[11] en 2005, se compose de documents issus de transcriptions d'émissions radiophoniques.

Les journaux radiophoniques ont été émis entre les années 1998 et 2000 sur les radios « France Inter » et « RFI ». Ces journaux durent 20 min ou une heure. La durée totale des enregistrements de 9h20min se répartit ainsi (cf. Tableau XIV) :

[11] Evaluation des Systèmes de Transcription Enrichie d'Émissions Radiophoniques

Corpus	Durée	Nombre de documents	Durée moyenne des documents
France Inter Journaux de 1h	4h	85	3 min
France Inter Journaux de 20min	1h20min	25	3 min
RFI Journaux de 1h	4h	122	2 min
TOTAL	9h20min	231	

Tableau IX. Description du corpus ESTER

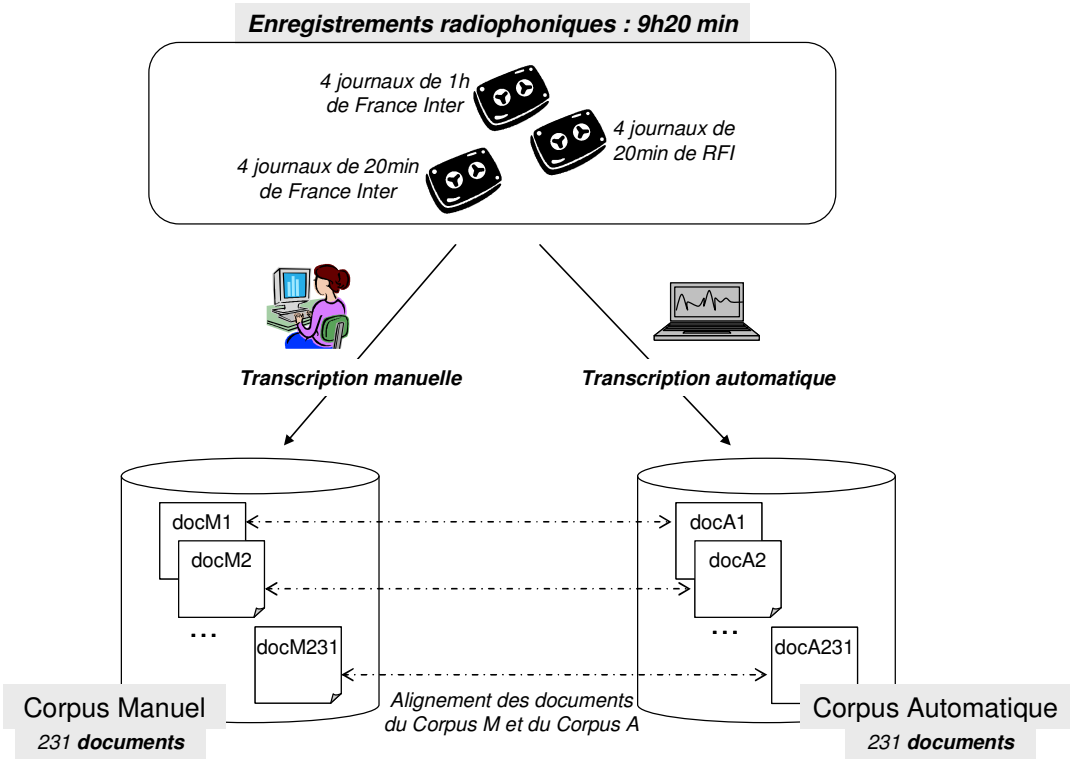


Figure 43. Description du corpus ESTER

Ces 9h 20min d'enregistrements correspondent à 231 documents. Pour chaque document, on dispose de la transcription manuelle et de la transcription automatique faite par un système de reconnaissance automatique de la parole (cf. Figure 55).

Ce traitement est effectué par le système de reconnaissance de la parole de l'équipe GEOD[12]. Le taux d'erreurs de reconnaissance est de l'ordre de 30-35%. Trois types d'erreurs apparaissent :

- Mots mal reconnus (par exemple '*élection'* au lieu de '*connexion'*)
- Mots oubliés (i.e. non retranscrits)
- Mots ajoutés (i.e. insertion de mots non présents dans l'enregistrement audio original).
-

Par souci de simplification, dans la suite du manuscrit nous appellerons :

- ***corpus manuel*** : le corpus composé des documents transcrits manuellement
- ***corpus automatique*** : le corpus composé des documents transcrits automatiquement par le système de reconnaissance de la parole
- ***corpus*** : le corpus manuel + le corpus automatique.

3. Vérification de l'applicabilité des hypothèses aux corpus incertains

3.1. Le corpus 'conversations téléphoniques'

3.1.1 Applicabilité de la loi de Zipf

a. Fréquences des termes du corpus

Dans un premier temps, nous établissons la fréquence des termes les plus courants dans chaque conversation. Ensuite, nous effectuons le coefficient R * F/N, où R est le rang du terme, F la fréquence du terme et N le nombre total des termes.

Les résultats obtenus pour notre corpus de conversations montre que l'on retrouve une constante pour le coefficient R*F/N (cf. Figure 44). Cette constante est de l'ordre de 0,1 (moyenne des valeurs).

[12] http://www-clips.imag.fr/geod/

Brainstorming				Entretien				Réunion projet			
Rang	Terme	F	R * (F/N)	Rang	Terme	F	R * (F/N)	Rang	Terme	F	R * (F/N)
111	projets	24	0,120	151	professionnel	4	0,143	137	coup	22	0,135
112	après	24	0,121	152	prochaine	4	0,144	138	chose	22	0,136
113	veux	23	0,117	153	poser	4	0,145	139	vais	21	0,130
114	sont	23	0,118	154	parfait	4	0,146	140	clair	21	0,131
115	Idée	23	0,119	155	oh	4	0,147	141	avoir	21	0,132
,,,											
849	membres	2	0,077	246	transcription	2	0,117	921	partout	2	0,082
850	mélanger	2	0,077	247	toute	2	0,117	922	parties	2	0,082
851	meilleure	2	0,077	248	toujours	2	0,118	923	particulier	2	0,082
852	maximum	2	0,077	249	tombe	2	0,118	924	pars	2	0,083
853	marrant	2	0,077	250	terme	2	0,119	925	parle	2	0,083
,,,											
1202	action	1	0,054	407	ambiance	1	0,097	1 265	accepter	1	0,056
1203	accueillent	1	0,054	408	aller	1	0,097	1 266	accélérer	1	0,057
1204	accessible	1	0,054	409	aider	1	0,097	1 267	accéder	1	0,057
1205	abstraction	1	0,054	410	agenda	1	0,097	1 268	absente	1	0,057
1206	abord	1	0,054	411	activités	1	0,098	1 269	aborder	1	0,057

Figure 44. Illustration de la loi rang-fréquence pour le corpus des conversations

b. Analyse qualitative de la loi de Zipf pour les documents « conversation »

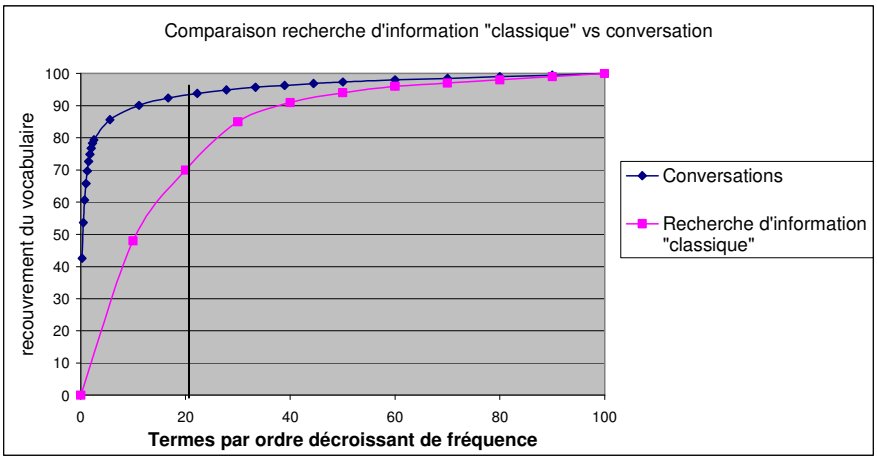


Figure 45. Statistiques sur l'usage des mots dans l'ensemble des conversations de notre corpus

La Figure 45 de la superposition de la courbe sur l'usage des mots dans les documents « conversations » avec celle de l'usage des mots dans les documents « classiques » de recherche d'information montre que les deux courbes suivent la même tendance. Cette figure fait ressortir le caractère encore plus répétitif du vocabulaire de la conversation.

L'étude des statistiques effectuées sur notre corpus de conversations montre que les mots les 10% les plus fréquents couvrent entre 70% et 80% du vocabulaire des conversations. La même tendance s'observe pour les statistiques faites sur l'ensemble des conversations avec un taux de couverture du vocabulaire de près de 90%. Ceci montre le caractère très répétitif du vocabulaire des conversations.

Nous constatons (cf. Tableau X) également que les transcriptions des réunions se rapprochant le plus de l'écrit correspondent au type entretien d'embauche.

Type de conversations	Taux de couverture du vocabulaire par les x% de mots les plus fréquents	
	10%	20%
Ecrit	≅ 50%	≅ 70%
Entretien d'embauche	≅ 70%	≅ 80%
Réunion de projet	≅ 80%	≅ 85%
Brainstorming	≅ 80%	≅ 90%
Toutes les conversations	≅ 90%	≅ 95%

↑ + proche de l'écrit

- proche de l'écrit

Tableau X. Taux de couverture du vocabulaire pour les différents types de conversation

c. Conclusion

Le vocabulaire des documents « conversation » respecte la loi de Zipf et permet d'établir une courbe sur la statistique d'usage des mots similaire à celle connue pour les documents textuels. Nous venons donc d'établir l'applicabilité et l'évaluation qualitative de la loi de Zipf sur les documents de type « conversation ».

3.1.2 Applicabilité de la conjecture de Luhn

a. Critère de sélection des termes porteurs de sens

Parmi les systèmes indexant des documents textuels en français, un certain nombre (IOTA [Bruandet, 1997], …) utilisent des indexations non pas basées sur des statistiques mais sur une analyse de la langue naturelle. En ce qui concerne l'analyse morpho-syntaxique du langage, on distingue deux grandes familles de mots :

- une famille regroupant les adjectifs qualificatifs, les substantifs propres, les substantifs communs et les verbes (groupe 1 sur la Figure 46)

– une autre regroupant le reste du vocabulaire (groupe 2 sur la Figure 46).

Le vocabulaire porteur de sens pour la recherche d'information se situe dans le groupe 1, comme le montre la Figure 46 [Palmer, 1990].

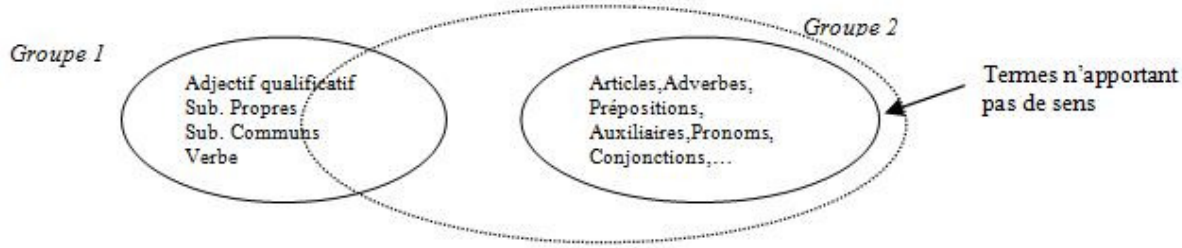


Figure 46. Catégories de vocabulaire

Nous étudions de façon qualitative le vocabulaire utilisé dans les documents « conversation ».

L'étude montre que les substantifs, termes porteurs de sens, se situent entre deux seuils déterminables (cf. Figure 47 et Figure 48) permettant de déduire l'applicabilité de la conjecture de Luhn.

En abscisse de la Figure 47, nous trouvons les termes par ordre décroissant de fréquence dans l'ensemble des conversations et en ordonnée, le nombre de termes de chaque catégorie.

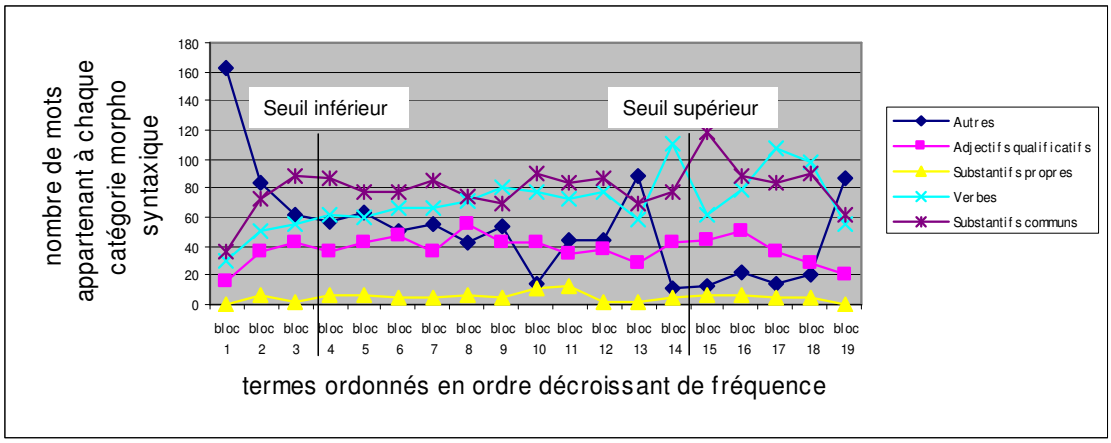


Figure 47. Fréquence des différentes catégories de mots par portion de 250 termes

Entre ces deux valeurs seuils, nous voyons que la proportion de mots les plus fréquents appartiennent à la catégorie morpho-syntaxique des substantifs communs.

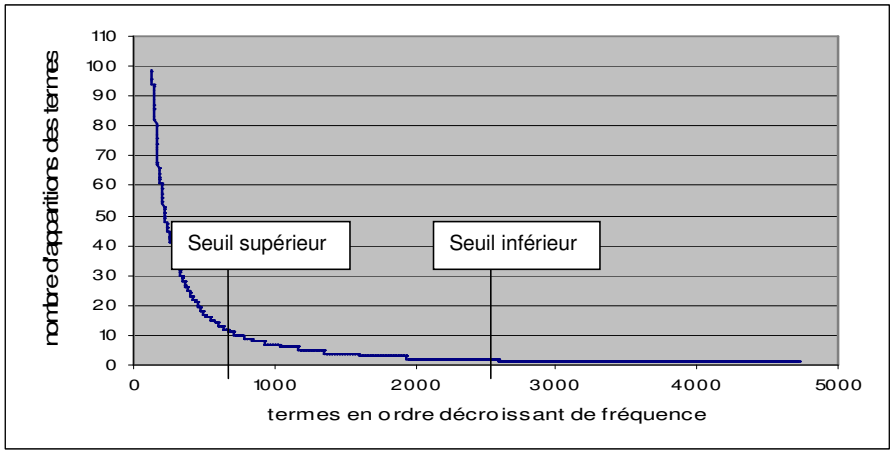


Figure 48. Zoom sur le nombre d'apparitions des termes par ordre décroissant de fréquence

De façon plus précise, la Figure 48 nous permet de déterminer un seuil inférieur à 2 apparitions et un seuil supérieur à 10 apparitions d'un même mot. Ces données confirment l'applicabilité de la conjecture de Luhn au corpus 'conversations' en indiquant que seuls les termes moyennement fréquents restent porteurs de sens.

3.2. Le corpus 'émissions radiophoniques'

3.2.1 Applicabilité de la loi de Zipf

a. Fréquences des termes du corpus

Tout comme pour le corpus 'conversations', nous procédons à un comptage de la fréquence des termes du corpus 'émissions radiophoniques' afin de vérifier l'applicabilité de la loi de Zipf. L'étude a été effectuée sur le corpus manuel (cf. Figure 49).

On retrouve une constante de l'ordre de 1 pour le corpus manuel.

Taille du corpus : N = 12 596			
Rang R	Terme	Fréquence F	R * F / N
1	de	4594	0,365
2	la	2689	0,427
3	le	2348	0,559
4	les	2108	0,669
5	à	1971	0,782
	…		
500	trouver	21	0,834
501	finalement	21	0,835
502	savez	20	0,797
503	presse	20	0,799
504	seront	20	0,800
	…		
5000	s'intéressent	2	0,794
5001	enseigne	2	0,794
5002	recommandations	2	0,794
5003	compromis	2	0,794
5004	amené	2	0,795
	…		
12592	couplage	1	1,000
12593	qu'in	1	1,000
12594	matins	1	1,000
12595	incontournable	1	1,000
12596	subsisté	1	1,000

Figure 49. Illustration de la loi rang-fréquence pour le vocabulaire du corpus manuel

b. Analyse de la loi de Zipf pour le corpus 'émissions radiophoniques'

La loi de Zipf permet d'obtenir un nuage de point relativement linéaire (cf. Figure 50) en traçant pour chaque mot le couple rang/effectif dans un repère logarithmique.

