Université de Neuchâtel
Faculté des sciences

Cette thèse intitulée

Recherche d'information bilingue et multilingue : Amélioration de la traduction automatique et sélection de traducteur

a été présentée le 9 juin 2006 par

Pierre-Yves Berger

pour l'obtention du grade de Docteur ès Sciences

Composition du jury

Professeur Jacques Savoy
Directeur de thèse
Université de Neuchâtel, Suisse

Professeur Jian-Yun Nie
Université de Montréal, Canada

Professeur Jacques Pasquier
Université de Fribourg, Suisse

Professeur Pascal Felber
Université de Neuchâtel, Suisse

Université de Neuchâtel

IMPRIMATUR POUR LA THESE

Recherche d'information bilingue et multilingue : Amélioration de traduction automatique et sélection de traducteur

Pierre-Yves BERGER

UNIVERSITE DE NEUCHATEL

FACULTE DES SCIENCES

La Faculté des sciences de l'Université de Neuchâtel,
sur le rapport des membres du jury

MM. J. Savoy (directeur de thèse), P. Felber,
J. Pasquier (Fribourg) et
J.Y. Nie (Montréal)

autorise l'impression de la présente thèse.

Neuchâtel, le 20 juin 2006

Le doyen :

J.-P. Derendinger

■ **Faculté des Sciences**
■ Rue Emile-Argand 11 ■ CP 158 ■ CH-2009 Neuchâtel
■ Téléphone : +41 32 718 21 00 ■ Fax : +41 32 718 21 03 ■ E-mail : secretariat.sciences@unine.ch ■ www.unine.ch

Résumé

Mots clés : Recherche d'information bilingue, apprentissage automatique, sélection de traduction automatique

Keywords: Bilingual information retrieval, machine learning, machine translation selection

Dans cette thèse, nous abordons différentes techniques visant à améliorer l'utilisation de la traduction automatique dans le contexte de la recherche d'information bilingue et multilingue. Nous avons quantifié la perte de performance liée à la traduction automatique de la requête d'une langue vers une autre. Pour diminuer cette perte d'efficacité, nous avons évalué six approches, à savoir :

- utilisation de la traduction inverse pour sélectionner les mots conservés;
- divers systèmes de combinaison de traductions;
- apprentissage automatique à l'aide de la méthode des plus proches voisins;
- apprentissage automatique à l'aide de la génération d'arbres de décision;
- apprentissage automatique à l'aide de la régression logistique;
- fusion de résultats.

Les approches orientées plutôt vers la linguistique n'ont pas apporté une amélioration claire et évidente de la performance de recherche. Les systèmes issus du domaine de l'apprentissage automatique nous ont donné des résultats encourageants lors de nos expérimentations utilisant l'anglais comme langue de départ et l'espagnol et l'allemand comme langues cibles.

Remerciements

Bon nombre de personnes et institutions ont contribué de près ou de loin à la réalisation de cette thèse. J'aimerais les remercier toutes même si je ne pourrai ici en donner la liste exhaustive.

Plus particulièrement, j'aimerais remercier,

Ma famille et mes amis pour leurs soutiens et encouragements durant toute cette période.

Mon directeur de thèse, le professeur Jacques Savoy, qui m'a permis de démarrer ce projet et m'a aiguillé et conseillé pendant toute sa durée. Son aide fut précieuse.

Les autres membres du jury de thèse, les professeurs Pascal Felber (Université de Neuchâtel), Jacques Pasquier (Université de Fribourg) et Jian-Yun Nie (Université de Montréal) pour le temps qu'ils ont consacré à l'expertise de cette thèse.

Ma collègue et amie Laura Perret Rolli avec qui j'ai été amené à collaborer étroitement, nos projets de thèse étant liés. Ses suggestions, ses conseils et son soutien m'ont permis de mener ce projet à bien.

C. Buckley de SabIR pour m'avoir donné l'opportunité d'utiliser le système SMART.

Enfin, cette recherche a été partiellement soutenue par le Fonds national suisse de la recherche scientifique avec la bourse 21-66 742.01. L'université de Neuchâtel m'a aussi apporté sa contribution en mettant à ma disposition un poste d'assistant.

Table des matières

1 Introduction

1.1 Recherche d'information

Les informations disponibles en format électronique sont de plus en plus nombreuses. La Toile comporte actuellement plusieurs milliards de pages et Google annonce plus de huit milliards de pages indexées au début 2005 (Google Search Engine). Les moteurs de recherche tiennent une part importante dans le développement rapide de la Toile, comme le montre Nielsen (2004) qui, dans le cadre d'un questionnaire à propos des usages sur Internet, a demandé à des utilisateurs de citer leurs sites favoris. Dans tous les cas, un moteur de recherche se trouvait parmi les trois premiers. De plus, lorsqu'on demande à un utilisateur de nous donner une information disponible sur la Toile, dans 88 % des cas, le premier site qu'il visite est un moteur de recherche qu'il utilisera comme point de départ de sa navigation.

Certaines bases de données comme les bases médicales, astronomiques, météorologiques ou encore de physique nucléaire contiennent des quantités d'informations encore plus importantes. Le problème actuel n'est plus seulement de stocker l'information mais de la retrouver pour pouvoir l'exploiter. Dans le cas des bases de données, les données étant atomiques, structurées et de sémantique simple, on arrive raisonnablement à résoudre le problème de l'accès à l'information souhaitée. Van Rijsbergen (1979) présente quelques différences entre l'extraction de données (typiquement utilisée dans les bases de données) et la recherche d'informations (voir tableau 1).

	Bases de données	Recherche d'information
Appariement	Exact	Partiel, recherche des meilleures réponses
Modèle	Déterministe	Probabiliste
Langage de requête	Artificiel (SQL)	Langue naturelle
Spécification de la requête	Complète	Incomplète
Éléments désirés	Égaux	Pertinents

Tableau 1 Principales différences entre l'interrogation dans les bases de données et la recherche d'information

Le problème de la recherche d'information se pose lorsqu'on veut retrouver une information dans une collection importante de données peu structurées. Imaginons que nous avons une collection de documents, par exemple la jurisprudence du Tribunal Fédéral ou toutes les dépêches de l'Agence France-Presse ou encore les

photos prises par la Royal Navy durant la dernière guerre. Nous avons aussi un utilisateur qui est capable d'exprimer un besoin d'information qui peut être satisfait par certains documents de la collection, par exemple respectivement, "quelle est la jurisprudence sur les excès de vitesse en localité ?" ou "qui a gagné le tour de France cycliste en 1987 ?" ou encore "trouver des images montrant l'étendue des destructions opérées par le bombardement allié de Dresde le 13 février 1945". Le but d'un système de recherche d'information est de retrouver rapidement tous les documents répondant au besoin d'information de l'utilisateur sans y inclure des documents non pertinents sachant que la requête est une représentation partielle imprécise et ambiguë du besoin réel de l'usager. Pour nos exemples, les documents à retrouver seraient les textes des arrêts concernés, les dépêches mentionnant le vainqueur du tour de France 1987 ou les images de Dresde les 12 et 13 février 1945 dans la zone de bombardement. La figure 1 illustre ce principe.