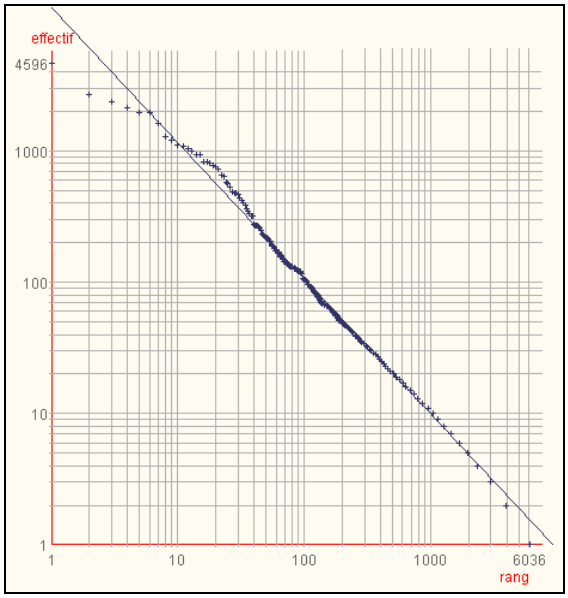


Figure 50. Représentation dans un repère logarithmique des couples rang/fréquence pour chaque terme

3.2.2 Applicabilité de la conjecture de Luhn

En recherche d'information, tous les termes ne sont pas considérés comme pertinents. En effet, les mots trop fréquents et trop peu fréquents sont considérés comme faiblement porteurs de sens. Seuls les mots moyennement fréquents sont discriminants au sein d'une fonction de correspondance (zone marquée sur la Figure 51).

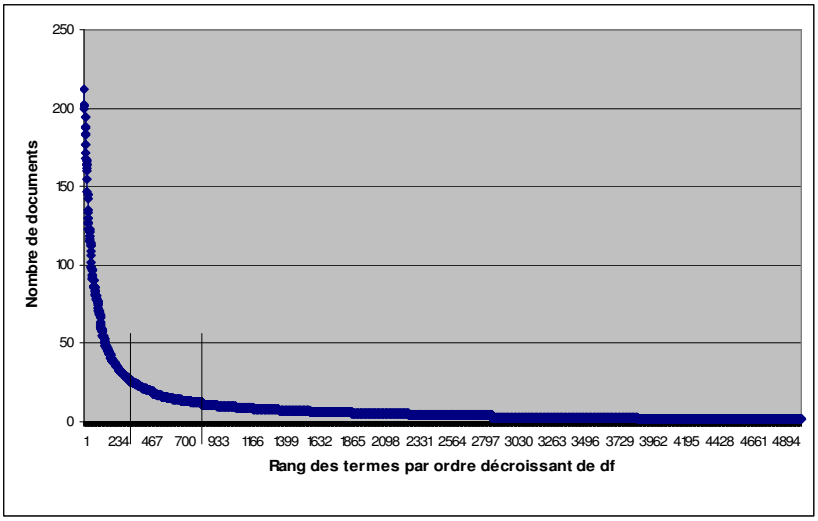


Figure 51. Nombre de documents dans lesquels un terme de rang N apparaît

La répartition des termes dans notre corpus est telle que :

- les 10 mots les plus fréquents correspondent à : de, la, le, les, à et, des, en, que, qui.

Ces mots sont effectivement non porteurs de sens puisque correspondant aux articles, adverbes, prépositions, auxiliaires, pronoms, conjonctions, …

- les 10 mots les moins fréquents correspondent à : londonienne, concédé, sonrai, tentent, Rogers, communisme, couplage, matins, incontournable, subsisté.

Ces mots n'apparaissant qu'une fois dans l'ensemble du corpus s'avèrent également faiblement porteurs de sens et donc de faible importance et pertinence.

- Au niveau des mots moyennement fréquents, on trouve :
- Des substantifs propres : Arafat, Butler, ONU, Figaro…
- Des substantifs communs : couple, sujet, conflit, dimanche, congrès…
- Des adjectifs qualificatifs : dangereux, étranger, économique, diplomatique…
- Des verbes : puisse, revenir, retrouve, penser…

La distribution des catégories morphosyntaxiques au sein de notre corpus illustre bien le fait que les mots les plus porteurs de sens, et de fait les plus pertinents, correspondent aux termes moyennement fréquents au sein d'un corpus.

4. Conclusion

En recherche d'information, des hypothèses sur la distribution des termes dans un corpus de documents textuels servent de base à la mise en place d'un système de recherche d'information. Ces hypothèses permettent de définir les protocoles d'extraction des mots afin d'effectuer une représentation des documents sous forme de documents indexés.

Ce chapitre décrit l'étude de la vérification de l'applicabilité de ces hypothèses, initialement établies sur des corpus de documents textuels, aux données issues de transcriptions de l'oral. Pour ce faire, deux corpus de transcriptions textuelles de l'oral servent à l'analyse : des transcriptions de conversations téléphoniques et des transcriptions d'émissions radiophoniques.

Les résultats montrent le caractère encore plus répétitif du vocabulaire de conversations comparé au vocabulaire des documents textuels classiques, puisque les mots les 10% les plus fréquents des documents 'conversations' couvrent 90% du vocabulaire contre 70 à 80% pour les documents textuels classiques. De plus, on retrouve la loi de Zipf montrant qu'en ordonnant les mots différents d'un texte par ordre de fréquence décroissante, la fréquence d'un mot est inversement proportionnelle à son rang.

Comme pour les documents textuels classiques, les mots moyennement fréquents dans le corpus correspondent aux mots porteurs de sens et les mots très ou peu fréquents aux mots non porteurs de sens. La conjecture de Luhn s'applique donc aux données issues de transcriptions de l'oral.

Chapitre VII. Validation de la pondération

L'objectif de ce chapitre consiste à valider la prise en compte de la certitude dans la fonction de pondération utilisée pour la représentation des documents sous forme de documents indexés.

Pour cela un corpus de documents issus de la collection-test de CLEF-2004 est utilisé. Nous décrivons ce corpus dans la première partie de ce chapitre. Ensuite, nous détaillons le protocole expérimental basé sur une recherche d'information étiquetée syntaxiquement. Nous montrons enfin l'apport de notre proposition par le biais des résultats obtenus aux expérimentations.

1. Corpus

1.1. Définition

La collection-test utilisée pour les évaluations provient des données de CLEF-2004. Seule la partie de la collection-test en français dont les documents correspondent à des articles issus du journal Le Monde est employée. La collection-test utilisée se compose de 47 646 documents et 50 requêtes résolues et rédigées en langage naturel. Ces requêtes comprennent un titre, une partie « description » correspondant à une phrase résumant la requête et une partie « narration » c'est-à-dire un court paragraphe détaillant les documents considérés comme pertinents ou non. Voici un exemple de requête :

<top>

<num> C204 </num>

<FR-title> Victimes d'avalanches </FR-title>

<FR-desc> Trouver des informations sur le nombre de morts par avalanche. </FR-desc>

<FR-narr> Les documents pertinents doivent donner des détails sur le nombre de personnes qui meurent à cause des avalanches, que ce soit dans la description de cas spécifiques d'avalanche avec des victimes ou des statistiques générales sur le nombre de morts à cause d'avalanches. </FR-narr>

</top>

Les expérimentations utilisent seulement les parties « title » et « desc ».

1.2. Prétraitements des documents

Le corpus comporte 47 646 documents composés en tout de 21 581 650 mots. Le corpus lemmatisé compte 10 979 202 mots. Le vocabulaire final de l'ensemble des documents se

compose de 125 981 mots (une fois les mots vides issus d'un anti-dictionnaire enlevés). Le nombre élevé de mots dans le vocabulaire final s'explique par la conservation notamment des nombres (dates, …) et des noms propres.

2. Cadre d'étude

Afin de valider l'intérêt d'introduire la valeur de certitude à la fonction de pondération, nous proposons une expérimentation basée sur une recherche d'information avec étiquetage syntaxique.

Cela signifie qu'un utilisateur n'exprime plus sa requête simplement sous la forme d'une liste de termes mais précise la catégorie morphosyntaxique de chacun des termes. Ainsi les requêtes « (porte, substantif commun) » et « (porte, verbe conjugué) » sont distinctes. L'une concerne les portes (d'une maison, d'un placard …) et l'autre le verbe porter. Un analyseur morphosyntaxique détermine les catégories morphosyntaxiques des termes durant la phase d'indexation des documents du corpus. Dans ce contexte, la certitude associée à chaque terme ne porte pas sur le terme en lui-même mais sur sa catégorie morphosyntaxique. La certitude associée à chaque terme correspond donc à la confiance donnée par l'analyseur à chaque catégorie morphosyntaxique.

Nous commençons par présenter l'adaptation de notre proposition de fonction de pondération incertaine à ce contexte de recherche d'information avec étiquetage syntaxique. Nous développons ensuite les systèmes comparés. En annexe 4, nous rappelons les fonctions de pondérations classiquement utilisées en recherche d'information.

2.1. Notations préliminaires

Voici les notations que nous utilisons dans la suite :

- N = nombre de documents dans le corpus
- tf (t,d) = term frequency = nombre d'occurrences du terme *t* dans le document *d*
- totfreq(t) = nombre d'occurrences de *t* dans le corpus = $\sum_{i=1}^{N} tf(t,d_i)$
- df (t) = document frequency = le nombre de documents indexés par *t*

2.2. La donnée incertaine dans le contexte « étiqueteur syntaxique »

A chaque terme du document, fourni en entrée d'un processus d'étiquetage syntaxique, correspond le même terme en sortie du processus associé à une étiquette syntaxique. Pour chaque terme, on dispose des différentes étiquettes syntaxiques possibles, chacune étant associée à une valeur de certitude (cf. Figure 52).

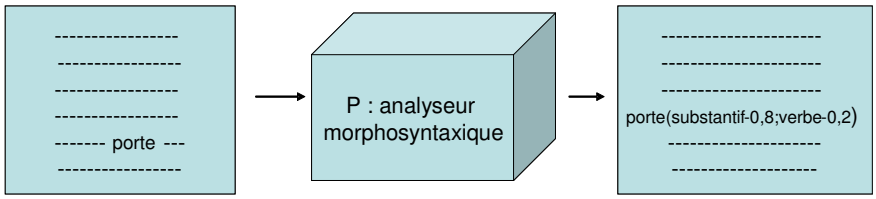


Figure 52. D'une donnée "simple" à une donnée étiquetée syntaxiquement avec valeur de certitude

Le processus d'étiquetage syntaxique prend en entrée une donnée « simple » (*terme*) et fournit en sortie une donnée de la forme : *terme (Catégorie$_1$-poidsCatégorie$_1$, Cat$_2$-poidsCat$_2$, …, Cat$_i$-poidsCat$_i$).*

Dans notre exemple (cf. Figure 52), en sortie, on a « porte (substantif-0,8 ; verbe-0,2) ». Cette dernière notation signifie qu'il existe une ambiguïté sur le terme porte, avec une certitude de 80% pour le type substantif commun contre 20% pour verbe conjugué.

Dans ce contexte, les mots s'avèrent parfaitement identifiés. A contrario il existe une incertitude sur leur catégorie morphosyntaxique, le système d'analyse associe une ou plusieurs catégories à chaque terme. Ainsi en sortie de l'analyseur, un document se présente sous la forme d'une liste de mots associés à une ou plusieurs catégories et ce de façon pondérée.

Nos expérimentations portent sur la pondération de ce dernier type de données : tout document est une séquence de mots, chacun associé à une liste pondérée de catégories. Par exemple, dans la Figure 52, le mot 'porte' s'avère lié à la liste ((Substantif, 0,8), (Verbe Conjugué, 0,2)).

Dans un même document, le même mot peut apparaître plusieurs fois. De fait, il existe un lien entre un mot et différentes listes pondérées de catégories. Par exemple, dans un même document, le terme 'porte' peut se trouver présent 3 fois, 1 fois associé à la liste (Substantif, 1), une fois à la liste (Verbe Conjugué, 1), et une fois à la liste ((Substantif, 0,8), (Verbe Conjugué, 0,2)).

2.3. Modélisation des documents

Ce paragraphe décrit la modélisation des documents et documents étiquetés syntaxiquement dans le cadre d'une recherche catégorisée syntaxiquement.

2.3.1 Vocabulaire

Soit $\mathcal{V}$ est un ensemble fermé de Z termes :

$\mathcal{V} = \{t_1, t_2, \ldots, t_Z\}$

2.3.2 Document

Un document $\mathcal{D}$ se représente par une séquence de longueur L :

$\mathcal{D} = [t_{d,1}, t_{d,2}, t_{d,3}, \ldots, t_{d,L}]$ avec $t_{d,i} \in \mathcal{V}$.

On peut bien évidemment rencontrer plusieurs fois le même terme dans un document et les $t_{d,i}$ correspondent aux termes dans l'ordre de leur apparition dans le document.

2.3.3 Document catégorisé

a. Catégories

Soit C un ensemble fini de C catégories (ex : l'ensemble des catégories morpho-syntaxiques d'un analyseur) :

$C = \{c_1, c_2, \ldots, c_c\}$

b. Document catégorisé

Un document catégorisé $\mathcal{DC}$ représente le document $\mathcal{D}$ à l'issue de l'étiqueteur syntaxique.

A partir de la représentation du document $\mathcal{D} = [t_{d,1}, t_{d,2}, t_{d,3}, \ldots, t_{d,L}]$ avec $t_{d,i} \in \mathcal{V}$, on définit son document catégorisé $\mathcal{DC}$:

$\mathcal{DC} = [\chi_{d,1}, \chi_{d,2}, \chi_{d,3}, \ldots, \chi_{d,L}]$

Avec $\chi_{d,i} = (t_{d,i}, \{c_j, p_{i,j}\})$ et $t_i \in \mathcal{V}$, $c_j \in C$, $p_{i,j} \in [0,1]$

$\forall (t, c_i, p_i) \in \chi_i$.

Ce qui signifie que, pour chaque terme de $\mathcal{D}$, on connaît, dans le document $\mathcal{DC}$, l'ensemble complet de toutes ses catégories potentielles.

2.3.4 Document indexé

Un document indexé $\mathcal{DI}$ représente le contenu d'un document $\mathcal{D}$, à partir du document catégorisé $\mathcal{DC} = [\chi_{d,1}, \chi_{d,2}, \chi_{d,3}, \ldots, \chi_{d,L}]$

$\mathcal{DI} = \{\chi_i\}$ avec $\chi_i = (t_i, \{c_j, w_{ij}\})$, $t_i \in \mathcal{V}$

Et $\forall\ \chi_i \in \mathcal{DC}$ – *on sait que* $\chi_i = (t_i, \{c_j, w_{ij}\})$ - , $\exists\ !\ \chi_k \in \mathcal{DI}$, $t_i \in \chi_k$

2.4. Pondération des données incertaines : Calcul des w_{ij}

Le poids w_{ij} du terme t_i avec la catégorie c_j dans le document indexé $\mathcal{DI}$ issu du document $\mathcal{D}$ se fait par analogie avec les fonctions locales et globales connues (cf. annexe 4). Dans notre approche, nous prenons comme cadre de travail les fonctions $fL_4(t,d) = \log(tf(t,d)+1)$ et $ifG_1(t,CO) = \log\left(\frac{N}{df(t)}\right)$.

2.4.1 Notations

On définit *dc* comme un document de DC : $dc \in \mathcal{DC}$.

- $tf(t,c,dc)$ = la somme des poids des apparitions de *t* avec la catégorie *c* dans le document *dc*

$$\text{Soit : } tf(t,c,dc) = \sum_{\chi_i \in dc} p_i \text{ avec } \chi_i \in dc,\ \chi_i = (t, \{c_j, p_j\}),\ (t,c,p) \in \chi_i$$

- $df(t,c,CO)$ = le nombre de documents contenant *t* avec la catégorie *c*

$$\text{Soit : } df(t,c,CO) = \left\| \{D \in CO, \text{ tel que } tf(t,c,DC) > 0\} \right\|$$

2.4.2 Détermination des valeurs de pondération

a. Force locale

Par analogie avec la force locale $fL_4(t,d) = \log(tf(t,d)+1)$, on détermine la force locale pour un terme *t* avec une catégorie *c* dans un document *d* comme :

$$fL_5(t,c,d) = \log((tf(t,c,d))+1)$$

b. Force globale

Pour exprimer la force globale d'un terme, nous utilisons la mesure $ifG_1(t,CO)$ adaptée à notre contexte afin de prendre en compte la catégorie du terme :

$$ifG_4(t,c,CO) = \log\left(\frac{N}{df(t,c,CO)}\right)$$

c. Fonction de pondération

Par analogie avec la fonction de pondération w_4, nous proposons d'utiliser une fonction de pondération de type *tf.idf,* que nous noterons w_5 :

$$w_5(t,c,d) = fL_5(t,d) * ifG_4(t,c,CO)$$

3. Expérimentations

3.1. Les systèmes comparés

Nous nous plaçons dans le cadre du modèle vectoriel [Salton, 1971] et nous comparons notre pondération prenant en compte les incertitudes et une pondération classique de type tf.idf. Les requêtes utilisées restent les mêmes pour les deux expérimentations, à savoir les requêtes de CLEF. Pour évaluer notre proposition, nous étiquetons manuellement ces requêtes, ainsi chaque catégorie associée à chacun des termes de la requête se trouve sûre.

La fonction de correspondance de type cosinus permet de déterminer les documents pertinents selon une requête. Ainsi, la similarité entre le document j et une requête q se définit par :

$$similarité(d_j, q) = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2 \sum_{i=1}^{n} w_{i,q}^2}}$$

Dans le cas « classique », la fonction de pondération utilisée est telle que $w_{i,j} = w_4(t,d)$.

Pour notre pondération, afin de prendre en compte le couple (mot, {catégorie, poids de la catégorie}), on a $w_{i,j} = w_5(t,c,d)$.

Le traitement des documents dans le cadre de notre proposition s'effectue grâce à l'étiqueteur syntaxique IOTA [Chiaramella, 1986]. Le système IOTA fournit pour chaque mot, l'ensemble de ses catégories possibles. Chaque mot se trouve suivi au minimum d'une catégorie et au maximum de 6 catégories. L'étiqueteur ne fournissant pas de valeurs de certitude associées aux catégories, nous fixons les poids à attribuer à chaque catégorie en fonction du nombre de catégories renvoyées par le système (cf. Tableau XI).