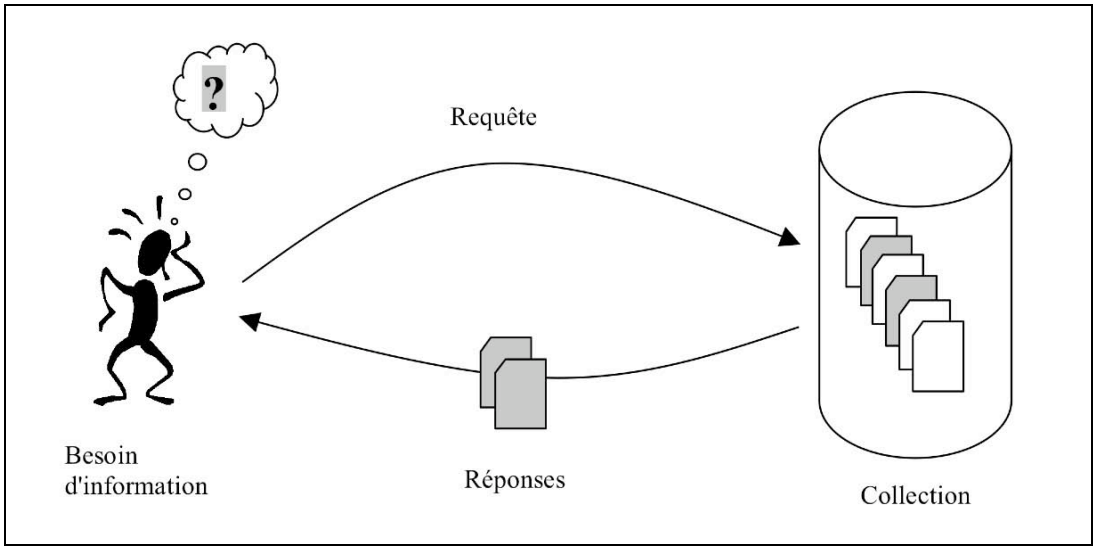

Figure 1 La recherche d'information

Un système de recherche d'information comporte plusieurs parties. Les documents qui seront la cible de la recherche sont groupés en collection. Avant la recherche proprement dite, la collection de documents est indexée. Ce processus va créer une représentation de chaque document selon un modèle choisi dépendant du type de données que comportent les documents (image, texte, parole, etc.). Le résultat du processus est un index qui permet de retrouver des documents en fonction de leur représentation.

Au moment de la recherche, l'utilisateur doit présenter son besoin d'information au système. Il peut le faire en naviguant dans la collection et en annonçant si cela correspond à ce qu'il cherche; ou transmettre un document et en demander de similaires; ou encore donner des mots-clés correspondant à sa question. Le

système doit alors créer une requête à partir de ces informations. La requête est ensuite traduite de manière compatible avec l'index de la collection.

Puis, le processus d'appariement se charge de constituer une liste ordonnée de documents que le système juge pertinent (du plus pertinent au moins pertinent). Le système peut aussi ajouter d'autres informations, comme la valeur que le système a attribuée au document en guise de mesure de pertinence.

La figure 2 présente les parties principales d'un système de recherche d'information. De plus, au lieu d'une simple référence au document, il indiquera le titre de l'article, un petit résumé ou des passages dans lesquels apparaissent le ou les mots de la requête. Lorsque la collection est relativement statique et les requêtes changeantes, comme dans les exemples ci-dessus, on parle de tâche *ad hoc*. Par contre, si les requêtes sont statiques et la collection change, il s'agit d'une tâche de *filtrage*. C'est par exemple le cas si la requête représente les centres d'intérêts et de spécialisation d'un expert et les documents sont les nouveaux articles publiés. Dans ce cas, le résultat n'est habituellement pas ordonné, le système traitant les documents au fur et à mesure de leur arrivée.

Certains systèmes ajoutent un principe de rétroaction en utilisant des données de pertinence fournies par l'utilisateur afin d'améliorer la requête (Rocchio 1971). D'autres vont plus loin en faisant l'hypothèse que les premiers documents retrouvés sont pertinents et en les utilisant pour améliorer les requêtes. On parle alors de pseudo-rétroaction (Buckley 1994).

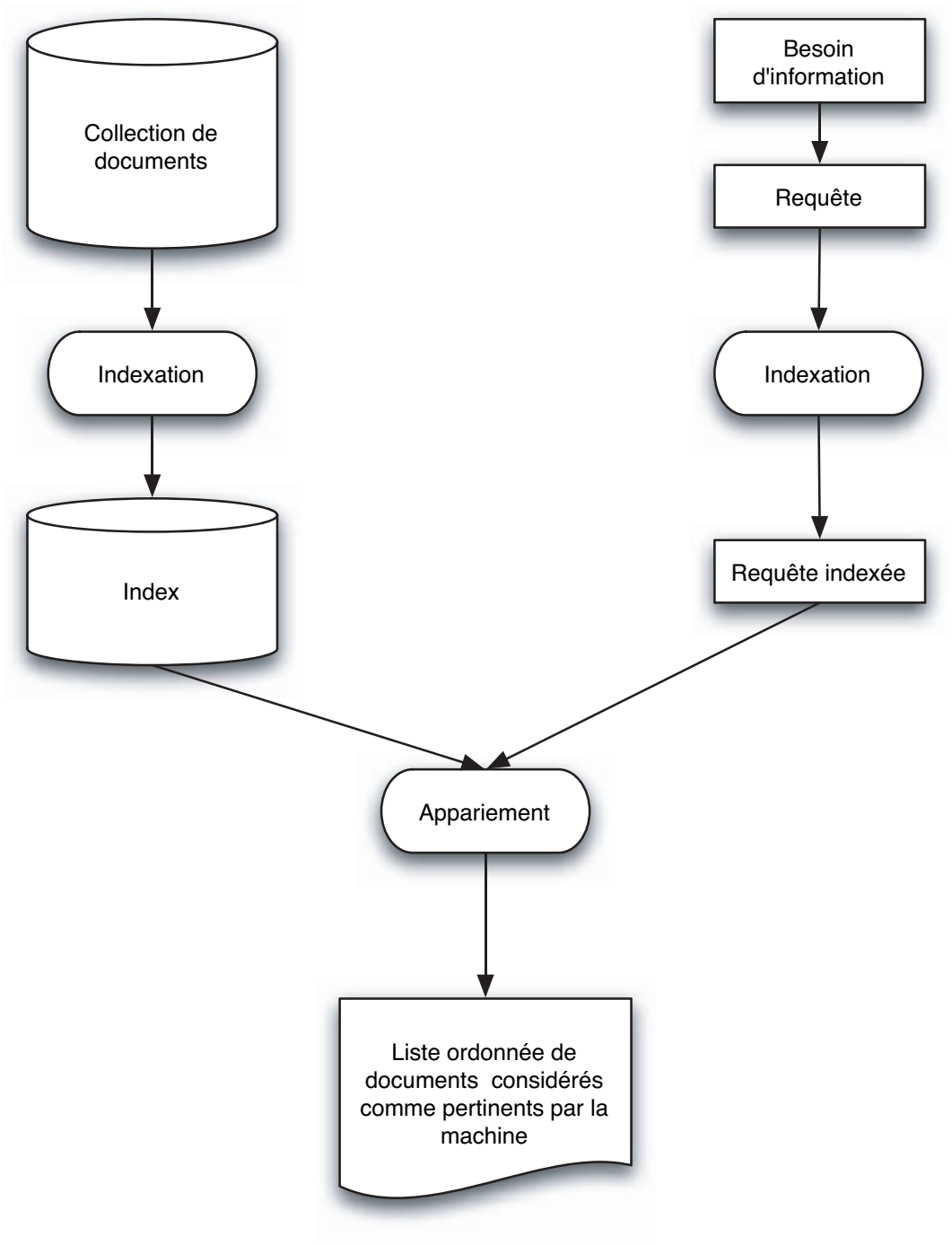

Figure 2 Fonctionnement d'un système de recherche d'information

1.2 Recherche d'information dans des textes

Le texte correspond à la forme la plus utilisée par l'être humain afin de conserver la connaissance et, après la parole, le principal medium de communication (Frakes *et al.*, 1992). Il n'y a pas si longtemps, rechercher des informations revenait à se rendre dans une bibliothèque et demander au bibliothécaire s'il avait un ouvrage concernant le sujet choisi. Avec la multiplication des livres à la fin du dix-neuvième et au début du vingtième siècle, des systèmes d'index ont été introduits, référençant des noms d'auteurs, des titres, des domaines hiérarchisés, voire des mots-clés pour les plus évolués. Avec l'apparition des textes sur support électronique, il devient potentiellement possible de référencer tous les mots d'un document dans l'index. C'est sur ce principe que vont se baser les systèmes de recherche d'information.

Pour créer une représentation d'un document, le système doit découper le texte en unités atomiques appelées termes. Celles-ci peuvent par exemple être les mots du texte, les lemmes (forme de référence d'un mot, celle que l'on trouve dans un dictionnaire) ou encore des n-grammes (suite de n caractères), ce dernier découpage étant particulièrement adapté pour les langues asiatiques (Savoy 2005).

Avant l'indexation, certains prétraitements peuvent être accomplis sur le texte. Ces traitements ont pour but d'améliorer la pertinence des documents restitués par le système.

La première technique est l'élimination des mots-outils tant dans les documents que dans les requêtes. Les mots-outils comme les déterminants, les pronoms, les conjonctions ou les prépositions n'ont pas ou peu de signification pour la recherche. De plus ils sont présents dans presque tous les documents. Plusieurs méthodes existent pour la création de listes de mots-outils. On peut utiliser les statistiques de fréquence des termes dans la collection en considérant que les plus fréquents n'apportent pas d'information pour la recherche (Fox 1990). Savoy (2002) propose de filtrer manuellement cette liste pour en retirer les nombres et les noms ou adjectifs importants pour le sujet de la collection, puis de rajouter les mots non porteurs de sens spécifiques au domaine. Une autre approche, si on a déjà une liste de mots-outils dans une autre langue, consiste à traduire cette liste de mots dans la langue qu'on veut utiliser (Chen & Gey 2002). Moulinier (2004) montre que l'élimination de mots-outils n'est pas toujours intéressante et cette question n'a pas encore trouvé de réponse pour certaines langues asiatiques comme le chinois ou le japonais. Les requêtes de longueur moyenne sont celles qui bénéficient le plus de cette technique. Pour les requêtes longues, les mots-outils n'influencent que peu le résultat et leur élimination n'a donc que peu d'effet. Pour les requêtes courtes, l'effet bénéfique est compensé par la détérioration du résultat d'autres requêtes.

Un autre traitement classique est la racinisation. Le but est de supprimer certaines séquences terminales comme les marques du féminin ou du pluriel ("volcan**s**"

devient "volcan"). Certaines dérivations suffixales peuvent aussi être retirées permettant le rapprochement de mots de même racine ("volcan**ique**" devient ainsi "volcan"). Ce traitement linguistique a ses supporters et ses détracteurs et son efficacité semble liée à la langue utilisée. Pour l'anglais, Harman (1991) soutient que la racinisation n'améliore pas significativement la performance de la recherche d'information. Hull (1996) a constaté que, si l'opération n'améliorait pas la performance de manière significative en moyenne, elle ne la détériorait pas non plus. Pour certaines requêtes, une amélioration peut toutefois être détectée. D'autres études sur le français (Savoy 1999) ou le néerlandais (Kraaij & Pohlmann 1996) montrent l'importance de la racinisation pour ces langues morphologiquement plus riches que l'anglais. Les enracineurs peuvent être plus ou moins évolués. Pour la langue anglaise, le plus simple, *S-stemmer*, se contente de retirer la marque du pluriel. D'autres, comme Porter (1980) ou Lovins (1968) se fondent sur des systèmes de règles basées sur la grammaire. Lovins cherche, pour un mot donné, la règle qui supprimera le suffixe le plus long alors que Porter applique récursivement les règles tant que possible.

Un autre traitement fréquent pour les langues européennes s'approchant de la racinisation est la conversion des majuscules en minuscules et la suppression des accents. Une translittération est aussi effectuée pour passer de l'alphabet cyrillique à l'alphabet latin lors du travail avec le russe ou le bulgare.

Certaines langues, comme l'allemand, le néerlandais ou le finnois, permettent de créer de nouveaux mots en juxtaposant plusieurs mots. Par exemple, en allemand, une "société d'assurance vie" se dit *Lebensversicherungsgesellschaft* et est formé de *Leben* (vie), *Versicherung* (assurance) et *Gesellschaft* (société). Dans ce cas, un système de décomposition des mots peut améliorer la recherche (Savoy 2003).

Il existe trois modèles classiques utilisés en recherche d'information textuelle : le modèle booléen, le modèle vectoriel et le modèle probabiliste.

Modèle booléen

Le premier à être proposé en recherche d'information fut le modèle booléen. Chaque document est représenté par un ensemble contenant ses termes. La requête définissant le besoin d'information est une expression logique contenant les termes recherchés liés entre eux par des opérateurs logiques (ET, OU, NON). Un document est alors jugé pertinent et extrait de la collection si sa représentation satisfait l'expression logique de la requête. Dans le cas contraire, il est considéré comme non pertinent. La réponse est donc binaire, sans possibilité de gradation ou d'ordonnancement des résultats.

Ce système connaît un certain succès par son efficacité. En effet, le temps de réponse reste relativement faible même sur de grandes collections de documents. La facilité d'utilisation est aussi un avantage. Un usager comprend facilement que s'il écrit une requête avec "recherche ET information", il obtiendra des documents comprenant ces deux termes. Le comportement du système est aussi facile à

expliquer, la présence d'un document dans le résultat étant visiblement fonction du contenu de la requête. Enfin, il est efficace si on a une connaissance a priori du contenu des documents de la collection afin d'utiliser les bons termes dans la requête.

Le désavantage principal de ce modèle est son absence de support pour la satisfaction partielle d'une requête et pour l'ordonnancement des résultats. Ensuite, il n'est pas toujours facile d'écrire une expression logique représentant le besoin d'information avec suffisamment de précision. Enfin, la langue peut poser un problème. La polysémie (plusieurs sens pour un terme) et la synonymie (plusieurs termes pour une signification) dégradent considérablement les résultats de la recherche ou allongent les requêtes d'une manière disproportionnée.