Nombre de catégories possibles	Poids catégorie en 1ère position	Poids catégorie en 2ème position	Poids catégorie en 3ème position	Poids catégorie en 4ème position	Poids catégorie en 5ème position	Poids catégorie en 6ème position
1	1					
2	1	0,5				
3	1	0,45	0,45			
4	1	0,4	0,4	0,4		
5	1	0,35	0,35	0,35	0,35	
6	1	0,3	0,3	0,3	0,3	0,3

Tableau XI. Valeurs de certitude selon la position et le nombre de catégories

3.2. Analyse des résultats

3.2.1 Pondération classique versus notre proposition

a. Rappel - précision

Notre fonction de pondération $w_5(t,c,d)$ améliore la précision pour les premiers points de rappel (cf. Tableau XII) par rapport à la fonction de pondération classique $w_4(t,d)$.

	$w_4(t,d)$ ou tf*idf	$w_5(t,c,d)$
0	0,4129	0,4555
0,1	0,3800	0,4066
0,2	0,3710	0,3505
0,3	0,3559	0,3379
0,4	0,3132	0,2892

Tableau XII. Rappel – précision

Ce constat correspond à nos attentes : notre système a pour but d'améliorer la précision par l'apport d'information supplémentaire pour chaque terme, en l'occurrence leurs catégories potentielles.

b. Mesure ESL

La mesure ESL (Expected Search Length), introduite par [Cooper, 1968] permet d'évaluer le nombre de documents non pertinents qu'un utilisateur rencontre avant de lire *n* documents pertinents :

$$ESL(n) = j + \frac{i.s}{r+1}$$ avec

- n : le nombre de documents pertinents que l'on veut lire
- j : le nombre total de documents non pertinents dans les rangs précédents le rang final (c'est-à dire le rang où on se situe lorsque l'on a lu *q* documents pertinents)
- r : le nombre de documents pertinents dans le rang final
- i : le nombre de documents non pertinents dans le rang final
- s : le nombre de documents pertinents requis dans le rang final (en général s=1)

Nous avons appliqué cette mesure au modèle vectoriel ainsi qu'à notre proposition (cf. Figure 53).

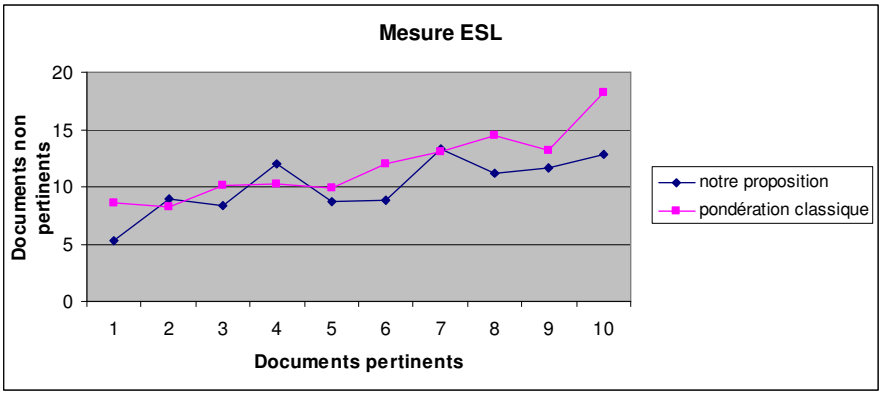


Figure 53. Mesure ESL : pondération classique vs notre proposition

Cette mesure montre que notre proposition améliore les résultats au niveau utilisateur. Ainsi notre proposition améliore non seulement la précision mais également la « facilité » avec laquelle notre système élimine les documents non pertinents.

3.2.2 Apport mutuel

Validant expérimentalement les meilleurs résultats de la pondération classique au niveau du rappel et ceux de notre proposition au niveau de la précision, nous choisissons d'effectuer une combinaison de ces deux méthodes :

MéthodeCombinée = (1-α) pondération_classique + α notre_proposition

Une valeur $\alpha = 0.7$ permet d'augmenter la précision du système tout en donnant un poids fort à notre proposition.

a. Rappel – Précision

La combinaison des résultats augmente significativement les résultats (cf. Tableau XIII). La méthode combinée permet un meilleur rappel.

	$w_4(t,d)$ ou tf*idf	$w_5(t,c,d)$	Méthode combinée
0	0,4129	0,4555	0,4865
0,1	0,3800	0,4066	0,4484
0,2	0,3710	0,3505	0,3924
0,3	0,3559	0,3379	0,3811
0,4	0,3132	0,2892	0,3255
0,5	0,3038	0,2760	0,3136
0,6	0,2696	0,2096	0,2280
0,7	0,2248	0,1346	0,1628
0,8	0,1856	0,1045	0,1292
0,9	0,1324	0,0938	0,1063
1	0,1108	0,0668	0,0768

Tableau XIII. Rappel – Précision

3.3. Mesure ESL

Si on applique la mesure ESL présentée plus tôt, les résultats s'avèrent un peu moins satisfaisants pour la méthode combinée que pour notre proposition (cf. ***Figure 54***). De ce fait, si le but du système de recherche d'information consiste à obtenir une précision élevée sur les premiers points de rappel ainsi qu'une bonne élimination des documents non pertinents, l'utilisation de notre proposition seule donne les meilleurs résultats. Si le système vise est une amélioration des résultats au niveau rappel et précision, la méthode combinée demeure plus appropriée.

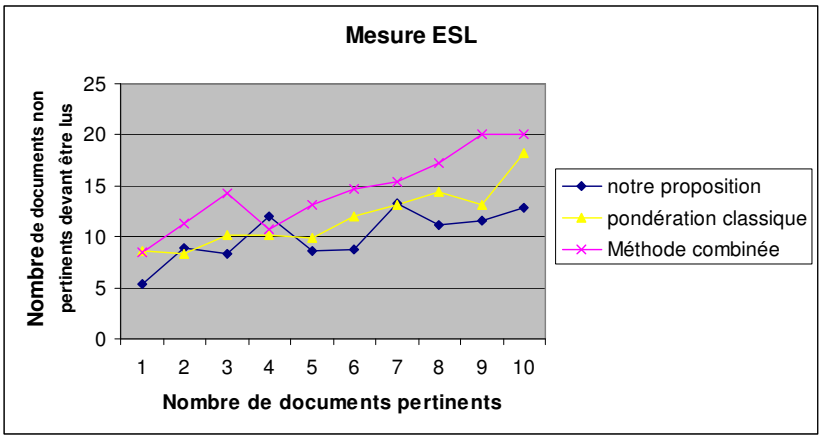


Figure 54.Mesure ESL

4. Conclusion

Cette expérimentation décrit une première approche de modélisation des documents incertains ainsi que leur intégration dans un système de recherche d'information au niveau de la pondération des mots. Les modèles classiques s'appuient sur un décomptage des apparitions des mots dans les documents. Nous revisitons ici cette notion en tenant compte de l'apparition des termes ainsi que de l'incertitude qui leur est associée.

Nous montrons qu'il s'avère possible d'adapter les fonctions de pondération classiquement utilisées en recherche d'information afin de prendre en compte l'incertitude des termes et que cette adaptation présente un apport au sein de systèmes de recherche d'information orientés précision. Pour cela, nous confrontons notre proposition à une pondération classique tf.idf au sein du modèle vectoriel.

Nous mettons en évidence que notre fonction de pondération améliore non seulement la précision sur les premiers points de rappel mais également la « facilité » avec laquelle notre système élimine les documents non pertinents, par le biais de l'évaluation de la mesure ESL. Enfin, nous montrons que la combinaison de la pondération classique et de notre proposition de pondération permet une amélioration générale du rapport rappel / précision.

Chapitre VIII. Validation de la fonction de correspondance

Après avoir montré expérimentalement l'apport de la prise en compte de l'incertitude et la façon de l'intégrer dans la fonction de pondération proposée dans notre modèle, nous validons la fonction de correspondance.

Pour cela, ce chapitre présente le corpus utilisé pour les expérimentations, à savoir des transcriptions d'émissions radiophoniques issues de la campagne d'évaluation d'ESTER en 2005. Nous décrivons ensuite le protocole expérimental ainsi que les résultats obtenus. Enfin, nous dressons un bilan général de la validation de notre proposition.

1. Corpus

1.1. Définition

Nous disposons d'un corpus issu de la campagne d'évaluation ESTER (Evaluation des Systèmes de Transcription Enrichie d'Émissions Radiophoniques) en 2005. Ce corpus se compose de documents issus de transcriptions d'émissions radiophoniques.

L'émission de ces journaux radiophoniques a eu lieu entre les années 1998 et 2000 sur les radios « France Inter » et « RFI ». Ces journaux durent 20 minutes ou une heure, soit une durée totale de 9 heures et 20 minutes réparties ainsi (cf. Tableau XIV) :

Corpus	Durée	Nombre de documents	Durée moyenne des documents
France Inter Journaux de 1h	4h	85	3 min
France Inter Journaux de 20min	1h20min	25	3 min
RFI Journaux de 1h	4h	122	2 min
TOTAL	9h20min	231	

Tableau XIV. Description du corpus ESTER

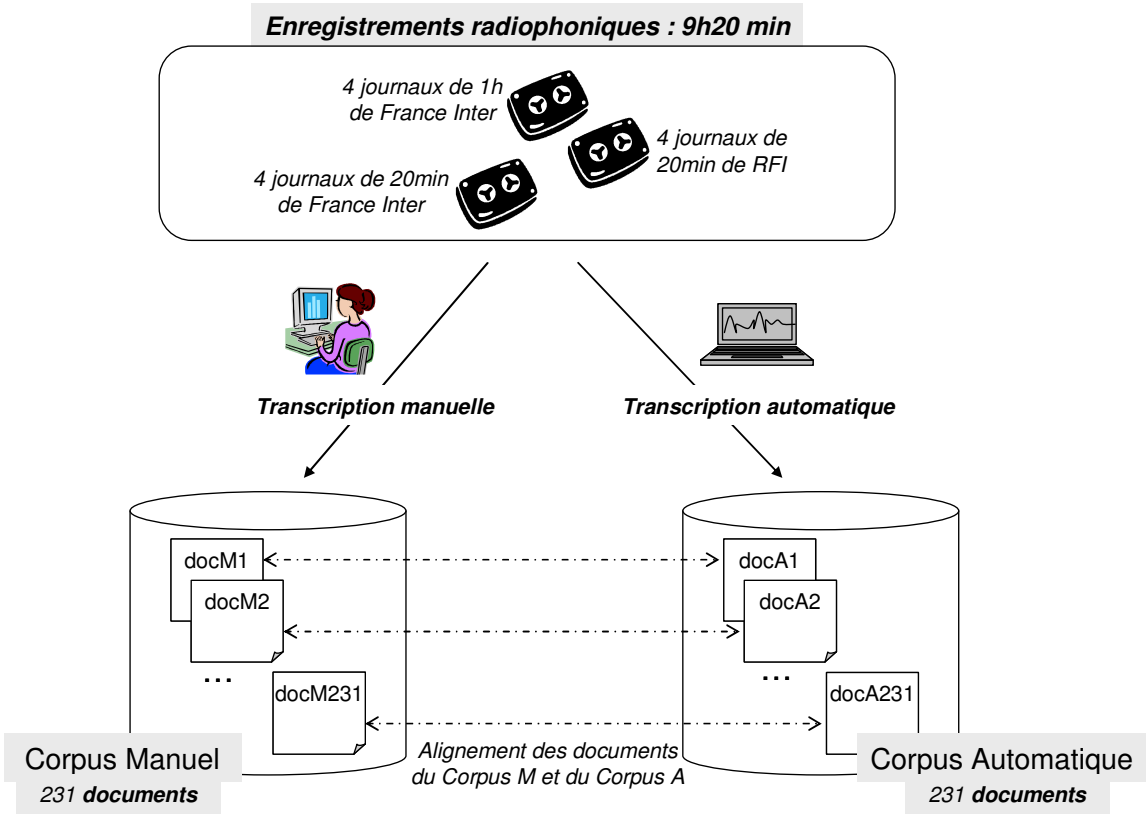


Figure 55. Description du corpus ESTER

Ces 9 heures 20 minutes d'enregistrements correspondent à 231 documents. Pour chaque document, nous disposons de la transcription manuelle et de la transcription automatique faite par un système de reconnaissance automatique de la parole (cf. Figure 55). Ce traitement s'effectue par le système de reconnaissance de la parole de l'équipe GEOD[13]. Le taux d'erreurs de reconnaissance avoisine les 30-35%. Ces erreurs sont de trois types :

- Mots mal reconnus (par exemple '*élection*' au lieu de '*connexion*')
- Mots oubliés (i.e. non retranscrits)
- Mots ajoutés (i.e. insertion de mots non présents dans l'enregistrement audio original).

[13] http://www-clips.imag.fr/geod/

1.2. Analyse des documents

Afin de décrire au mieux les documents du corpus, une analyse sur la distribution des mots a été effectuée.

Un comptage du nombre moyen de mots et du nombre moyen de mots différents par document est effectué sur chacun des deux corpus manuel et automatique (cf. Figure 56).

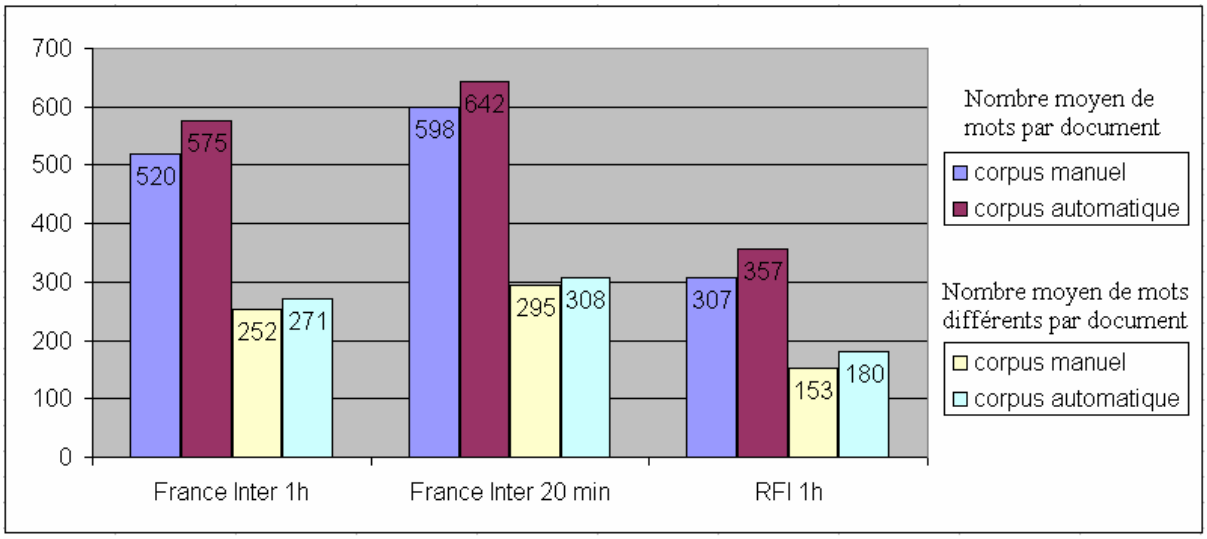


Figure 56. Nombre moyen de mots et mots différents par document

La comparaison des statistiques obtenues pour chacun des deux corpus (manuel et automatique) met en évidence que le nombre moyen de mots et le nombre moyen de mots différents par document s'avère plus élevé pour le corpus automatique que pour le corpus manuel (cf. Figure 56). Ceci s'explique par le fait que souvent plusieurs mots remplacent les mots mal reconnus, le système de reconnaissance automatique de la parole ajoute donc des mots.

Exemple :

Manuel	Automatique	Différence nombre de mots (automatique – manuel)
sur France-Inter	sur fond sa terre	+1
selon bill clinton ces attaques aériennes	selon des films comme ses attaques aériennes	+2

1.3. Analyse des requêtes

Elles se composent de peu de mots, deux ou trois mots. Elles traitent d'un sujet bien ciblé en rapport avec l'actualité (cf. Tableau XV).

Requête 1 : blocus transports routiers
Requête 2 : sommet New York
Requête 3 : jeux olympiques
Requête 4 : temps soleil
Requête 5 : bombardements irak

Tableau XV. Requêtes utilisées

2. Expérimentations

2.1. Protocole expérimental

Les expérimentations ont pour objectif de montrer l'apport de la prise en compte des approximations de termes dans la fonction de correspondance par rapport à la seule prise en compte des égalités de termes.

Nous rappelons que la fonction de correspondance s'exprime de la façon suivante (Définition 17) :

$$\mathcal{P}(Q|\mathcal{M}_D) = \prod_{i=1}^{\mathcal{N}_q} Cert(t_{q,i}) \times \left[\mu\lambda\,\mathcal{P}(t_{q,i}|\mathcal{M}_D) + \mu(1-\lambda)\mathcal{P}(t_{q,i}|\mathcal{M}_C) + (1-\mu)\mathcal{P}(\sim t_{q,i}|\mathcal{M}_D)\right]$$

Nous testons un certain nombre de paramètres utiles à la fonction de correspondance :

- $\boldsymbol{\lambda}$: ce paramètre permet de faire varier l'importance donnée au lissage, c'est-à-dire à la prise en compte du fait qu'un terme de la requête peut être absent du document et ainsi éviter un score nul pour la requête toute entière. Si λ = 1, il n'y a pas de lissage dans la fonction de correspondance. Le but de ces expérimentations ne portant pas sur l'évaluation de l'importance du lissage dans les modèles de langue (leur intérêt a été démontré dans de nombreux travaux, cf. état de l'art), nous fixons une valeur λ comme une constante pour l'ensemble des expérimentations afin que cette valeur n'influe pas sur les résultats.
- $\boldsymbol{\mu}$: ce paramètre permet de faire varier l'importance donnée aux approximations de termes. Dans nos expérimentations, nous faisons fluctuer sa valeur afin de voir son impact sur les performances du système de recherche d'information.