Modèle vectoriel

Le modèle vectoriel cherche à permettre la satisfaction partielle d'une requête et à donner la possibilité de formuler avec plus de précision l'importance des mots dans un document ou une requête (Salton 1971). Nous considérons donc que chaque terme d'indexation définit une dimension d'un espace. Chaque document est représenté par un vecteur dans cet espace dont les composantes contiennent les poids attribués aux termes de ce document. La manière d'attribuer ces poids est présentée ci-dessous. De la même manière, les requêtes sont aussi représentées par des vecteurs. Nous calculons alors la mesure de similarité entre le vecteur du document $\vec{d}$ et celui de la requête $\vec{q}$ en fonction de l'angle entre ces deux vecteurs. Plus l'angle est aigu, plus le document est proche de la requête, donc plus la similarité est grande. La figure 3 illustre ce concept avec deux termes t_1 et t_2 pour deux documents d_1 et d_2 et une requête notée q. Nous avons donc les angles ∂_1 et ∂_2 comme représentation de la mesure de similarité entre d_1 (respectivement d_2) et la requête.

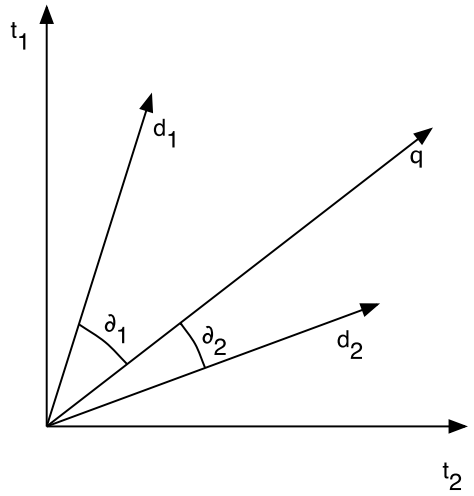

Figure 3 Illustration du modèle vectoriel

Pour calculer la valeur numérique de similarité, on peut par exemple utiliser le cosinus de l'angle :

$$similarité(d,q) = \frac{\vec{d} \bullet \vec{q}}{\left|\vec{d}\right| \times \left|\vec{q}\right|} \qquad (1)$$

Les documents peuvent alors être ordonnés en fonction de leur valeur de similarité avec la requête.

Selon des expériences menées sur la langue anglaise (Salton & Buckley 1988, Voorhees & Harman, 2000), la pondération d'un terme t_i dans un document D_j devrait tenir compte du nombre d'occurrence du terme dans le document (tf_{ij}), de la fréquence documentaire $df_i = \log_2\left(n/df_j\right)$ et de la longueur du document.

Il existe plusieurs schémas de pondération reprenant certains des critères énoncés ci-dessus. Les formules de ceux que nous avons utilisés se trouvent en Annexe 1. La dénomination retenue est dérivée du système SMART. Ainsi, pour décrire précisément un modèle de dépistage, un premier triplet de lettres décrit la pondération utilisée lors de l'indexation des documents et, un second triplet, celle appliquée aux requêtes. Par exemple, une stratégie « bnn-bnn » indique une indexation binaire (terme présent ou non) tandis que la séquence « nnn-nnn » signifie que seul le nombre d'occurrences est retenu pour pondérer les termes des documents et des requêtes.

Pour le modèle vectoriel classique « ntc-ntc », l'indexation tient compte à la fois de la fréquence d'occurrences dans le document et de l'inverse de la fréquence documentaire (nombre de documents dans lesquels le terme apparaît). De plus, dans cette stratégie « ntc-ntc », les poids sont normalisés selon la formulation du cosinus.

Cependant la fréquence d'occurrences peut être modifiée pour tenir compte du fait que l'apparition de la première occurrence devrait posséder un poids important. De plus, nous devrions accorder une importance décroissante au fil des répétitions d'un même terme dans un document. Ainsi, la différence entre une fréquence d'occurrences de neuf ou de huit n'apporte pas une information très précieuse tandis que la différence entre une fréquence unitaire ou nulle constitue une information très pertinente. Afin de respecter ces deux principes, nous proposons de pondérer un terme selon l'équation $[0,5 + 0,5 \cdot (tf_{ij}/\max tf_{i.})]$, de prendre le logarithme de la fréquence d'occurrences ($\ln(tf_{ij})+1$) ou de recourir au double logarithme ($\ln(\ln(tf_{ij})+1)+1$).

De plus, de nouvelles formules de pondération plus complexes ont été mises au point, en particulier, le modèle probabiliste Okapi (Robertson *et al.* 2000), le modèle vectoriel « Lnu-ltc » (Buckley *et al.* 1995) ou la stratégie « dtu-dtn » (Singhal *et al.* 1998). Ces dernières possèdent l'avantage de tenir compte de la longueur des documents en cherchant à pénaliser les longs documents abordant

généralement plusieurs sujets et qui répondent, en moyenne, moins bien aux attentes de l'usager.

Dans toutes nos expériences, le score de chaque document (ou son degré de pertinence jugé par la machine) est obtenu par le calcul du produit interne. Par exemple, pour l'approche « bnn-bnn », ce score indiquera le nombre de termes communs entre le document et la requête. Pour l'approche « nnn-nnn », ce score tiendra compte de la fréquence d'occurrences des termes communs entre le document et la requête.

Le modèle vectoriel permet de traiter la satisfaction partielle des requêtes. Il permet aussi, grâce à la mesure de similarité, de classer les différents documents retournés. D'un autre côté, les hypothèses faites au niveau linguistique peuvent avoir une influence négative sur la performance de la recherche d'information. Par exemple, nous faisons l'hypothèse qu'un terme est une unité sémantique dans notre représentation des documents. Or, dans les langues naturelles, ce n'est pas forcément le cas. Les problèmes de polysémie et de synonymie doivent donc être traités hors du modèle lui-même.

Modèle probabiliste

Le modèle probabiliste proposé par Robertson (1977) se base sur le "principe du classement probabiliste". L'idée de base de ce modèle est de classer les documents retrouvés par probabilité de pertinence. Le but est donc de déterminer la probabilité que si on retrouve un document *D*, celui-ci appartient à l'ensemble des documents pertinents *P* ou non pertinents *NP*. Ces probabilités sont notées P(*P*|*D*), respectivement P(*NP*|*D*). La fonction de similarité avec la requête sera donc le rapport de ces deux probabilités, soit :

$$similarité(D,Q) = \frac{\mathrm{P}(P \mid D)}{\mathrm{P}(NP \mid D)} \qquad (2)$$

Comme ces deux probabilités ne sont pas directement calculables, nous utilisons le théorème de Bayes pour obtenir :

$$similarité(D,Q) = \frac{\mathrm{P}(D \mid P) \cdot \mathrm{P}(P)}{\mathrm{P}(D \mid NP) \cdot \mathrm{P}(NP)} \qquad (3)$$

avec

- P(*P*) la probabilité que si on tire un document au hasard dans la collection, il soit pertinent,
- P(*NP*) la probabilité que si on tire un document au hasard dans la collection, il ne soit pas pertinent,
- P(*D*|*P*) la probabilité que le document *D* appartienne à la classe des documents pertinents *P* et
- P(*D*|*NP*) la probabilité que le document *D* appartienne à la classe des documents non pertinents *NP*.

Comme P(*P*) et P(*NP*) sont constants pour une requête donnée, ils n'ont pas d'influence sur le classement des documents retrouvés et peuvent donc être éliminés de la formule. Il nous reste donc à estimer P(*D*|*P*) et P(*D*|*NP*). Plusieurs modèles existent pour ces estimations. L'un d'eux proposé par Robertson et Walker (1994) est à l'origine de la famille de moteurs de recherche Okapi. Le score attribué au document noté RSV (pour *retrieval status value*) est calculé comme suit :

$$RSV(D_i,Q) = \sum_{j=1}^{t} \frac{(k_1+1)\cdot tf_{ij}}{K + tf_{ij}} \text{ avec } K = k_1 \cdot \left[(1-b) + \frac{b \cdot l_i}{avdl}\right] \quad (4)$$

Dans cette formule, k_1 et b sont des constantes, *avdl* indique la taille moyenne des documents de la collection, l_i la longueur du document D_i et tf_{ij} la fréquence d'occurrence du terme t_j dans la requête.