La fonction $\mathcal{P}(\sim t_{q,i}|\mathcal{M}_D)$ tenant compte des approximations se décompose comme suit :

Définition 15 : $\mathcal{P}\left(\sim t_{q,i} \middle| \mathcal{M}_{\mathcal{D}}\right) = \frac{\sum_{t_{d,j} \in Approximation\left(t_{q,i}\right)} ValConc(t_{q,i}, t_{d,j}) \times ValInter\left(t_{q,i}, t_{d,j}\right)}{|\mathcal{D}|}$

deux paramètres sont pris en compte : *ValConc* et *ValInter*.

- ***ValConc*** : les valeurs de *ValConc* dépendent du type de concordance rencontré entre deux termes. Nous pourrons également faire varier ces valeurs afin de donner plus ou moins d'importance à l'un ou l'autre des types de concordance.
- ***ValInter*** : comme pour *ValConc*, ses valeurs on été fixées et nous pourrons les faire varier afin d'évaluer leur impact.

Enfin, nous discutons le seuil d'appariement servant à ne pas prendre en compte toutes les approximations de termes.

- ***Valeur seuil d'appariement*** : l'appariement entre deux termes se calcule à l'aide des valeurs *ValConc* et *ValInter*. La valeur d'appariement exprime le degré de 'presque égalité' entre deux termes. Cette valeur s'avère très faible pour certains termes, d'où la nécessité de déterminer une limite de considération de 'presque égalité' entre deux termes : le seuil d'appariement. Les expérimentations permettent de le définir de façon optimale. Toutefois, cette valeur peut différée suivant le contexte d'utilisation du système. En effet, un système dont les données s'avèrent majoritairement exactes doit prendre en compte moins d'approximations qu'un système où les données se trouvent fortement incertaines.

2.2. Résultats

2.2.1 Seuil d'approximations

Afin de montrer l'utilité de la mise en place d'un seuil de prise en compte des approximations, nous étudions statistiquement le nombre d'approximations prises en compte par chaque document et l'évolution de ce nombre en fonction d'un seuil sur la valeur d'appariement.

Pour ce faire, nous avons utilisé le corpus automatique de 232 documents et une requête composée d'un seul terme : « monde ».

Ainsi la Figure 57 montre que, sans fixer de seuil, sur les 232 documents du corpus, 115 prennent en compte moins de 50 approximations, 56 entre 50 et 100 approximations, 23 entre 100 et 150 approximations, 24 entre 150 et 200 approximations, 6 entre 200 et 250 approximations et 8 plus de 250 approximations.

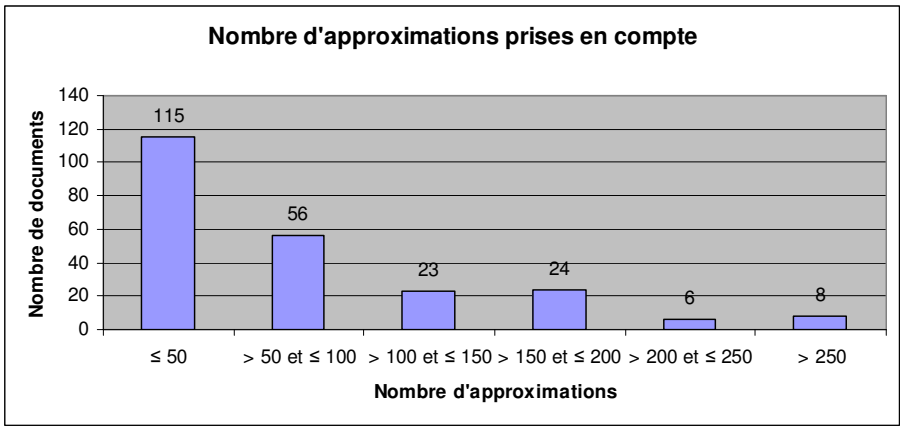


Figure 57. Nombre d'approximations prises en compte par document

La Figure 58 montre que le nombre d'approximations diminue considérablement lorsque nous fixons un seuil pour les valeurs d'appariement. Quasiment les mêmes répartitions du nombre d'approximations prises en compte par document apparaissent mais le nombre se trouve de l'ordre de 10 fois moins important. Ainsi pour un seuil = 0,4, on a 211 documents ayant moins de 5 approximations, 19 documents entre 5 et 10, et 2 documents entre 10 et 15. Pour un seuil = 0,2, on constate que 146 documents tiennent compte de moins de 5 approximations, 65 entre 5 et 10, 20 entre 10 et 15 et 1 plus de 15.

Il s'avère donc nécessaire de fixer un seuil pour la prise en compte des approximations afin de ne pas introduire trop de bruit en conservant des mots ayant une presque égalité très faible avec un terme de la requête.

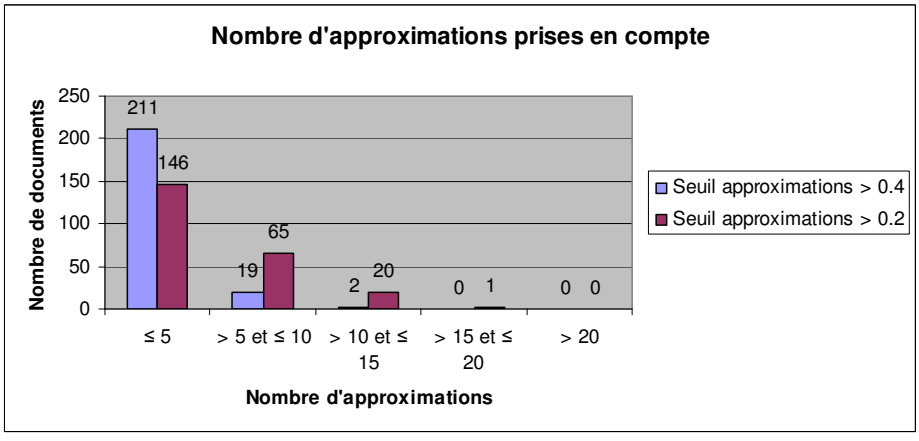


Figure 58. Nombre d'approximations prises en compte par document en fonction d'un seuil

Le seuil d'appariement fixé, nous étudions quelle importance donner aux 'presque égalités' au sein de la fonction de correspondance, afin d'améliorer les résultats de recherche.

2.2.2 Variations de μ

Nous rappelons que la fonction de correspondance s'exprime de la manière suivante (Définition 17) :

$$\mathcal{P}(Q|\mathcal{M}_D) = \prod_{i=1}^{\mathcal{N}_q} Cert(t_{q,i}) \times [\mu\lambda \mathcal{P}(t_{q,i}|\mathcal{M}_D) + \mu(1-\lambda)\mathcal{P}(t_{q,i}|\mathcal{M}_C) + (1-\mu)\mathcal{P}(\sim t_{q,i}|\mathcal{M}_D)]$$

Le paramètre μ influe sur l'importance donnée à la prise en compte des 'presque égalité' dans la fonction de correspondance. Plus la valeur de μ s'avère élevée, moins la prise en compte des 'presque égalité' demeure importante.

2.2.3 Analyse des résultats

Nous étudions les résultats obtenus avec trois valeurs de μ et pour des valeurs du seuil d'approximations de 0,2, 04 et 0,6 :

- μ = 0,5 : autant d'importance donnée aux 'presque égalité' qu'aux égalités de termes (cf. Figure 59)
- μ = 0,8 : plus d'importance donnée aux égalités qu'aux 'presque égalités' (cf. Figure 60)
- μ = 0,9 : encore plus d'importance donnée aux égalités (cf. Figure 61).

Ces évaluations permettent de définir le seuil d'approximations le plus approprié et l'importance à donner aux approximations par le biais du paramètre μ.

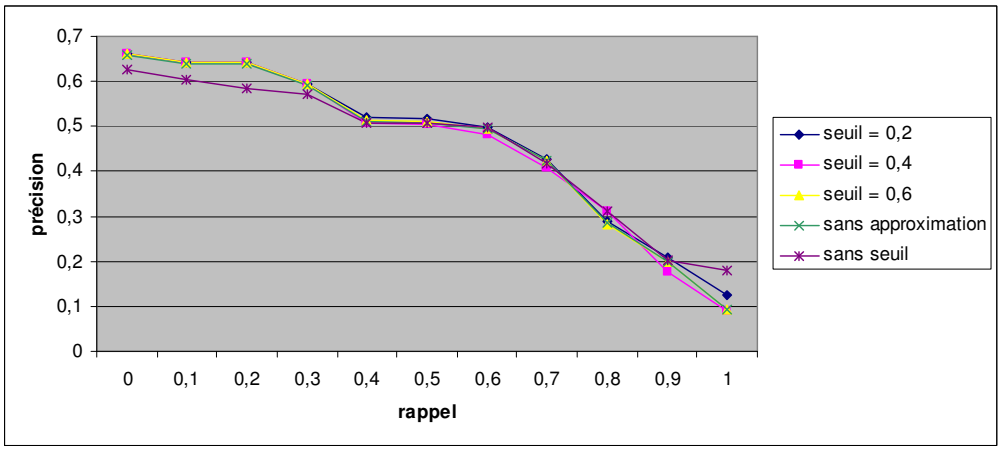


Figure 59. Courbe rappel précision avec μ = 0,5

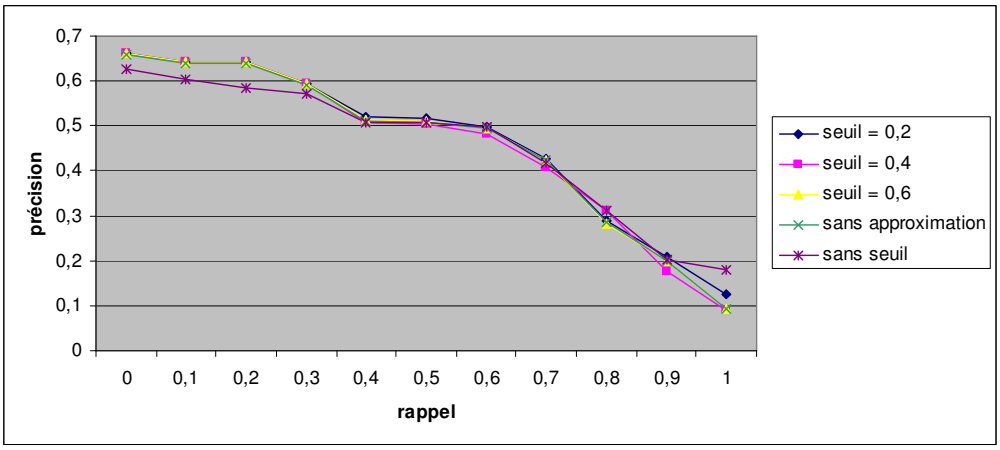


Figure 60. Courbe rappel précision avec μ = 0,8

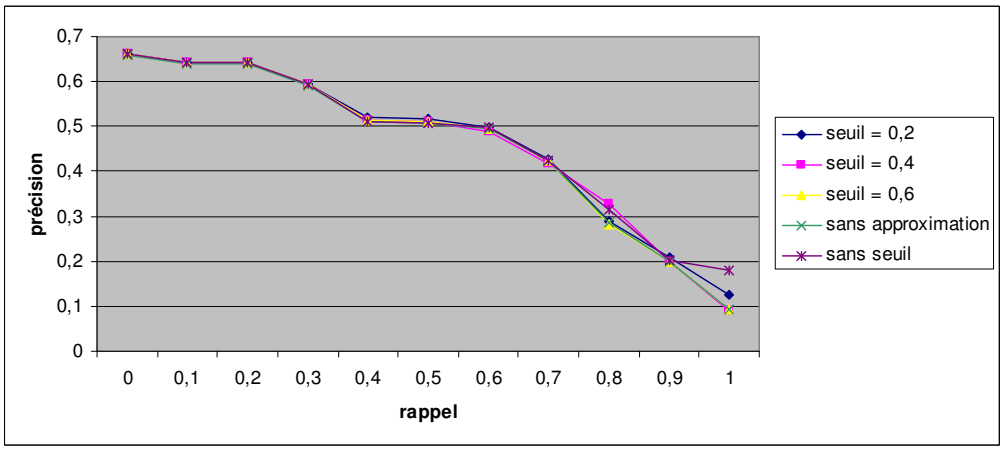


Figure 61. Courbe rappel précision avec μ = 0,9

a. Seuil d'approximation

Ces expérimentations confirment que la mise en place d'un seuil d'approximations à prendre en compte est nécessaire. En effet, les résultats sont moins bons lorsque aucun seuil n'est fixé. Cette affirmation demeure moins visible dans le cas où μ = 0,9 à cause de la faible importance donnée à la prise en compte des approximations dans le score du document : 10%. On peut donc dire qu'un seuil d'approximation fixé à 0,2 donne les meilleurs résultats en matière de rappel/précision quelque soit la valeur de μ. Il s'avère nécessaire de ne pas prendre en compte toutes les approximations des termes mais seulement ceux donc la combinaison de la concordance et l'intersection amène à une valeur d'appariement $ValApp > 0,2$.

Le seuil d'approximation fixé, nous déterminons la valeur de μ la plus adaptée à notre contexte.

b. Prise en compte des approximations

Nous effectuons nos évaluations en fixant le seuil d'approximation à 0,2. Les résultats nous montrent les plus fortes améliorations avec μ = 0,8 (cf. Figure 62). Nous avons pu voir sur les courbes précédentes qu'une trop forte prise en compte des approximations, μ = 0,5, diminuaient les performances (cf. Figure 59).

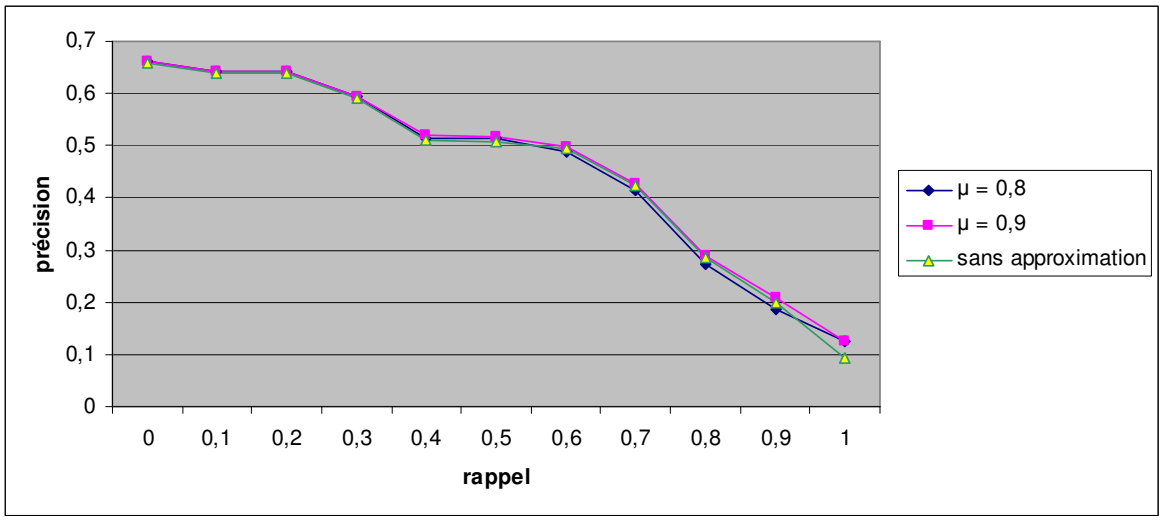


Figure 62. Courbe rappel précision avec un seuil d'approximations = 0,2

Le Tableau XVI montre que la prise en compte des approximations, avec une proportion de 20% des 'presque égalité' et de 80% des égalités, améliore ou tout du moins égale les performances de recherche.

Rappel	Transcription Automatique Sans approximation	μ = 0,8		
		0,2	0,4	0,6
0	0,6588	0,6609	0,6609	0,6609
0,1	0,6388	0,6409	0,6409	0,6409
0,2	0,6388	0,6409	0,6409	0,6409
0,3	0,5912	0,5933	0,5933	0,5933
0,4	0,5106	0,5143	0,5071	0,5126
0,5	0,5076	0,5128	0,5057	0,5096
0,6	0,4936	0,4894	0,4822	0,4895
0,7	0,4226	0,4151	0,408	0,4135
0,8	0,2844	0,2723	0,3116	0,2689
0,9	0,1988	0,1867	0,1757	0,1762
1	0,0928	0,1257	0,0909	0,0919

Tableau XVI. Rappel précision avec $\mu = 0,8$

3. Vers une fonction de lissage adaptée au contexte des données incertaines

Les fonctions de lissage utilisées dans les modèles de langue pallie le problème des mots de la requête absents des documents afin d'éviter les scores nuls pour des documents ne contenant pas certains mots de la requête. Dans notre fonction de correspondance, nous proposons de prendre en compte la probabilité des termes dans le corpus comme fonction de lissage. A cette fonction de lissage, nous ajoutons la prise en compte des approximations de termes. Toutefois, nous nous posons la question de considérer la prise en compte des approximations comme une fonction de lissage adaptée au contexte des données incertaines.

La fonction de correspondance devient (Définition 17) :

$$\mathcal{P}(Q|\mathcal{M}_{\mathcal{D}}) = \prod_{i=1}^{\mathcal{N}_q} Cert(t_{q,i}) \times \left[\mu\, \mathcal{P}(t_{q,i}|\mathcal{M}_{\mathcal{D}}) + (1-\mu)\mathcal{P}(\sim t_{q,i}|\mathcal{M}_{\mathcal{D}})\right]$$

Le même protocole expérimental est utilisé que pour les évaluations précédemment décrites.

Les résultats obtenus montrent une nette amélioration des performances de recherche en considérant les approximations comme une fonction de lissage (cf. Figure 63 et Tableau XVII).