1.3 Un environnement multilingue

Il est instructif de s'intéresser à la répartition des langues utilisées par les internautes. En effet, à fin 1996, l'essentiel (85 %) des utilisateurs de la Toile parlaient l'anglais alors qu'il y avait environ 47 millions d'internautes. En décembre 2000, sur les 407 millions d'internautes, il n'y avait plus que 47 % d'anglophones. À fin 2004, cette proportion a chuté à 30 % sur 968 millions d'utilisateurs connectés[1]. La figure 4 montre cette évolution.

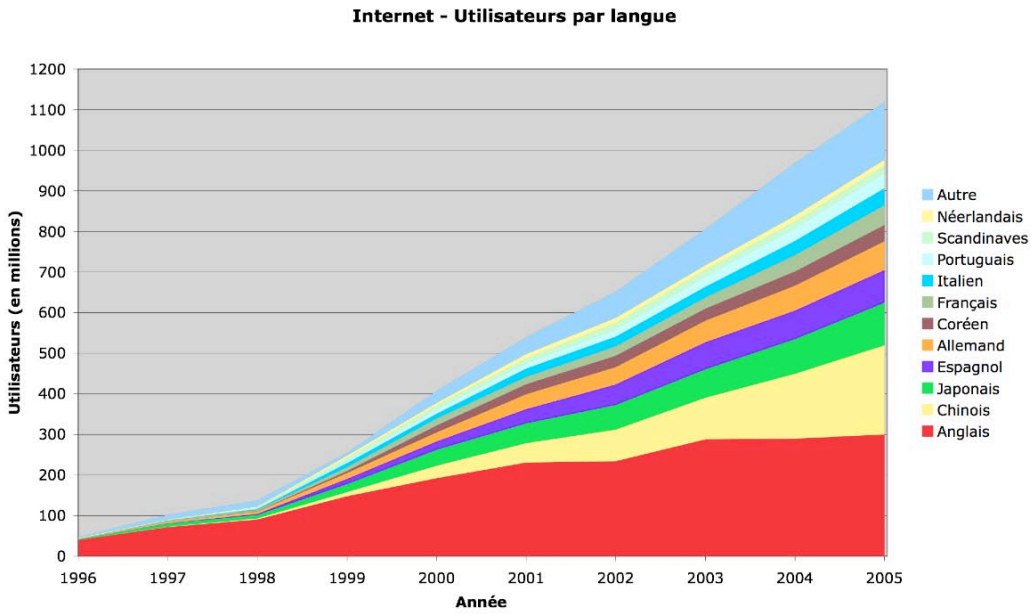

Figure 4 Évolution du nombre d'utilisateurs d'Internet par langue.

Si les utilisateurs anglophones ne sont plus qu'un tiers, ce n'est pas le cas des pages disponibles. En 2003, les index de AllTheWeb comportent encore 58 % de documents en langue anglaise alors que Google en compte 63 % (AllTheWeb Search Engine, Google Search Engine). De plus, peu de sites transmettent les mêmes informations dans plusieurs langues. La plupart des sites, lorsqu'ils proposent plusieurs langues, ne donnent que des informations lacunaires dans les langues qui ne sont pas celle de l'institution.

Mais la Toile n'est pas le seul environnement multilingue. La plupart des pays ne sont pas monolingues. En simplifiant, il y a environ 5000 langues sur la surface du globe pour environ 200 pays, ce qui nous donne une moyenne de 25 langues par pays (Calvet 2000). Toutes ne sont pas sur un pied d'égalité. Celles du nord-est de la Russie ne sont parlées au plus que par quelques villages alors que

[1] Les statistiques sont données en annexe 2. Elles ont été compilées à partir des sources d'information suivantes : (Global Reach), (Internet World Stats) et (Nua Ltd)

d'autres comme les langues dites *internationales* sont utilisées par plusieurs centaines de millions de locuteurs. Certains pays, comme l'Inde, la Belgique ou la Suisse, ou des organisations internationales, comme l'Union Européenne ou l'ONU, doivent aussi composer avec des informations dont la traduction dans les différentes langues n'est pas toujours disponible. Les multinationales sont elles aussi concernées par ces problèmes.

Pour reprendre le cas de l'Union Indienne, celle-ci peut se targuer d'avoir 1652 langues usitées dont 33 comptent plus 100'000 locuteurs et 15 sont officielles (Gauthier 2000). Contrairement aux langues orales du Nord de la Russie, qui n'ont qu'un très faible ancrage culturel et ne sont parlées que par quelques centaines d'individus, les langues indiennes ont chacune une écriture différente et une longue tradition culturelle, millénaire pour certaines. Le hindi, langue officielle de l'Union Indienne, n'est une langue officielle que dans six des vingt-huit états membres. On remarque aussi que l'une des langues officielles minoritaires, le bengali, n'est parlée que par 7,6 % de la population indienne, ce qui représente tout de même plus de personnes que l'ensemble des francophones d'Europe. Lorsqu'il veut obtenir des informations de l'Union Indienne, un fonctionnaire bengali devra donc probablement opérer des recherches dans une langue qui n'est pas la sienne. Pour beaucoup de bengali, les documents en hindi sont compréhensibles, mais ils sont incapables de formuler une requête avec les mots appropriés dans cette langue.

Un autre exemple similaire est le fonctionnement judiciaire en Suisse. Le Tribunal Fédéral rend ses jugements dans la langue de l'instance cantonale dont la décision est attaquée[2] (donc en allemand, français ou italien). Les jugements ne sont pas systématiquement traduits. Un juriste doit pouvoir rechercher dans les différents jugements, quelle que soit leur langue, pour vérifier si un des jugements existants s'apparente au cas qu'il est en train de traiter. Une fois qu'il a trouvé certains documents dans une langue qui n'est pas la sienne, il va peut-être utiliser un système de traduction automatique afin de connaître le sens général du texte, puis, si des parties sont réellement intéressantes, les faire traduire par un traducteur juré.

Enfin, il existe de plus en plus de banques d'images dans lesquelles on peut opérer des recherches en se basant sur les légendes qui accompagnent ces images. Quelle que soit la langue des légendes, si quelqu'un cherche une image de « jonques dans un port », il sera capable de déterminer si l'image qui lui est présentée va satisfaire sa requête. Il lui sera par contre beaucoup plus difficile de créer une requête correcte dans chacune des langues des légendes pour obtenir l'image désirée.

[2] Loi fédérale d'organisation judiciaire, article 37, alinéa 3.

1.4 Recherche bilingue ou multilingue d'information

Pour résoudre le problème posé par ces exemples, l'utilisateur pourra recourir à la recherche bilingue ou multilingue d'information. L'action consistant à chercher des informations dans une collection dont la langue n'est pas la même que celle de la requête s'appelle la recherche bilingue d'information. Si la recherche s'effectue dans plusieurs collections de langues différentes, il s'agit alors de recherche multilingue d'information.

Lorsqu'une traduction est réalisée, ce n'est pas toujours la plus correcte grammaticalement qui est la plus efficace quant à la performance de la recherche d'information. Certaines traductions automatiques peu compréhensibles peuvent amener suffisamment d'informations à l'outil de recherche pour obtenir une bonne performance alors qu'une excellente traduction du point de vue linguistique utilisera des mots si spécifiques que l'outil ne parviendra peut-être pas à retourner des documents pertinents. Le tableau 2 donne un exemple de deux traductions allemandes d'une requête avec leurs précisions moyennes[3]. On constate que REVERSO, dont la traduction est tout à fait correcte pour un être humain, obtient de moins bons résultats que la traduction peu compréhensible de FREE.

Traduction	Requête	Précision
Source (anglais)	U.N./US Invasion of Haiti. Find documents on the invasion of Haiti by U.N./US soldiers.	
REVERSO	Invasion der Vereinter Nationen Vereinigter Staaten Haitis. Finden Sie Dokumente auf der Invasion Haitis durch Vereinte Nationen Vereinigte Staaten Soldaten.	40.07
FREE	U N UNS Invasion von Haiti. Fund dokumentiert auf der Invasion von Haiti durch U N UNS Soldaten	72.14

Tableau 2 Exemples de traductions

Les traducteurs automatiques ont un certain nombre de lacunes dans le traitement de la polysémie. Ainsi, le mot *sentence* en anglais peut signifier *phrase* ou *condamnation* en français. Quelle signification choisir lors de la traduction ?

Les noms propres sont aussi source d'erreur. La phrase « President Bush visited Japan » a été traduite par « le président buisson a visité le Japon », ce qui n'a pas permis de retrouver de documents pertinents. D'autre part, lorsque « Bush fires in Sydney » est traduit par « le président Bush renvoie Sydney », le résultat est identique. Un autre problème se situe dans la traduction de certains noms propres. Ainsi, *Gorbatchov* peut être traduit par *Gorbatchev* en français, mais ce n'est pas systématique. Dans des textes français, on peut trouver aussi bien la première forme que la seconde.

[3] La *précision moyenne* est expliquée plus loin.

1.5 Objectifs

Le but de ce travail est d'implémenter et évaluer différentes techniques d'amélioration de la traduction automatique de textes utilisée dans le contexte spécifique de la recherche d'information.

La première technique considérée (section 3.3) s'attache au renforcement de la traduction en utilisant la traduction inverse.

La deuxième (section 3.4) cherche à combiner les différentes traductions selon divers critères (par exemple par concaténation des traductions ou par choix des mots appartenant à plusieurs traductions).

Les trois suivantes font intervenir l'apprentissage automatique afin de sélectionner la meilleure traduction possible pour une requête donnée : la méthode des plus proches voisins (section 3.6), la génération automatique d'arbres de décision (section 3.7) et la régression logistique (section 3.8).

La dernière (section 3.9) utilise la fusion de listes de résultats pour mettre ensemble les résultats de plusieurs recherches, une pour chaque traduction effectuée.

2 État des connaissances

2.1 Recherche bilingue ou multilingue d'information

L'activité consistant à chercher des informations dans une collection dont la langue n'est pas la même que celle de la requête s'appelle la recherche bilingue d'information. Si la recherche s'effectue dans plusieurs collections de langues différentes, il s'agit alors de recherche multilingue d'information.

Pour effectuer une recherche d'information bilingue ou multilingue, il existe différentes méthodes qui sont exposées ci-dessous.

Lors du travail sur des langues similaires, par exemple entre l'anglais et le français, Buckley *et al.* (1998) ont constaté de grandes similitudes entre le vocabulaire des deux langues. Ils proposent donc de considérer l'anglais comme du français mal orthographié et utilisent des listes d'appariement. Ces listes sont confectionnées automatiquement à l'aide de quelques règles simples. Gey (2004) a repris cette idée pour minimiser les absences de traduction dans le cas du chinois et du japonais qui partagent la même écriture. Les résultats dans les deux cas ne sont pas très probants.

Lorsqu'il y a traduction, trois possibilités existent, qui ont chacune leurs avantages et inconvénients. La méthode la moins coûteuse est de ne traduire que les requêtes car elles sont beaucoup plus courtes que la collection de documents complète. Si la collection comprend plusieurs langues, on traduit les requêtes dans chaque langue, mais on doit, après avoir retrouvé des documents dans chaque langue, mettre ensemble les différentes listes de documents retournés dans un processus appelé fusion de collection (Voorhees *et al.* 1995). Chen (2002) a aussi tenté de traduire les requêtes dans toutes les langues et de concaténer ces traductions pour interroger une collection comprenant les documents de toutes les langues, s'affranchissant ainsi de la fusion, mais ses résultats sont moins bons qu'en utilisant la fusion.

On peut aussi ne traduire que les documents. L'avantage principal consiste en l'absence de fusion dans le cas multilingue au prix de la traduction de la collection entière. Cette méthode a été utilisée avec succès par Lam-Adesina *et al.* (2004) dans la piste multilingue de la campagne d'évaluation CLEF 2003. Mais si les différentes requêtes prévues ne sont pas toutes dans la même langue, il faut traduire toute la collection dans chacune des langues de requête, ce qui peut devenir très coûteux.

On peut enfin traduire les deux. Lam-Adesina *et al.* (2004) utilisent la notion de langage pivot : les requêtes et les documents sont tous traduits (si nécessaire) dans une langue commune (l'anglais dans ce cas précis), puis une recherche monolingue est effectuée. Braschler (2002) et Chen *et al.* (2004) proposent quant à eux de faire deux recherches, la première en traduisant les requêtes dans les

différentes langues des collections puis en fusionnant les résultats, la seconde en traduisant tous les documents dans la langue de la requête. Ensuite, on fusionne les résultats de ces deux recherches en faisant la somme des scores si un document est retourné deux fois. Cette méthode a donné de meilleurs résultats que la traduction des requêtes seules lors de la piste multilingue de CLEF 2003. Le désavantage de ces méthodes reste le coût de la traduction de la collection dans la langue de requête.

2.2 Méthodes de traduction

Traduction manuelle

Afin de traduire les requêtes ou les documents, plusieurs méthodes peuvent être employées. La première consiste à demander une traduction à un traducteur humain. Cette méthode que nous appelons traduction manuelle, par opposition à la traduction automatique, s'avère habituellement la plus efficace en termes de recherche mais s'avère très coûteuse, tant en temps qu'au niveau financier. Nous ne l'utiliserons donc que comme référence pour évaluer l'efficacité de différents systèmes de traduction automatique pour la recherche d'information.

Dictionnaires

Une première méthode de traduction automatique est la consultation d'un dictionnaire bilingue. Le dictionnaire bilingue consiste essentiellement en une liste de mots avec pour chacun, la ou les traductions qui lui correspondent. Le texte à traduire est donc pris mot à mot et pour chacun, on le remplace par le ou les mots traduits (Chen *et al.* 2004). Cette méthode de traduction à l'avantage d'être plus rapide et de consommer peu de ressources machine. Comme elle n'utilise pas d'informations linguistiques, elle ne peut pas résoudre le problème de la polysémie (plusieurs sens pour un même mot). Elle ne permet pas non plus de traduire correctement les mots composés comme *clair de lune* ou les expressions idiomatiques comme *il a cassé sa pipe*. Toutefois, pour les requêtes, qui ne comportent que peu d'informations linguistiques permettant la désambiguïsation, voire qui ne sont composées que d'une suite de mots et non d'une phrase bien formée, les performances sont équivalentes à d'autres systèmes de traductions (Jones 2005).