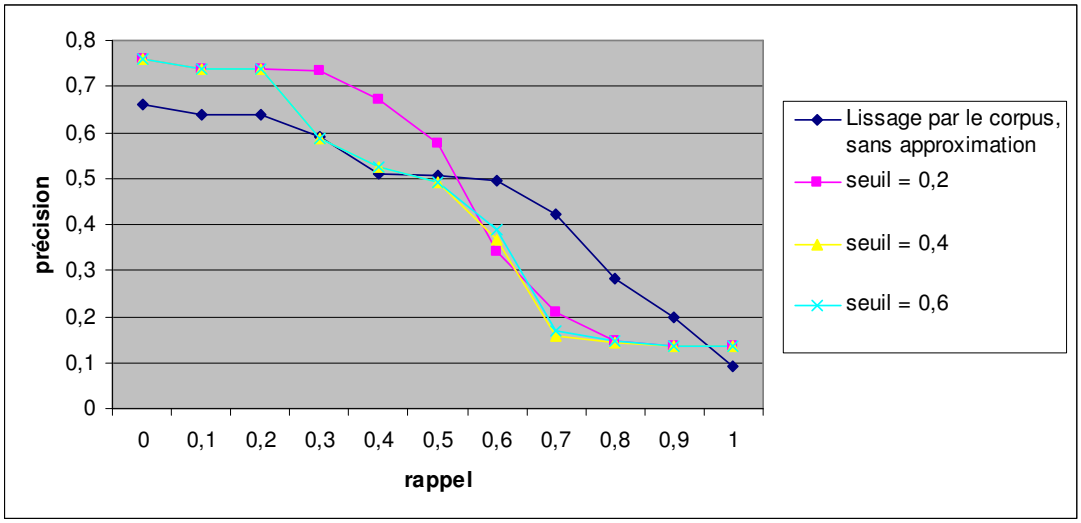


Figure 63. Courbe rappel/précision en considérant les approximations comme une fonction de lissage et avec μ = 0,8

Rappel	Transcription Automatique Avec lissage par le corpus et sans approximation	μ = 0,8		
		0,2	0,4	0,6
0	0,6588	0,7614	0,7614	0,7614
0,1	0,6388	0,7392	0,7364	0,7364
0,2	0,6388	0,7392	0,7364	0,7364
0,3	0,5912	0,7347	0,5873	0,5873
0,4	0,5106	0,6715	0,5241	0,5241
0,5	0,5076	0,5744	0,4908	0,4908
0,6	0,4936	0,3405	0,3654	0,3888
0,7	0,4226	0,2093	0,1564	0,1683
0,8	0,2844	0,1452	0,1446	0,1451
0,9	0,1988	0,1356	0,1349	0,1349
1	0,0928	0,1354	0,1342	0,1343

Tableau XVII. Rappel précision en considérant les approximations comme une fonction de lissage et avec μ = 0,8

Ces résultats mettent en avant l'intérêt d'une prise en compte des approximations des termes de la requête comme fonction de lissage. Cette proposition permet un lissage dépendant uniquement du document à juger et non du corpus comme généralement utilisé

dans les lissages. Cette alternative s'avère intéressante car elle ne nécessite pas un corpus d'apprentissage pour évaluer le poids d'un document.

Toutefois, bien qu'en considérant un seuil d'approximation très bas (0,2) les chances de ne pas trouver de terme approximant un terme de la requête absent du document soient faibles, cette éventualité reste possible. Dans ce cas de figure, deux possibilités demeurent envisageables : abaisser le niveau du seuil des approximations à prendre en compte ou introduire la composante lissage par le corpus très faiblement.

Cette expérimentation montre que, dans un contexte de données incertaines, la prise en compte de l'incertitude peut être vue comme une fonction de lissage au sein d'un modèle de langue et non pas seulement comme une dimension à ajouter à une fonction de correspondance contenant déjà une fonction de lissage.

4. Conclusion

Le système de recherche d'information adapté aux données incertaines que nous proposons se base sur la prise en compte de la certitude au sein de la pondération et de la notion de 'presque égalité' au sein de la fonction de correspondance.

Compte tenu de la difficulté à disposer d'un corpus permettant l'évaluation globale de notre système, nous avons divisé la validation de la proposition en deux parties : validation de la pondération et validation de la fonction de correspondance.

Un corpus de documents de journaux du Monde issus de la campagne CLEF-2004 sert à la validation de la fonction de pondération. Le Chapitre VII. a montré l'apport de l'intégration de la certitude au sein de la pondération avec une augmentation des résultats de recherche étiqueté syntaxiquement.

La validation de la fonction de correspondance décrite dans le Chapitre VIII. utilise un corpus de documents transcriptions d'émissions radiophoniques. Les expérimentations montrent également une amélioration des résultats et confirment l'intérêt de considérer la notion de 'presque égalité' en l'intégrant, par le biais de l'appariement entre deux termes et à fortiori de la concordance et de l'intersection, dans la fonction de correspondance. Nous montrons également que la prise en compte des approximations de termes peut être vue comme une nouvelle fonction de lissage au sein des modèles de langue.

Le modèle étant validé expérimentalement, nous avons élaboré une interface pour un outil d'aide à la réunion téléphonique. Cet outil manipulant des documents issus de transcriptions de conversations téléphoniques, notre modèle s'intègre au système de recherche d'information de l'application.

Partie 4 : Vers une application

Cette partie présente une simulation d'outil d'aide à la réunion et montre l'intégration de notre système de recherche d'information adapté aux données incertaines dans ce contexte.

Un outil d'aide à la réunion téléphonique s'avère une application à même de répondre aux attentes des entreprises d'aujourd'hui.

En effet, les entreprises sont aujourd'hui de plus en plus nombreuses à se trouver en différents lieux par le biais notamment de la mondialisation et de la sous-traitance, la mise en place de système permettant à une véritable relation de travail collaboratif d'avoir lieu malgré l'éloignement devient une réelle nécessité.

De même, les entreprises virtuelles c'est-à-dire sans mur et/ou permettant le travail à domicile s'avèrent en pleine expansion. Ces nouvelles façons de travailler ensemble engendrent la nécessaire mise en place d'un environnement pour entreprise virtuelle.

C'est pourquoi il devient essentiel d'inventer un espace d'expression et d'échange permettant le travail collaboratif. Une telle plate-forme doit permettre une collaboration "naturelle" tout en offrant des moyens nouveaux pour la construire de manière efficace.

Les échanges entre les différents utilisateurs d'une telle plateforme peuvent s'effectuer par le biais de documents écrits ou oraux.

De nombreuses applications deviennent alors possibles, comme notamment un outil d'aide à la réunion téléphonique. Dans une telle configuration, d'un point de vue fonctionnel, la recherche d'information, intervient « avant la réunion », « pendant la réunion » et « après la réunion ». Il s'agit donc de fournir des fonctionnalités nouvelles et avancées lors de réunions téléphoniques. On imagine aisément les plus-values évidentes que peut fournir un tel outil lors du déroulement de réunions téléphoniques. Ces fonctionnalités peuvent également s'intégrer dans un logiciel de type collecticiel afin de permettre l'utilisation pleine et effective de ce nouvel environnement.

Dans cette partie, il ne s'agit donc pas de recréer un dispositif de type collecticiel mais bien de montrer comment compléter les outils existants en intégrant des fonctionnalités de recherche d'information telle que la recherche thématique et ceci par le biais d'un système de recherche d'information adapté aux données incertaines.

Nous n'en sommes qu'à l'étape préliminaire de la mise en place d'une application, à savoir l'élaboration d'une interface pour un outil d'aide à la réunion.

Chapitre IX. Elaboration d'une interface pour un outil d'aide à la réunion téléphonique

Parmi les applications existantes, nous portons attention à l'outil d'aide à la réunion 'Ferret Meeting Browser' développée par l'IDIAP[14] (cf. Figure 64). Cet outil permet un suivi textuel, temporel et visuel d'une réunion. Toutefois, aucune fonctionnalité de recherche d'information n'est disponible.

Figure 64. Démonstration de l'outil 'Ferret Meeting Browser'

A ce type d'outil nous voulons ajouter des fonctionnalités permettant à un travail collaboratif d'avoir lieu durant la réunion et non simplement a posteriori comme le fait l'application citée précédemment. De ce fait, cette interface sert de base à notre application.

1. Un outil d'aide à la réunion téléphonique

Dans le cadre de la mise en place d'un outil d'aide à la réunion téléphonique, un certain nombre de fonctionnalités nécessitant un système de recherche d'information adapté aux données incertaines ressortent. Nous présentons ici un fonctionnement possible d'un outil d'aide à la réunion téléphonique (cf. Figure 65). A partir des transcriptions textuelles de la

[14] IDIAP : Institut Dalle Molle d'Intelligence Artificielle Perceptive, http://www.idiap.ch

conversation fournies par le système de reconnaissance automatique de la parole, les participants à la réunion accèdent à un suivi de l'ordre du jour. Cette fonctionnalité réside de l'utilisation d'un suivi thématique (1) nécessitant des outils de recherche d'information adaptés aux données incertaines. De même, une recherche documentaire (2) adaptée aux données incertaines s'avère nécessaire pour fournir des documents en rapport avec la réunion durant son déroulement.

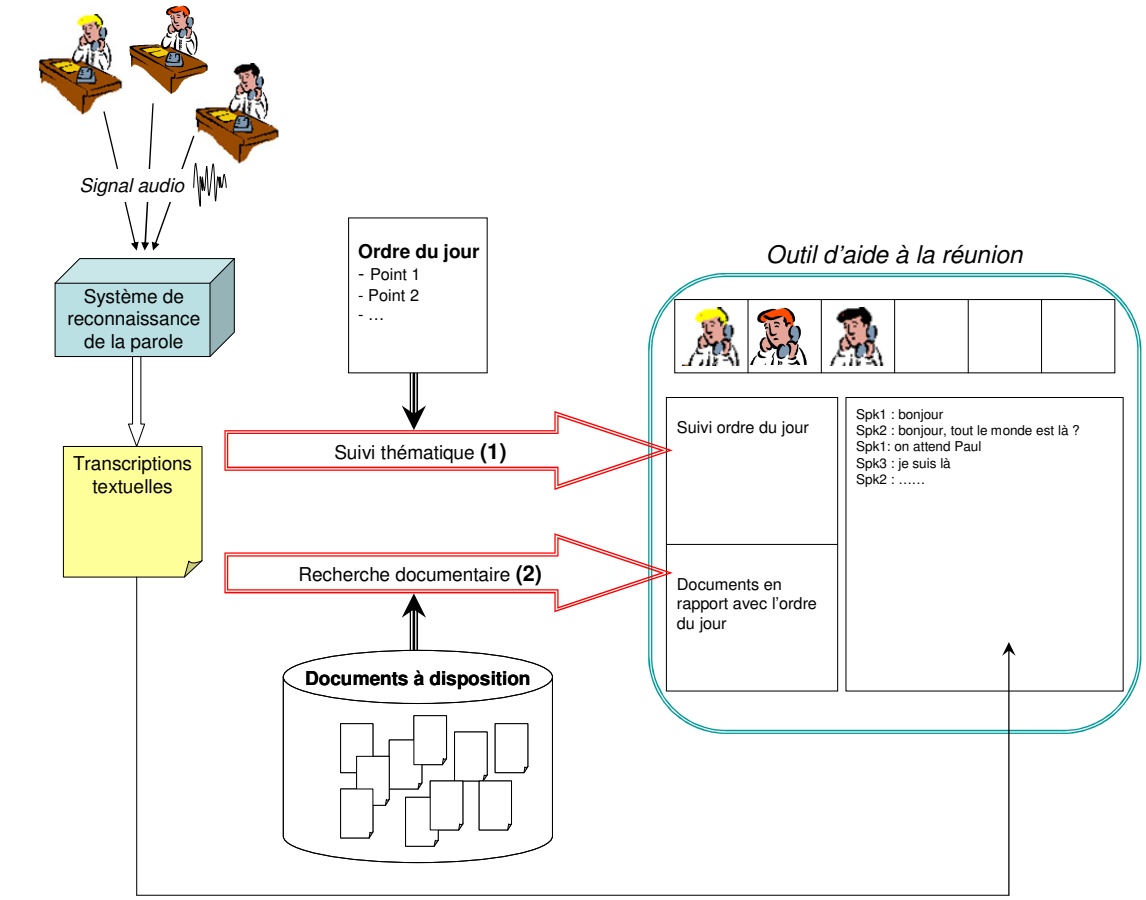


Figure 65. Principe de l'outil d'aide à la réunion téléphonique

Cette partie a pour but de présenter l'intérêt de la mise en place d'un outil d'aide à la réunion téléphonique en donnant une vue d'ensemble des fonctionnalités possibles.

2. L'outil d'aide à la réunion téléphonique et les données incertaines

Les transcriptions automatiques de conversation étant des données incertaines, un système de recherche d'information adapté aux données incertaines s'avère nécessaire dans la mise en place d'un outil d'aide à la réunion. Nous détaillons le principe de trois tâches de recherche d'information nécessitant la prise en compte des données incertaines.

2.1. Fournir automatiquement des documents en rapport avec la conversation

Le but de cette fonctionnalité consiste à fournir des documents au fil de la conversation (cf. Figure 66). Les documents sont en rapport avec le sujet en train d'être abordé. Ils proviennent du Web ou d'une base de données prédéfinies. Cette tâche correspond à une classique tâche en recherche d'information consistant, à partir d'un besoin d'information, de fournir à l'utilisateur un ensemble de documents pertinents. Toutefois, la requête n'est pas formulée directement en langage naturel ou par le biais de mots-clés mais extraite de la conversation en cours. La requête issue des transcriptions automatiques de la conversation par le système ASR contient donc des données incertaines (1). Les données contenues dans les documents sont quant à elles certaines (2). Dans ce contexte, le système de recherche d'information adapté aux données incertaines s'avère approprié.

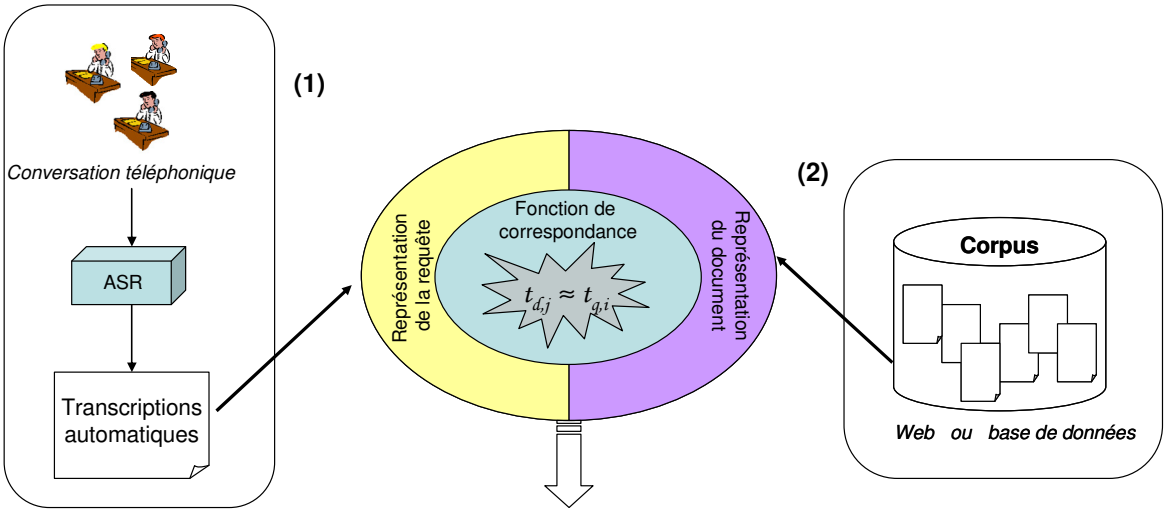


Figure 66. Fournir des documents à l'utilisateur

2.2. Suivi de l'ordre du jour

Une autre possibilité de l'application réside dans le suivi de l'ordre du jour (cf. Figure 67). Dans cette configuration, ce sont les documents qui contiennent des données incertaines. En effet, les points de l'ordre de jour correspondent à autant de requêtes (1). Les requêtes utilisent les mots de l'énoncé du point de l'ordre du jour tel quel ou bien en les enrichissant. Quant aux documents, ils se forment de parties de la conversation (2). Ici encore, un système de recherche d'information adapté aux données incertaines s'avère nécessaire pour prendre en compte l'incertitude existant dans les documents.

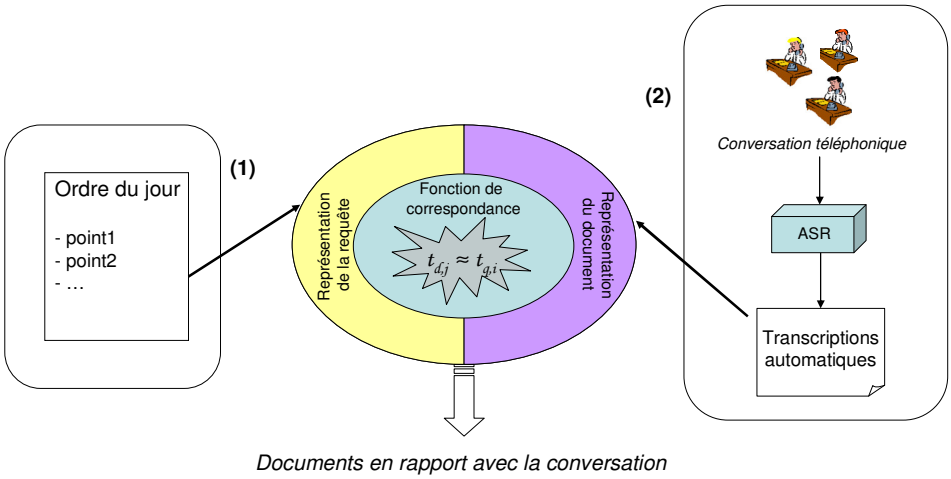


Figure 67. Suivi de l'ordre du jour

2.3. Moteur de recherche de réunions

Cette fonctionnalité s'utilise a posteriori des réunions. Elle permet à un utilisateur de retrouver une réunion (ou partie de réunion) traitant d'un sujet particulier. Il exprime sa requête à l'oral. De ce fait, requête et documents contiennent des données incertaines puisque la requête utilisée par le système de recherche d'information provient de la transcription de la parole et les documents des transcriptions des réunions. Là encore, l'utilisation d'un système de recherche d'information adapté aux données incertaines prend tout son sens.

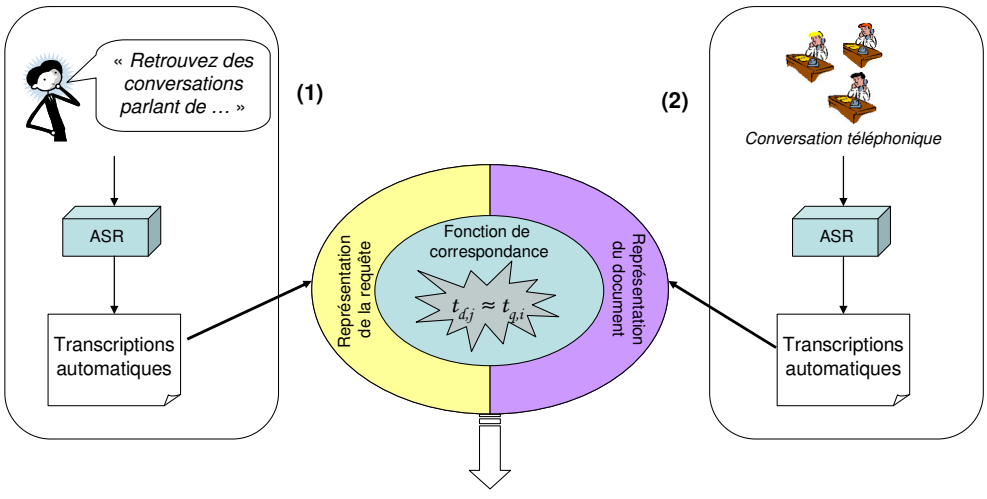


Figure 68. Moteur de recherche de réunions

Afin de montrer ces fonctionnalités, nous avons développé une application les mettant en œuvre. Certains aspects demeurant à l'état de travail préliminaire s'avèrent simulés. Nous commençons par énumérer les possibilités de l'application puis nous détaillons chaque partie de l'environnement réalisé.

3. Contexte de développement

A partir d'enregistrements de réunions téléphoniques et des transcriptions correspondantes, nous développons un logiciel de simulation de réunion téléphonique.

De manière statistique, l'ensemble des participants à la réunion est visible dans l'application, de même que l'ordre du jour et les documents mis à disposition des orateurs.

Une synchronisation entre l'enregistrement audio de la réunion, la transcription de celle-ci et la mise en évidence du locuteur s'établit de manière dynamique. Au fil de la conversation, les temps de parole relatif et absolu des participants se mettent à jour. Par temps de parole relatif, on entend le temps de parole d'un locuteur par rapport à l'ensemble des temps de parole et par temps de parole absolu, la durée de prise de parole. On retrouve ces temps relatif et absolu pour la réunion permettant de voir si une réunion prend du retard par rapport à la durée prévisionnelle.

Au niveau des tâches de recherche d'information, on retrouve le suivi de l'ordre du jour, un filtrage des documents en rapport avec la conversation et un moteur de recherche des réunions (ou passage de réunions).

4. Environnement réalisé

Ce paragraphe présente l'interface de l'application afin d'illustrer les fonctionnalités de celle-ci précédemment décrites.

L'environnement réalisé est composé de quatre parties principales (cf. Figure 69) :

- (1) Visualisation des participants et de la personne ayant la parole
- (2) Suivi de l'ordre du jour et documents à disposition
- (3) Suivi textuel de la conversation
- (4) Un espace de travail

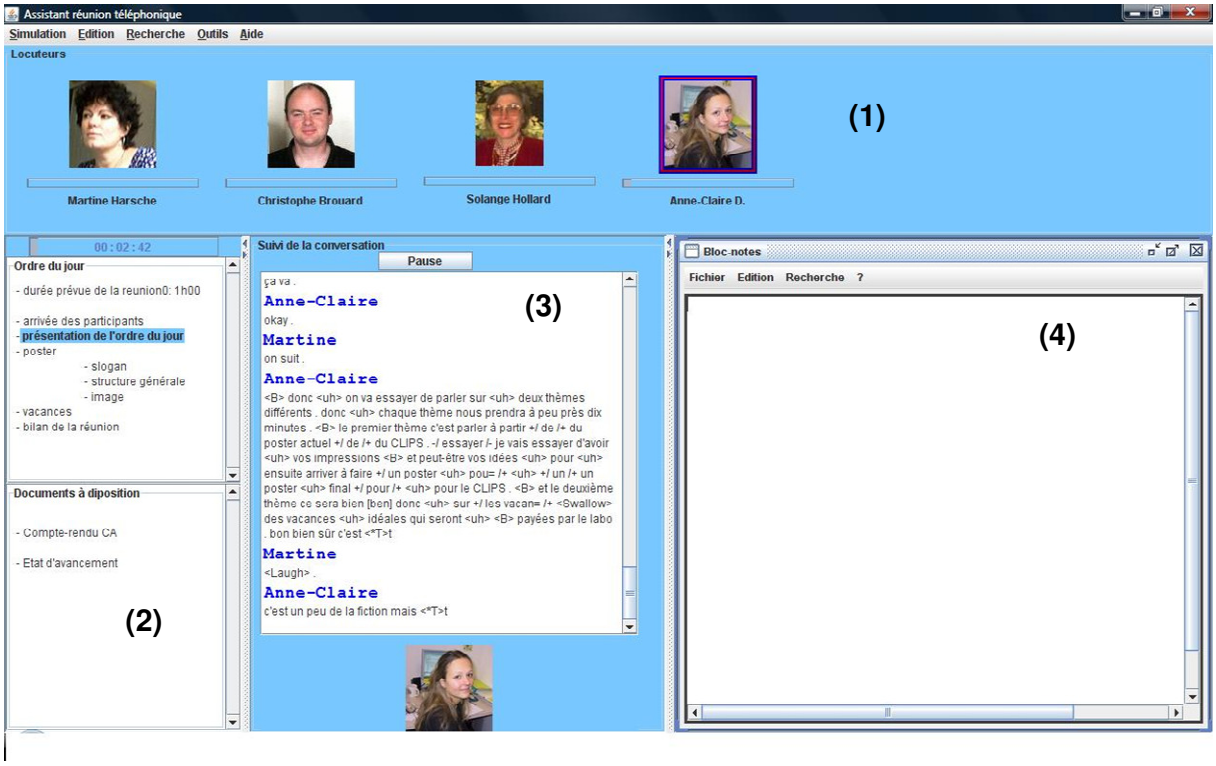


Figure 69. Interface de l'assistant de réunion téléphonique

4.1. Zone 'locuteurs'

Cet espace fixe peut contenir jusqu'à six photos et représente les participants de la réunion (cf. Figure 70).

Figure 70. Zone 'locuteurs'

Il permet la mise en évidence du locuteur par un encadrement de sa photo et la présence ou non d'un participant par un grisé de la photo. Le temps de parole de chaque personne apparaît également dans cette zone. De plus, une description de la personne apparaît dans la zone de travail lors d'un clic sur sa photo.

4.2. Zones 'ordre du jour' et 'documents à disposition'

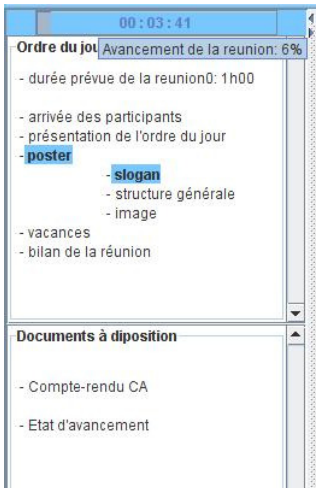


Figure 71. Zone 'ordre du jour' et 'documents à disposition'

Cet espace de travail (cf. Figure 71) permet un suivi de l'ordre du jour avec une mise en évidence du point abordé, un accès aux différents documents à disposition et un suivi de la durée de la réunion.

4.3. Zone 'suivi de la conversation'

Cette zone (cf. Figure 72) fournit un suivi de la réunion par le biais de la transcription textuelle.

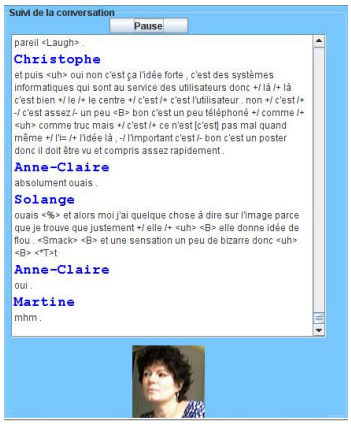


Figure 72. Zone 'suivi de la conversation'

Cet espace se compose d'une zone de déroulement de la transcription au fil de la conversation, d'une photo du locuteur et d'un bouton lecture/pause.

4.4. Zone 'zone de travail'

Cet espace (cf. Figure 73) fonctionne selon le même principe que le bureau d'un ordinateur. Cette partie de l'application permet une utilisation de différents outils tels que le bloc-notes ou l'explorateur Internet. La visualisation des documents s'effectue également dans cette zone de l'application.

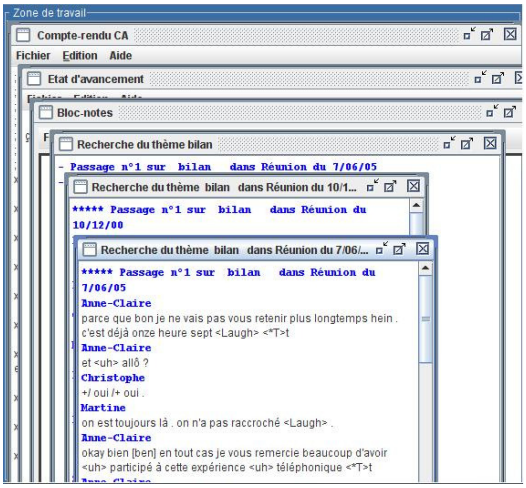


Figure 73. Zone 'zone de travail'

5. Recherche thématique

L'application permet à posteriori une recherche thématique sur les réunions par le biais d'un moteur de recherche de réunions. Ainsi l'utilisateur a la possibilité de chercher :

- Soit un thème dans l'ensemble des réunions ayant eu lieu (cf. Figure 74)
- Soit un point de l'ordre du jour choisi dans une réunion particulière (cf. Figure 75).

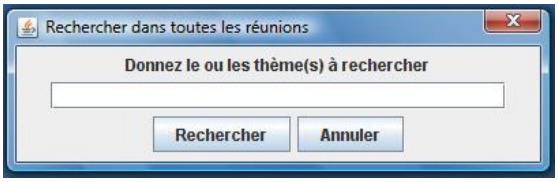


Figure 74. Recherche d'un thème

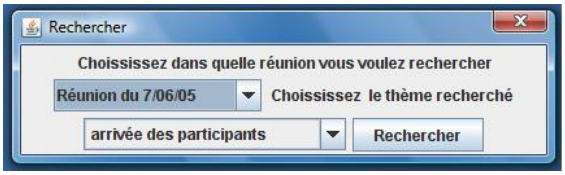


Figure 75. Recherche d'un point de l'ordre du jour dans une réunion

Pour fonctionner de manière optimale, la recherche thématique doit intégrer la notion de données incertaines puisque les systèmes de reconnaissance de la parole fournissent des transcriptions incertaines.

6. Conclusion

Le développement de cette application d'aide à la réunion demeure encore au stade préliminaire. Aussi, cette partie a pour unique but de montrer un cadre applicatif possible du système de recherche d'information adapté aux données incertaines que nous proposons dans notre thèse.

Une telle application illustre l'utilité de notre système, les données utilisées correspondant à des transcriptions automatiques de conversations traitées par un système de reconnaissance de la parole. Nous avons énuméré un certain nombre de tâches de recherche d'information nécessitant l'intégration de la notion d'incertitude à savoir fournir des documents en rapport avec un thème, suivre un ordre du jour préétabli et la mise en place d'un moteur de recherche de réunions.

Bilan et perspectives

1. Synthèse et contribution

Pour trouver l'ensemble des documents répondant à une requête, tout système de recherche d'information développe une méthodologie formelle ou opérationnelle pour affirmer si les termes de chaque document correspondent à ceux de la requête de l'utilisateur.

La plupart des systèmes s'appuie sur l'hypothèse que les termes extraits des documents ont été parfaitement reconnus ou identifiés, et de fait leur fonction de correspondance se base sur une capacité à disposer d'une relation d'égalité entre termes du document et termes de la requête (cf. Figure 76).

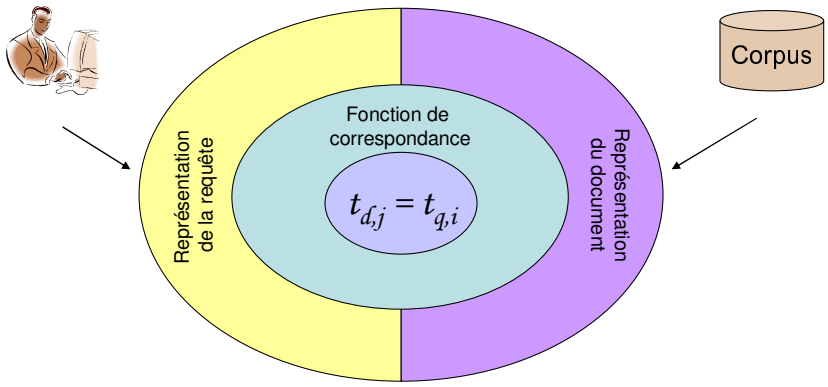


Figure 76. Un système de recherche d'information centré sur l'égalité des termes

Dans certains contextes l'extraction des termes est source d'incertitude et de fait le système dispose de données incertaines. Les processus d'interprétation de données commettant des erreurs se regroupent sous le nom de processus d'interprétation incertain.

La notion de données incertaines désigne l'ensemble des données issues d'un processus d'interprétation incertain et associées à un coefficient de certitude. Les données interprétées deviennent alors des données incertaines (cf. Figure 77).

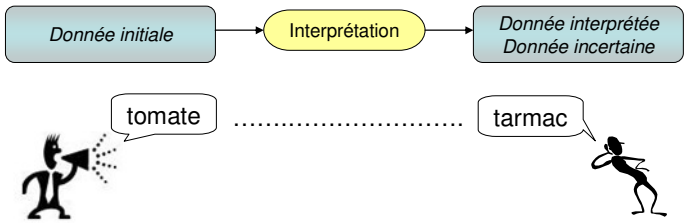


Figure 77. Exemple de données incertaines

La présence d'incertitude au sein des données prises en compte par le système de recherche d'information engendre la remise en cause de la base du système d'information qui ne s'avère plus valide (cf. Figure 78). Ainsi, à la simple égalité $t_{d,j} = t_{q,i}$ se substitue la presque égalité $t_{d,j} \approx t_{q,i}$. La présence de cette presque égalité s'explique par l'existence d'erreurs dans les données incertaines, erreurs commises lors de l'interprétation des données initiales.

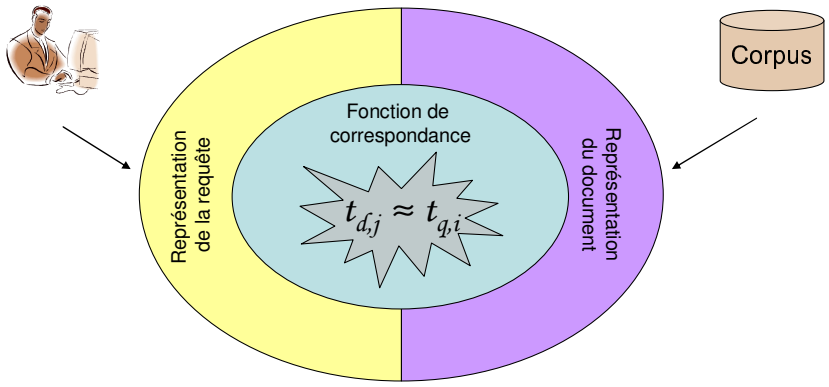


Figure 78. Remise en cause de l'égalité au centre d'un système de recherche d'information

Notre travail de thèse propose un système de recherche d'information prenant en compte la dimension d'incertitude. La présence de la 'presque égalité' remettant en cause la fonction de correspondance et la représentation des documents, nous proposons de les adapter au contexte incertain.

La contribution de notre travail réside dans la proposition d'un système de recherche d'information adapté aux données incertaines qui associe une fonction de pondération intégrant la notion de certitude des termes et une fonction de correspondance basée sur le modèle de langue et sur l'appariement des termes. Nous introduisons une nouvelle mesure : l'appariement, qui permet d'évaluer le taux de 'presque égalité' entre deux termes par le biais de la concordance et l'intersection. La concordance définit le positionnement relatif entre deux termes et l'intersection mesure la proximité entre les zones communes aux deux termes.

A travers des expérimentations basées sur des données issues de la campagne d'évaluation de CLEF-2004 pour la fonction de pondération et provenant de la campagne ESTER pour la fonction de correspondance, nous validons expérimentalement notre système de recherche d'information. Ces expérimentations mettent en avant l'intérêt de l'intégration de l'incertitude dans un système de recherche d'information traitant des données incertaines.

Enfin, nous proposons un cadre applicatif nécessitant l'utilisation d'un système de recherche d'information adapté aux données incertaines : un outil d'aide à la réunion téléphonique. Cette première ébauche d'application illustre l'intérêt d'intégrer la notion d'incertitude au sein des tâches de recherche d'information.

2. Perspectives

Comme toute proposition de thèse, notre travail ouvre de nombreuses perspectives de recherche.

Dans ce paragraphe, nous citons les perspectives de travail à court et plus long terme nous semblant les plus intéressantes.