Traduction automatique

Différents systèmes de traduction automatique sont disponibles en ligne. En plus de dictionnaires bilingues de grande taille, ces traducteurs utilisent des outils linguistiques importants (analyseur syntaxique et morphologique) afin d'obtenir la meilleure traduction. Jones (2005) constate que les résultats de la traduction sont relativement hachés et peu lisibles pour l'humain mais que ceci n'a que peu d'effet sur les systèmes de recherche d'information pour lesquels la seule chose importante est la pertinence des mots retournés comme traduction. On remarque

encore que les systèmes qui sont fondés sur des outils linguistiques complexes ont tendance à voir leur capacité de traduction se détériorer rapidement si le texte à traduire n'est pas syntaxiquement correct. Ces outils sont donc plus utiles dans le cas de traduction de documents ou si les requêtes sont suffisamment longues pour comporter des phrases complètes.

Traduction probabiliste

Le principe de base de la traduction probabiliste est de considérer que toute phrase (T) dans une langue est une traduction possible d'une autre phrase (S) écrite dans une autre langue. On peut alors calculer pour chaque paire de phrases (S, T) la valeur Pr(T|S), qu'on peut interpréter comme la probabilité qu'un traducteur nous donne T dans la langue cible si on lui présente S dans la langue source. On s'attend à ce que la paire (*Le chien mord le facteur* | *Paul loves Mary*) aura une très faible probabilité alors que (*Paul aime Marie* | *Paul loves Mary*) en aura une très forte (Brown *et al.* 1990). Le problème de la traduction automatique est donc de trouver, étant donnée une phrase cible T, la phrase source S qui a été traduite par T. Nous cherchons donc la phrase S qui maximise Pr(S|T). Partant de ce principe, différents systèmes sont proposés pour calculer les probabilités à partir de corpus alignés, et de là, construire la phrase S cherchée.

Un corpus bilingue aligné est une collection de documents dont chaque document existe en deux langues différentes soit parce que les documents d'une langue sont la traduction de l'autre, soit parce que les deux langues sont la traduction d'une troisième langue. De plus, il doit exister une correspondance entre les documents voire entre des parties plus petites (paragraphes ou phrases). Un exemple de corpus multilingue aligné est la Bible avec ses traductions en 320 langues et sa numérotation en versets de quelques phrases au maximum. Ce corpus n'est pas vraiment utilisé en traduction automatique car son vocabulaire souvent archaïque et sa forme lyrique l'éloigne beaucoup des textes contemporains. Un autre exemple de corpus multilingue aligné qui peut être utilisé dans les langues européennes est formé des procès-verbaux des séances du Parlement européen[4]. Dans ce cas, les textes, disponibles en dix à vingt langues selon la période, sont les interventions des parlementaires et le découpage en paragraphes est le même dans toutes les langues, mais les paragraphes ne sont pas étiquetés. Koehn (2003) en a tiré un corpus aligné.

Lors de la piste multilingue de la campagne CLEF 2004, Kamps *et al.* (2005) ont obtenu de bons résultats en utilisant la traduction probabiliste pour la traduction de la collection de documents.

[4] Disponibles en ligne : http://www.europarl.eu.int/activities/archive/cre.do
(visité le 20 novembre 2005)

Traduction multigrammes

Un multigramme (aussi noté n-gramme) est une suite de n caractères. Dans le cas de la traduction multigramme, on découpe la phrase en une suite de n-grammes. Par exemple, *Paul aime Marie* donne les 5-grammes suivants : 'Paul_', 'aul_a', 'ul_ai', 'l_aim', '_aime', 'aime_', 'ime_M', 'me_Ma', 'e_Mar', '_Mari' et 'Marie'. En suivant le même principe que la traduction probabiliste, on utilise un corpus aligné pour calculer les probabilités que le n-gramme T est la traduction du n-gramme S. Ensuite, on effectue la traduction en cherchant les n-grammes en langue cible qui ont la meilleure probabilité d'être la traduction de la langue source. Cette méthode a été utilisée en recherche d'information par McNamee *et al.* (2004) avec un certain succès.

Expansion de requête préalable par pseudo-rétroaction

D'autres recherches (Ballesteros *et al.* 1998, Kwok *et al.* 2001) proposent d'opérer une expansion de requête avant la traduction en effectuant une recherche dans une collection de documents dans la langue de la requête. Le résultat de cette recherche est utilisé pour extraire par peudo-rétroaction, par exemple à l'aide de la méthode de Rocchio (Buckley *et al.* 1995), un nombre donné de termes qui seront ajoutés à la requête à traduire. La nouvelle requête étant constituée de termes qui ne forment plus une phrase, les systèmes de traductions basés sur des éléments syntaxiques risquent de ne pas fonctionner correctement.

2.3 Évaluation en recherche d'information

Pour pouvoir comparer différents techniques de recherche d'information, il nous faut une méthode d'évaluation objective de l'efficacité des systèmes. Traditionnellement, deux types d'évaluations peuvent être considérés : d'une part, l'évaluation de la satisfaction de l'usager pour un système donné, et d'autre part, la capacité d'un système à classer les documents (évaluation du système). Le but étant de déterminer quel système est le plus à même de répondre au besoin d'information d'un utilisateur, la première méthode d'évaluation serait la plus adaptée. Mais cette méthode est très difficile à mettre en place. En effet, comme on ne veut évaluer que le système, il faut éviter les effets dus à l'interface ou à l'apprentissage des utilisateurs durant les tests. De plus, l'échantillon d'utilisateurs doit être suffisamment vaste. C'est pourquoi les différentes campagnes d'évaluation (TREC, CLEF et NTCIR[5]) évaluent l'efficacité des systèmes en considérant qu'un bon classement des documents est équivalent à une bonne performance de recherche (Vorhees 2002). À la fin des années 1960, des expériences réalisées à Cranfield (Cleverdon 1967) vont servir de modèle expérimental sous le nom de "paradigme de Cranfield" pour les campagnes

[5] http://trec.nist.gov/ (TREC), http://www.clef-campaign.org/ (CLEF) pour les langues européennes et http://research.nii.ac.jp/ntcir/ (NTCIR) pour les langues asiatiques.

d'évaluation, notamment l'idée de collection-test invariante entre les expériences, la mesure de l'efficacité basée sur la précision et le rappel, et trois hypothèses simplificatrices.

Collection-test

Une collection-test est formée d'un ensemble de documents, de requêtes présentant le besoin d'information et de jugements de pertinences établis par des experts du thème de la collection (l'aéronautique dans le cadre du corpus à Cranfield) indiquant quels sont les documents répondant effectivement à chaque requête. Si la collection est petite (un millier de documents), il est possible de juger la pertinence de chaque document pour chaque requête. Pour les collections-tests développés lors des campagnes d'évaluation TREC, CLEF ou NTCIR comprenant plusieurs centaines de milliers de documents, ceci n'est plus possible, le temps (et donc le coût) du jugement devenant trop important. Dans ce cas, une méthode nommée *pooling* est utilisée. Elle consiste à utiliser les résultats fournis par les participants à une campagne d'évaluation pour créer un sous-ensemble de documents par requête, lesquels seront jugés par les experts (Harman 1995). Les résultats fournis par chaque participant sont triés de manière à avoir en tête de liste, les documents dont la probabilité de pertinence est la plus grande pour le système du participant. Pour créer les sous-ensembles, on prend dans chaque résultat de participant, les X premiers documents (habituellement, X= 100). On obtient une liste de documents dont on élimine les doublons et qu'on trie par identificateur de document afin de ne pas donner d'informations aux juges qui pourraient biaiser leur jugement (rang du document, nombre de systèmes qui le retournent, identité de ces systèmes). Malgré le fait que certains documents pertinents peuvent ne pas avoir été jugés, Vorhees (2002) constate que cette manière de faire ne remet pas en cause la fiabilité des résultats dérivés des collections-tests concernées.