2.1. A court terme

Validation globale du modèle

Il s'avère difficile de trouver des collections tests composées de transcriptions automatiques avec des valeurs de certitude associées à chaque terme extrait par le système de reconnaissance, des transcriptions manuelles correspondantes et des requêtes résolues. Compte tenu de cette difficulté, la validation de notre modèle se fait en deux temps : validation de la fonction de pondération et validation de la fonction de correspondance. A court terme, nous souhaitons disposer d'un corpus permettant une évaluation globale du système.

Etude des valeurs de concordance et intersection

Nous souhaitons effectuer une étude plus poussée des valeurs associées à chaque type de concordance et d'intersection. Nous posons la question de l'importance à donner à chaque type de concordance et d'intersection. Ainsi, quel rapport de valeur doit-il y avoir entre une concordance de type *concorde* et une de type *chevauche* ?

2.2. A plus long terme

Développement de l'outil d'aide à la réunion téléphonique

Les fonctionnalités de l'application nécessitent un certain nombre de prétraitements des transcriptions de réunions.

Tout d'abord, pour la tâche fournissant des documents pertinents au fil de la conversation, une extraction des mots-clés s'avère nécessaire. Pour déterminer le choix des termes, certaines catégories morphosyntaxiques de mots sont plus porteuses de sens que d'autres. Les points de l'ordre du jour apportent également un indice.

La tâche de suivi de l'ordre du jour engendre un découpage en passages de la réunion, découpage utile également pour le moteur de recherche de réunions. Ce travail se base sur un découpage thématique. Différentes méthodes permettent une segmentation thématique d'un document [Callan, 1994], [Kaszkiel, 1997], [Moffat, 1994], [Salton, 1993]. Nous nous positionnons dans une optique de suivi thématique positif et non négatif tel que le fait

un système de TextTiling [Hearst, 1993]. En effet, le TextTiling cherche les ruptures de thèmes et les identifie lorsqu'un passage du document présente un moins grand nombre de mots traitant du thème. Nous proposons de privilégier le suivi thématique positif, c'est-à-dire de considérer qu'il y a un changement de thème lorsque nous rencontrons une forte densité de mots considérés comme des mots de rupture. Ceci est possible grâce à la présence d'un discours organisationnel propre aux conversations téléphoniques. Nous montrons l'intérêt d'une telle approche dans [Tambellini, 2004]. Nous souhaitons donc intégrer cette approche à notre modèle de recherche d'information adapté aux données incertaines.

Le découpage des conversations enregistrées en passage s'avère également nécessaire pour le moteur de recherche de réunions. Ainsi un utilisateur peut retrouver un passage de la réunion traitant d'un point particulier.

Enfin, il nous parait essentiel de penser à une indexation efficace commune aux trois tâches de recherche d'information évoquées dans l'application : fournir des documents pertinents en rapport avec la conversation, suivi de l'ordre du jour et moteur de recherche de réunions a posteriori, la multiplication des indexations de documents diminuant l'efficacité du système.

Nous envisageons donc de poursuivre le développement de l'outil d'aide à la réunion téléphonique. En effet, l'application étant encore au stade préliminaire, nous souhaitons y intégrer notre modèle afin d'effectuer des tâches de recherche d'information telle que la recherche thématique ou le résumé automatique. Ceci devrait être fait assez rapidement, compte tenu du fait que nous disposons déjà de l'interface graphique.

Extension du modèle aux bi-grammes

Nous désirons étendre notre modèle, basé sur le modèle de langue, aux bi-grammes. Cette extension permettrait la prise en compte d'autres types d'erreurs. On peut citer par exemple, pour l'oral, les liaisons mal interprétées par le système de reconnaissance de la parole : « il est talonneur » au lieu de « il est à l'honneur ».

Cette prise en compte nécessite une reformulation des types de concordance actuellement entre uni-grammes. Nous pensons qu'une combinaison uni-grammes et bi-grammes s'avère plus judicieuse qu'une simple considération des bi-grammes.

Extension du modèle à d'autres contextes

Nous voulons étendre le modèle proposé à d'autres contextes de travail. Nous pensons aux cas des synonymes pour des documents certains. Notre vision consiste à considérer les synonymes d'un terme comme des termes 'presque égaux'. Cet usage de la 'presque égalité' se rapproche des systèmes de recherche d'information basés sur les ontologies.

Annexes

1. Mesures

Précision moyenne

La précision moyenne *avgP* est définie comme suit :

$$avgP = \frac{1}{\left|H_q\right|} \sum_{n \in rel} P(n)$$

où *rel* est l'ensemble des positions dans l'ordonnancement des documents pertinents,

n est le rang du document

P(n) la précision du document au rang n

q est une requête donnée

H_q est l'ensemble de documents réellement pertinents.

BEP (Break-even point)

Le *BEP* est une autre mesure d'évaluation. Rappel et précision sont calculés à la position $|H_q|$ dans l'ordonnancement, où $|H_q|$ est le nombre de documents dans l'ensemble H_q. A cette position, *P(n)* et *R(n)* sont égaux et leur valeur est appelée *BEP*. *BEP* peut être vu comme l'intersection de la courbe rappel/précision et l'axe où R=P.

2. Les algorithmes phonétiques

(Extrait de : http://www-lium.univ-lemans.fr/~carlier/recherche/soundex.html#L3)

Soundex

Voici l'algorithme original de Russel & O'Dell datant de 1918.

- Retranscrire le mot en majuscules
- Conserver la première lettre du mot
- Eliminer toutes les voyelles, le H et le W
- Transcoder les lettres restantes à l'aide de la table suivante (cf. Tableau XVIII pour l'anglais et Tableau XIX pour le français)

Lettre	Type de consonnance	code
B F P V	Bilabiales	1
C G J K Q S X Z	Labiodentales	2
D T	Dentales	3
L	Alvéolaires	4
M N	Vélaires	5
R	Laryngales	6

Tableau XVIII. Tableau de correspondance pour les Soundex anglais

Lettre	Type de consonnance	code
B P	Bilabiales	1
C K Q	Labiodentales	2
D T	Dentales	3
L	Alvéolaires	4
M N	Vélaires	5
R	Laryngales	6
G J	Labiodentales	7
S X Z	Labiodentales	8
F V	Bilabiales	9

Tableau XIX. Tableau de correspondance pour les Soundex français

- Eliminer toutes les paires consécutives de chiffres dupliquées
- Ne conserver que 4 caractères du Soundex et compléter par des zéros le cas échéant

Soundex2

Soundex2 correspond à l'algorithme du Soundex francisé.

L'algorithme du Soundex2 est le suivant :

- Éliminer les blancs à droite et à gauche du nom
- Convertir le nom en majuscule
- Convertir les lettres accentuées et le c cédille en lettres non accentuées
- Eliminer les blancs et les tirets
- Remplacer les groupes de lettres suivantes par leur correspondance, en conservant l'ordre du tableau (cf. Tableau XX) :

GUI	KI
GUE	KE
GA	KA
GO	KO
GU	K
CA	KA
CO	KO
CU	KU
Q	K
CC	K
CK	K

Tableau XX. Correspondance des groupes de lettres pour Soundex2

- Remplacer toutes les voyelles sauf le Y par A exceptée s'il y a un A en tête
- Remplacer les préfixes par leur correspondance (cf. Tableau XXI)

MAC	MCC	
ASA	AZA	(ASAmian)
KN	NN	(KNight)
PF	FF	(PFeiffer)
SCH	SSS	(SCHindler)
PH	FF	(PHilippe)

Tableau XXI. Tableau de correspondance des préfixes pour Soundex2

- Supprimer les H sauf s'ils sont précédés par C ou S
- Supprimer les Y sauf s'il est précédé d'un A
- Supprimer les terminaisons suivantes A, T, D et S
- Enlever tous les A sauf le A de tête s'il y en a un
- Enlever toutes les sous chaînes de lettre répétitives
- Conserver les 4 premiers caractères du mot et si besoin le compléter avec des blancs pour obtenir 4 caractères

Phonex

Phonex est un algorithme de Soundex plus perfectionné encore que la version francisée de Soundex2. Phonex est optimisée pour le langage français.

L'algorithme Phonex est le suivant © *Frédéric BROUARD (31/3/99)* :

- Remplacer les y par des i
- Supprimer les h qui ne sont pas précédés de c ou de s ou de p
- Remplacer les couples 'ph' par f
- Remplacer les groupes de lettres suivantes :

gan	kan
gam	kam
gain	kain
gaim	kaim

- Remplacer les occurrences suivantes, si elles sont suivies par une lettre a, e, i, o, ou u :

ain	yn
ein	yn
aim	yn
eim	yn

- Remplacer les groupes de 3 lettres suivants :

eau	o
oua	2
ein	4
ain	4
eim	4
aim	4

- Remplacer le son 'é' :

é	y
è	y
ê	y
ai	y
ei	y
er	yr
ess	yss
et	yt

- Remplacer les groupes de 2 lettres suivantes (son 'an' et 'in'), sauf s'ils sont suivis par une lettre a, e, i o, u ou un son 1 à 4 :

an	1
am	1
en	1
em	1
in	4

- Remplacer le son "on"

on	1

- Remplacer les s par des z s'ils sont suivis et précédés des lettres a, e, i, o,u ou d'un son 1 à 4
- Remplacer les groupes de 2 lettres suivants :

oe	e
eu	e
au	o
oi	2
oy	2
ou	3

- Remplacer les groupes de lettres suivants

ch	5
sch	5
sh	5
ss	s
sc	s

- Remplacer le c par un s s'il est suivi d'un e ou d'un i
- Remplacer les lettres ou groupe de lettres suivants :

c	k
q	k
qu	k
gu	k
ga	ka
go	ko
gy	ky

- Remplacer les lettres suivantes :

a	o
d	t
p	t
j	g
b	f
v	f
m	n

- Supprimer les lettres dupliquées
- Supprimer les terminaisons suivantes : t, x
- Affecter à chaque lettre le code numérique correspondant en partant de la dernière lettre

0	1
1	2
2	3
3	4
4	5
5	e
6	f
7	g
8	h
9	i
10	k
11	l
12	n
13	o
14	r
15	s
16	t
17	u
18	w
19	x
20	y
21	z

– Convertir les codes numériques ainsi obtenus en un nombre de base 22 exprimé en virgule flottante.

3. Distance de Hamming

(Définition extraite d'un article du site http://www.techno-science.net/)

Définitions

Soit A un alphabet et F l'ensemble des suites de longueur n à valeur dans A. La distance de Hamming entre deux éléments a et b de F est le cardinal de l'ensemble des images de a qui diffèrent de celle de b.

Formellement, si $d(.,.)$ désigne la distance de Hamming :

$$\forall a, b \in \mathcal{F} \quad a = (a_i)_{i \in [0,n-1]} \text{ et } b = (b_i)_{i \in [0,n-1]}, \quad d(a,b) = \#\{i : a_i \neq b_i\}$$

La notation $\#E$ désigne le cardinal de l'ensemble E.

Un cas important dans la pratique est celui des symboles binaires. Autrement dit $A = \{0,1\}$, On peut alors écrire, si $\oplus$ désigne le ou exclusif.

$$d(a,b) = \sum_{i=0}^{n-1} a_i \oplus b_i$$

Dans le cas, fréquent, où l'alphabet est un corps fini, F possède une structure d'espace vectoriel de dimension n. La distance dérive alors d'une pseudo-norme :

Soit K est un corps fini et F l'ensemble des suites de longueur n à valeur dans K. Le poids de Hamming $p(a)$ d'un élément a de F est le cardinal de l'ensemble des images de a non nulles.

L'alphabet est souvent F_2 le corps à deux éléments $\{0,1\}$. Le poids de Hamming est une pseudo-norme car :

$$\forall a \in \mathcal{F}, \forall \lambda \in \mathcal{K}, p(\lambda.a) = p(a)$$

Néanmoins, si l'alphabet est un corps fini, alors la distance dérive du poids de Hamming, en effet:

$$\forall a, b \in \mathcal{F}, d(a,b) = p(b - a)$$

Exemples

Considérons les suites binaires suivantes :

a = (0 0 0 1 1 1 1) et b = (1 1 0 1 0 1 1) alors d = 1+1+0+0+1+0+0 = 3

La distance entre a et b est égale à **3** car 3 bits diffèrent.

La distance de Hamming entre **1011101** et **1001001** est 2.

La distance de Hamming entre **2143896** et **2233796** est 3.

La distance de Hamming entre "**ramer**" et "**cases**" est 3.

4. Pondération des termes

Fonction générale de pondération

Une fonction de pondération attribue à chaque terme *t* de chaque document *d* une valeur *w*. Ce poids se calcule en tenant compte de deux grands critères : la force locale *fL* du terme *t* dans *d* et la force globale *fG* de *t* dans le corpus *CO* :

$$w = F\left(fL(t,d), fG(t,CO)\right)$$

La force locale d'un terme dans un document *fL(t,d)* mesure l'importance de ce terme dans le document. La force globale *fG(t,CO)* d'un terme mesure son importance dans le corpus.

Schématiquement, plus un terme est présent dans un document, plus sa force locale *fL* est importante. Plus ce terme est présent dans le corpus, plus sa force globale *fG* s'avère élevée.

fL et *fG* doivent être combinées de manière à évaluer au mieux le poids d'un terme *t* dans un document *d* : la fonction *F* assure la combinaison des deux forces connues pour *t*. *F* doit assurer que plus la force locale d'un terme *t* dans un document *d* est forte, plus *w* doit être élevé, mais aussi que plus la force globale de *t* est élevée, plus *w* doit être faible.

De fait, le poids d'un terme *t* dans un document *d* se calcule généralement par le produit d'une force locale *fL(t,D)* et de l'inverse d'une fonction globale, appelée *ifG(t, CO)* :

$$w = fL(t,d) * ifG(t,CO)$$

Pour définir la force locale et la force globale d'un terme, [Salton, 1971] [Saltonb, 1983] [Salton, 1988] proposent plusieurs fonctions possibles présentées dans la suite.

Force locale : importance du terme dans un document

Comme nous l'avons vu précédemment, la force locale d'un terme s'avère un critère important dans la détermination du poids *w* d'un terme dans un document. Pour représenter cette valeur, le nombre d'occurrences du terme *t* dans le document *d tf(t,d)* demeure communément utilisé.

Nous donnons ici quelques une des fonctions locales les plus couramment utilisées :

- $fL_1(t,d) = tf(t,d)$
- $fL_2(t,d) = \frac{tf(t,d)}{Max\left(tf(t',d)\right)}$

où $Max\left(tf(t',d)\right)$ est la fréquence maximale de l'ensemble des termes de *d*

- $fL_3(t,d) = \log\left(tf(t,d)\right)$
- $fL_4(t,d) = \log\left(tf(t,d)+1\right)$

Inverse de fonction globale : le pouvoir discriminant du terme

Nous avons vu précédemment que la force globale d'un terme tient un rôle important dans la définition de la fonction du poids d'un terme : plus un terme a une force globale élevée, plus le poids de ce terme doit être atténué dans les documents. En effet, un terme souvent présent dans un document et peu présent dans les autres documents sera plus discriminant pour un document qu'un terme apparaissant le même nombre de fois dans le document mais étant présent dans beaucoup d'autres documents. Ceci montre l'intérêt d'utiliser une mesure du pouvoir discriminant (ou non uniformément distribué) du terme dans le corpus de documents.

Cette mesure *ifG* se base souvent sur *df(t)* (document frequency). D'autres méthodes telles que le ratio signal-bruit ou la valeur de discrimination du terme peuvent être utilisées.

- $ifG_1(t,CO) = \log\left(\frac{N}{df(t)}\right)$
- $ifG_2(t,CO) = discvalue(t)$ = « term discrimination value » mesure via une mesure de similarité entre les documents combien l'utilisation d'un terme augmente (faible discrimination) ou diminue (forte discrimination) la similitude des documents.

$ifG_2(t,CO) = AVGSIM_t - AVGSIM$

Avec $AVGSIM = cte\sum_{i=1,i\neq j}^{N}\sum_{j=1}^{N} similarité(d_i,d_j)$, (par exemple, $cte = \frac{1}{N(1-N)}$)

- $ifG_3(t,CO) = signal(t)$

$signal(t) = \log(totfreq(t)) - noise(t)$ avec $noise(t) = \sum_{i=1}^{N} \frac{tf(t,d_i)}{totfreq(t)} \log \frac{totfreq(t)}{tf(t,d_i)}$

<u>*Pondérations des termes*</u>

Le poids *w* d'un terme *t* dans un document *d* se calcule comme le produit entre une force locale *fL(t,d)* et une force globale *ifG(t, CO)*. Les formules les plus classiques sont :

- $w_1 = fL_1(t,d) * (ifG_1(t,CO)+1) = tf(t,d) * \log\left(\frac{N}{df(t)}\right)$
- $w_2 = ifL_1(t,d) * ifG_2(t,CO) = tf(t,d) * discvalue(t)$
- $w_3 = ifL_1(t,d) * ifG_3(t,CO) = tf(t,d) * signal(t)$
- $w_4 = ifL_4(t,d) * ifG_1(t,CO) = \log(tf(t,d)+1) * \log\left(\frac{N}{df(t)}\right)$

5. Conversations téléphoniques

Extrait de transcription de conversation téléphonique

; CDR: 00.00

; TRV: 00.00

; File: Bg2

; Last changes made on 12/02/2002

; Transcriber: ACD

; Comments:

;

xxxxBG2_1_0077_ACD_00: super <Laugh> . <B> bon eh bien [bé] <uh> <hes> peut-être que le mieux ce serait de commencer +/ pa= /+ <uh> d'abord +/ par par /+ le haut +/ du /+ +/ du /+ du poster donc c'est-à-dire +/ le /+ le slogan . est-ce que vous auriez <uh> des commentaires ou +/ des /+ des suggestions , peut-être des critiques <uh> à faire ?

xxxxBG2_1_0078_EM_00: mhm .

xxxxBG2_1_0079_BM_00: <uh> alors moi j'ai +/ peut-être une /+ <uh> peut-être une suggestion avant c'est , je voudrais savoir à qui est destiné le poster ? parce que suivant +/ la /+ la cible visée il n'aura +/ pas le même /+ <uh> pas le même discours .

xxxxBG2_1_0080_ACD_00: mhm .

xxxxBG2_1_0081_JCD_00: ouais .

xxxxBG2_1_0082_EM_00: mhm .

xxxxBG2_1_0083_ACD_00: oui .

xxxxBG2_1_0084_EM_00: <%> c'est vrai ouais .

xxxxBG2_1_0085_ACD_00: +/ ouais /+ ouais c'est vrai .

xxxxBG2_1_0086_BM_00: donc est-ce qu'il est grand public ou est-ce qu'il est <uh> plutôt à un public <uh> averti ? et peut-être que ça orientera <uh> le <*T>t

xxxxBG2_1_0087_JCD_00: mhm .

xxxxBG2_1_0088_ACD_00: oui bien sûr ouais . c'est vrai que là par exemple , celui-là -/ il est vraiment /- il a été fait pour un grand public <B> mais <uh> bon a priori ce serait plus pour faire +/ un /+ un vrai poster +/ bien /+ bien sérieux bon à présenter +/ aux /+ aux autres équipes , à d'autres labos <uh> internationaux ou nationaux donc <uh> <B> ce serait plus +/ un /+ un public averti .

xxxxBG2_1_0089_JCD_00: mhm .

xxxxBG2_1_0090_JCD_00: mhm .

xxxxBG2_1_0091_BM_00: d'acco= <*T>t

xxxxBG2_1_0092_EM_00: ah oui d'accord . parce que +/ dans la /+ +/ dans nos /+ dans nos consignes <uh> disons , on disait que ce poster c'était pour <uh> présenter le CLIPS aux futures conférences ou fêtes de la science ou colloques donc là +/ ça /+ c'est trop large là .

xxxxBG2_1_0093_JCD_00: +/ oui /+ oui c'est incompatible .

xxxxBG2_1_0094_ACD_00: <B> oui mais pourquoi <*T>t

xxxxBG2_1_0095_JCD_00: ouais .

xxxxBG2_1_0096_ACD_00: tu penses ? +/ pourquoi /+ pourquoi <*T>t

xxxxBG2_1_0097_EM_00: +/ non /+ non mais si tu dis <uh> ce qui est très juste <uh> que c'est plutôt orienté vers un public averti donc ça cadre bien avec conférence , colloque . par contre fête de la science bon c'est vrai que c'est peut-être un petit peu plus grand public quoi <%> <*T>t

xxxxBG2_1_0098_BM_00: voilà .

xxxxBG2_1_0099_ACD_00: mhm .

xxxxBG2_1_0100_BM_00: tout à fait <%> <*T>t

xxxxBG2_1_0101_ACD_00: <B> mais pourquoi -/ on ne serait pas /- on ne pourrait pas être plus clair <uh> . je veux dire <uh> pourquoi on ne serait pas capable de faire +/ un /+ <hm> <B> un poster +/ qui /+ qui puisse <uh> coller +/ aux /+ aux deux cas de figure .

Statistiques sur les mots pour chaque conversation

Dialogues	Nombre de mots	Nombre de tours de parole	Nombre de mots par tour de parole
Brainstorming1	8373	801	10,45
Brainstorming2	7892	991	7,96
Brainstorming3	8261	927	8,91
Brainstorming4	7226	993	7,28
Brainstorming5	6083	562	10,82
Entretien1	2793	207	13,49
Entretien2	1552	138	11,25
Entretien3	2304	171	13,47
Entretien4	2687	185	14,52
Réun. Projet 1	9142	723	12,64

Réun. Projet 2	7671	470	16,32
Réun. Projet 3	13228	1024	12,92
Réun. Projet 4	9079	718	12,64

Rappel et précision pour différents seuils en utilisant le vocabulaire de rupture 1 (les mots d'approbation + les mots de changement + les points d'interrogation)

	Seuil	pertinents	retrouvés	pert retrouvés	Rappel	Précision
Brainstorming 1	5	6	11	4	0,67	0,36
	6	6	8	3	0,50	0,38
	7	6	6	2	0,33	0,33
Brainstorming 2	4	4	15	2	0,50	0,13
	5	4	6	2	0,50	0,33
	6	4	3	1	0,25	0,33
Brainstorming 3	6	6	9	3	0,50	0,33
	7	6	6	2	0,33	0,33
	8	6	4	2	0,33	0,50
Brainstorming 4	4	6	8	3	0,50	0,38
	5	6	3	1	0,17	0,33
	6	6	1	1	0,17	1,00
Brainstorming 5	4	4	14	1	0,25	0,07
	5	4	10	1	0,25	0,10
	6	4	3	1	0,25	0,33
Moyenne					0,37	0,35

6. Détail du rappel – précision des expérimentations sur la fonction de correspondance

Rappel	Transcription Automatique Sans approximation	μ = 0,5			
		sans seuil	0,2	0,4	0,6
0	0,6588	0.5958	0,6298	0,5965	0,6298
0,1	0,6388	0.5736	0,6098	0,5765	0,6098
0,2	0,6388	0.5736	0,5848	0,5765	0,6098
0,3	0,5912	0.5473	0,5848	0,5698	0,5765
0,4	0,5106	0.4583	0,4898	0,488	0,4968
0,5	0,5076	0.3930	0,4898	0,4784	0,4938
0,6	0,4936	0.3762	0,4446	0,4359	0,4455
0,7	0,4226	0.3027	0,3479	0,3588	0,3687
0,8	0,2844	0.2316	0,2521	0,2888	0,2519
0,9	0,1988	0.1773	0,1544	0,1494	0,1499
1	0,0928	0.1087	0,1261	0,0909	0,0919

Tableau XXII. Rappel précision avec μ = 0,5

Rappel	Transcription Automatique Sans approximation		μ = 0,8		
		sans seuil	0,2	0,4	0,6
0	0,6588	0,625	0,6609	0,6609	0,6609
0,1	0,6388	0,605	0,6409	0,6409	0,6409
0,2	0,6388	0,585	0,6409	0,6409	0,6409
0,3	0,5912	0,5717	0,5933	0,5933	0,5933
0,4	0,5106	0,5059	0,5143	0,5071	0,5126
0,5	0,5076	0,5059	0,5128	0,5057	0,5096
0,6	0,4936	0,4964	0,4894	0,4822	0,4895
0,7	0,4226	0,4185	0,4151	0,408	0,4135
0,8	0,2844	0,3113	0,2723	0,3116	0,2689
0,9	0,1988	0,2024	0,1867	0,1757	0,1762
1	0,0928	0,181	0,1257	0,0909	0,0919

Tableau XXIII. Rappel précision avec μ= 0,8

Rappel	Transcription Automatique Sans approximation		μ = 0,9		
		sans seuil	0,2	0,4	0,6
0	0,6588	0,6609	0,6609	0,6609	0,6609
0,1	0,6388	0,6409	0,6409	0,6409	0,6409
0,2	0,6388	0,6409	0,6409	0,6409	0,6409
0,3	0,5912	0,5933	0,5933	0,5933	0,5933
0,4	0,5106	0,5099	0,5197	0,5126	0,5126
0,5	0,5076	0,5084	0,5167	0,5096	0,5096
0,6	0,4936	0,4989	0,4966	0,4895	0,4961
0,7	0,4226	0,4227	0,4259	0,4187	0,4228
0,8	0,2844	0,315	0,2881	0,3274	0,2833
0,9	0,1988	0,2024	0,2089	0,1981	0,1986
1	0,0928	0,181	0,1257	0,0909	0,0919

Tableau XXIV. Rappel précision avec μ= 0,9

Liste des publications

✓ Revue nationale

- Loïc Maisonnasse, **Caroline Tambellini**, *Dépendances syntaxiques et méthodes de détection de passages pour une segmentation sur le locuteur et le thème*, revue RNTI Défi fouille de textes : reconnaissance automatique des auteurs de discours - Campagne DEFT'05 (TALN'05), parution mars 2007

✓ Conférence internationale avec comité de lecture

- **Caroline Tambellini**, *A language model which integrates uncertainty*, FDIA 2007, Glasgow, 2007

- **Caroline Tambellini**, Catherine Berrut, Christophe Brouard, *Information filtering and retrieval applied to delocalized meetings*, ECIR'04 Sunderland, 2004.

✓ Conférences nationales avec comité de lecture

- **Caroline Tambellini**, Catherine Berrut, Pondération des données incertaines dans les systèmes de recherche d'informations : une première approche expérimentale, INFORSID 2006, Hammamet, Tunisie, 1-4 juin 2006

- **Caroline Tambellini**, Catherine Berrut, Christophe Brouard, *Une Analyse préalable à l'indexation de transcriptions de conversations téléphoniques*, in CORIA'04, Toulouse, pp307-331, 10-12 mars , 2004.

✓ Conférence nationale sans comité de lecture

- Zohra Khalis, **Caroline Tambellini**, Loïc Maisonnasse, *A chaque corpus son découpage et une segmentation pour tous*, in DEFT 06, Fribourg, 2006

- Loïc Maisonnasse, **Caroline Tambellini**, *Dépendances syntaxiques et méthodes de détection de passages pour une segmentation sur le locuteur et le thème*, in DEFT 2005, TALN 2005 tome 2, Dourdan , pp155-164, 6-10 juin, 2005.

✓ Rapports

- **Caroline TAMBELLINI**, Extraction d'indices contextuels pour la recherche d'information dans des transcriptions d'entretiens téléphoniques, rapport de DEA, Groupe MRIM - CLIPS-IMAG, Juin, 2003.

Bibliographie

[Allen, 1981] Allen J.F., *A general model of action and time*, TR 97, University of Rochester, Departement of computer Science, 1981

[AUDIOSURF, 2001] AUDIOSURF : http://www.telecom.gouv.fr/rntl/AAP2001/Fiches_Resume/AUDIOSURF.htm,

[Boufaden, 2002] Boufaden, Lapalme, Bengio, *Découpage thématique des conversations : un outil d'aide à l'extraction*, TALN 2002, 24-27 juin 2002

[Boughanem, 2004] M. Boughanem, W. Kraaij, J.Y. Nie, *Modèles de langue pour la recherche d'informations*, Les systèmes de recherche d'informations - Modèles conceptuels, ed. M. Ihadjadene, Hermes, pp. 163-184, 2004

[Brouard, 1999] Brouard Frédéric, *http://sqlpro.developpez.com/cours/soundex/#L5*, 1999

[Brown, 1993] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer, *The Mathematics of Statistical Machine Translation: Estimation*, Computational Linguistics, Vol. 19(2), pp. 263--311, 1993

[Bruandet, 1997] M-F. Bruandet, J.P. Chevallet, F. Paradis, *Construction de thesaurus dans le systeme de recherche d'information IOTA : application a l'extraction de la terminologie*, in 1eres Journees Scientifiques et Techniques du Reseau Francophone de l'Ingerierie de la Langue de l'AUPELF-URF, Avignon, pp537-544, 15-16 Avril 1997

[Callan, 1994] Callan, *Passage-level evidence in document retrieval*, Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1994

[Celko, 1995] Joe Celko, *SQL avancé*, Thomson International Publishing, p. 85, 1995

[Chiaramella, 1986] Y. Chiaramella and B. Defude and M.F. Bruandet and D. Kerkouba, *IOTA: a full test information retrieval system*, in ACM conference on research and development in information retrieval, Pisa, Italy, pp. 207-213, 1986

[Cooper, 1968] W. Cooper, *Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems*, American Documentation, 19(1), pp. 30-41, 1968

[Croft, 1994] W.B. Croft, S.M. Harding, K. Taghva and J. Borsack, *An evaluation of information retrieval accuracy with simulated OCR output*, in Proceedings of the third annual symposium on document analysis and information retrieval, Las Vegas, NV, pp. 115-126, april 1994

[Garofolo, 2000] J.S. Garofolo, C.G. Auzanne, E.M. Voorhees, *The TREC Spoken Document Retrieval Track : a success story*, 2000

[Grangier, 2003] D. Grangier, A. Vinciarelli, H. Bourlard, *Information retrieval on noisy text*, september 2003

[Hamming, 1950] R. Hamming, *error-detecting and error-correcting codes Bell*, System Technical Journal 29(2), pp. 147-160, 1950

[Harman, 1993] D. Harman, *Overview of the TREC conference*, Proceedins of the 16th ACM SIGIR International Conference on Research and Development in Information Retrieval, pp 36-47, 1993

[Hearst, 1993] Marti A. Hearst, Christian Plaunt, *Subtopic structuring for full-length document access*, Proceedings of the Sixteenth Annual International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 59-68, 1993

[Hiemstra, 1998] Djoerd Hiemstra and Wessel Kraaij, *Twenty-One at TREC-7: Ad Hoc and Cross Language track*, TREC7, pp. 227-238, 1998

[Jelinek, 1997] Frederick Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, 1997

[Kaszkiel, 1997] Marcin Kaszkiel, Justin Zobel, *Passage retrieval revisited*, SIGIR, 1997

[Lopresti, 1996] Daniel Lopresti and Jiangying Zhou, *Retrieval strategies for noisy text*, in Proc. Symp. Document Analysis and Information Retrieval, pp. 255-270, 1996

[Luhn, 1957] H. Luhn, *A statistical approach to mechanized encoding and searching of literary information*, in IBM Journal of Research and Development 1 (4), pp. 309–317, 1957

[Maron, 1960] M. E. Maron and J. L. Kuhns, *On relevance, probabilistic indexing and information retrieval*, Journal of the ACM, Vol. 7(3), pp. 216-243, July 1960

[Miller, 1998] David R.H. Miller, Tim Leek and Richard M. Schwartz, *BBN at TREC-7: Using Hidden Markov Models for Information Retrieval*, TREC7, pp. 133-142, 1998

[Miller, 1999] David R. H. Miller and Tim Leek and Richard M. Schwartz, *A Hidden Markov Model Information Retrieval System*, Research and Development in Information Retrieval, Proc. ACM-SIGIR, pp. 214-221, 1999

[Moffat, 1994] Moffat, Sacks-Davis, Wilkinson, Zobel, *Retrieval of partial documents*, Text REtrieval Conference, 1994

[Moirand, 1987] Sophie MOIRAND, *Situations d'Ecrit,*, Coll. Didactique des langues étrangères, Paris : CLE International, 1988

[Ng, 1999] Kenney Ng, *A Maximum Likelihood Ratio Information Retrieval Model*, TREC8, pp. 483-492, 1999

[Nie, 2007] http://www.iro.umontreal.ca/%7Enie/IFT6255/Modeles_Probabilistes.html, 2007

[Palmer, 1990] Patrick Palmer, *Etude d'un analyseur de surface de la langue naturelle: application à l'indexation automatique de textes*, Ph.D. thesis, Université Joseph Fourier, 1990

[Ponte, 1998] Jay M. Ponte and W. Bruce Croft, *A Language Modeling Approach to Information Retrieval*, Research and Development in Information Retrieval, Proc. ACM-SIGIR, pp. 275-281, 1998

[RNRT, 1999] RNRT : http://www.telecom.gouv.fr/rnrt/projets/res_d97_ap99.htm,

[Robertson 1976] S.E. Robertson and K. Sparck Jones, *Relevance weighting of search terms*, Journal of the American Society for Information Sciences, Vol. 27(3), pp; 129-146, 1976

[Rocchio, 1971] J. Rocchio, *Relevance feedback information retrieval*, In G. Salton, editor, The Smart Retrieval System-Experiments in Automatic Document Processing, Prentice-Hall, pp. 313-323, 1971

[Salton, 1971] G. Salton, *The SMART Retrieval System ; Experiments in Automatic Document Processing*, Englenwood Cliffs, Prenctice-Hall, New Jersey, 1971

[Salton, 1975] G. Salton, *A theory of indexing*, 1975

[Salton, 1983] G. Salton, E. A. Fox, et H. Wu, *Extended Boolean information retrieval*, Communications of the ACM, Vol. 26(12), pp 1022-1036, 1983

[Salton, 1988] Salton, Buckley, *Term weighting approaches in automatic text retrieval*, Information Processing and Management 24, pp. 513-523, 1988

[Salton, 1993] Salton, Allan, Buckley, *Approaches to passage retrieval in full text information systems*, ACM-SIGIR, 1993

[Saltonb, 1983] Salton and McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983

[Schmitt, 1988] SCHMITT, M.-P. ; A. VIALA, *Savoir-lire*, Précis de Lecture Critique, Paris : Didier, 5e édition, 1988

[Smeaton, 1989] Alan F. Smeaton, *Information retrieval and natural language processing*, In proceedings of a conference jointly sponsored by ASLIB, University of York, page 2, march 1989

[Taghva, 1994a] Kazem Taghva, Julie Borsack, Allen Condit and Srinivas Erva, *The effects of noisy data on text retrieval*, JASIS vol. 45, pp. 50-58, 1994

[Taghva, 1994b] K. Taghva, J. Borsack, A. Condit and J. Gilbreth, *Results and implications of the noisy data projects*, In annual report of UNLV Information Science research institute, Las Vegas, NV, pp. 49-58, march 1994

[Takagi, 1996] Kazuyuki Takagi, Shuichi Itahashi, *Segmentation of spoken dialogue by interjections, disfluent utterances and pauses*, 1996

[Tambellini, 1994] C. Tambellini, C. Berrut, C. Brouard, *Une Analyse préalable à l'indexation de transcriptions de conversations téléphoniques*, CORIA 2004, Toulouse, pp 307-331, 2004

[Tsuda, 1995] K. Tsuda, S. Senda, M. Minoh and K. Ikeda, *Clustering OCR-ed texts for browsing document image database*, in Proceedings of the third international conference on document analysis and recognition, Montreal, Canada, pp. 171-174, august 1995

[Wayne, 2000] Charles L. Wayne, *Multilingual topic Detection and Tracking : Successful Research Enabled by Corpora and Evaluation*, 2000

[Zipf, 1932] G. Zipf, *Selective Studies and the Principle of Relative Frequency in Language*, in Harvard University Press, 1932