Mesure d'efficacité

Afin de comparer des systèmes de recherche d'information, il est nécessaire de disposer d'une mesure de son efficacité. Salton *et al.* (1983) proposent de calculer différentes valeurs à partir du tableau 3.

	Pertinent	Non pertinent	
Retourné	R_P	R_NP	R
Non retourné	NR_P	NR_NP	NR
	P	NP	P + NP = R + NR

Tableau 3 Cas possibles pertinent - retourné

Ces valeurs sont :

- Le rapport entre le nombre de documents pertinents retournés et le nombre de documents retournés, appelé **précision**. $\left(\frac{R\_P}{R}\right)$
- Le rapport entre le nombre de documents pertinents retournés et le nombre de documents pertinents, appelé **rappel**. $\left(\frac{R\_P}{P}\right)$
- Le rapport entre le nombre de documents non pertinents retournés et le nombre de documents non pertinents, appelé **bruit**. $\left(\frac{R\_NP}{NP}\right)$

Quelques constats sont déduits de ces définitions. D'abord, il est relativement simple d'obtenir un rappel de 100 %, il suffit de retourner la collection entière. De même, pour obtenir un bruit de 0 %, il suffit de ne rien retourner. Enfin, à mesure de l'augmentation du rappel, la précision aura tendance à décroître si la valeur de généralité n'est pas trop élevée (plus on retourne de documents pour augmenter le rappel, plus on risque d'ajouter des documents non pertinents).

Afin de tenir compte de ce dernier constat, on utilise un graphe représentant la valeur de précision pour un certain nombre de valeurs de rappel. Comme la collection-test comporte plusieurs requêtes, on calcule une moyenne (Braschler *et al.* 2004). Ces différentes paires de valeurs rappel/précision sont calculées à l'aide du logiciel trec_eval[6]. Un exemple de graphe rappel/précision est donné en figure 5.

[6] Mis à disposition par TREC : http://trec.nist.gov/

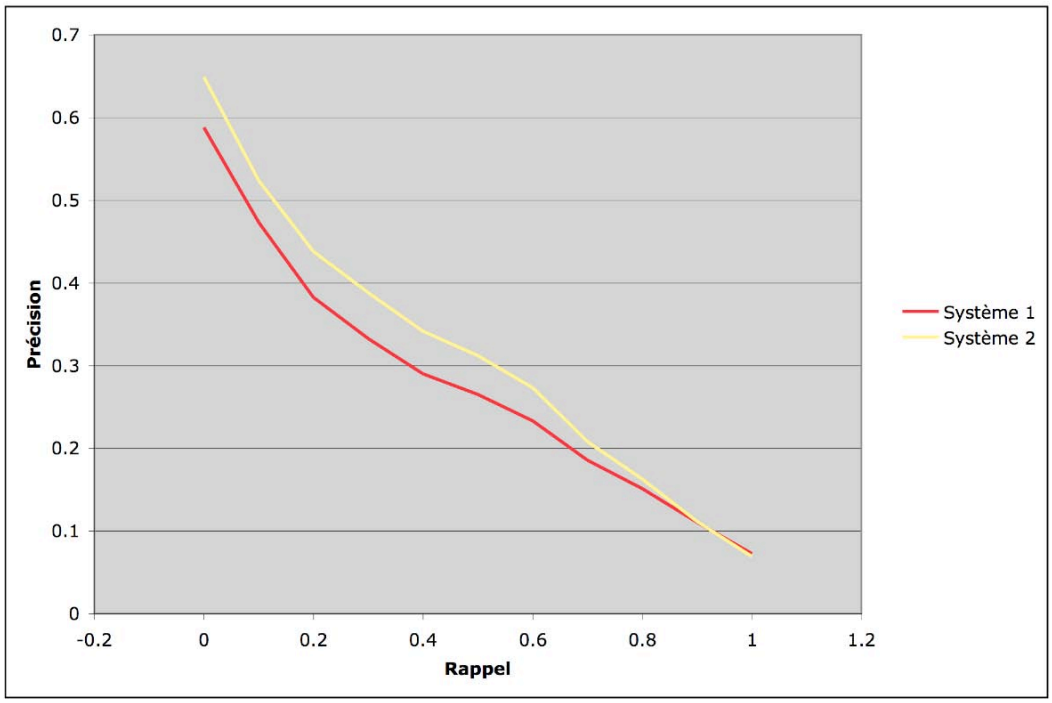

Figure 5 Exemple de graphe rappel/précision.

Ces graphes ne sont pas toujours très faciles à analyser, un système pouvant obtenir une meilleure précision pour de faibles valeurs de rappel et se trouver dépassé dès qu'on augmente le nombre de documents retournés. La solution proposée pour pallier ce problème est le calcul d'une valeur unique représentant la performance globale du système. Les différentes campagnes d'évaluation ont choisi la **précision moyenne** (Salton *et al.* 1983) calculée comme suit. Pour chaque requête, on détermine la précision chaque fois qu'on retourne un document pertinent. On calcule alors la moyenne de ces précisions. Par exemple, si sur dix documents retournés, nous avons des documents pertinents en rang 3, 4 et 9, nous calculons les trois précisions $p_{rang3} = 1/3$, $p_{rang4} = 2/4$ et $p_{rang9} = 3/9$. En prenant la moyenne de ces trois nombres, nous arrivons à 7/18 ou 0,39.

Ensuite, nous pouvons agréger les valeurs des différentes requêtes d'une même expérience en calculant la moyenne des précisions moyennes (*mean average precision* en anglais) que nous appellerons précision moyenne par simplification. Pour calculer ces précisions, nous utiliserons à nouveau le logiciel trec_eval. Cette mesure est utilisée par les campagnes d'évaluation TREC, CLEF et NTCIR[7].

Diverses autres mesures d'efficacité ont été proposées et utilisées pour des tâches spécifiques des campagnes d'évaluation ou d'autres conférences. On notera les méthodes basées uniquement sur le rang de la première réponse pertinente

[7] http://trec.nist.gov/ (TREC), http://www.clef-campaign.org/ (CLEF) pour les langues européennes et http://research.nii.ac.jp/ntcir/ (NTCIR) pour les langues asiatiques.

retournée (systèmes de question-réponse, QA@TREC, QA@CLEF, EQueR[8]), sur le recouvrement entre la réponse attendue et la réponse donnée (Medline@TREC Secondary Task) ou la précision après un nombre donné de documents retournés (WebCLEF).

Hypothèses simplificatrices (Cranfield)

Comme indiqué plus haut, trois hypothèses simplificatrices ont été définies lors des expériences de Cranfield. La première est de considérer que la pertinence peut être approximée par la similarité à la requête. Cette hypothèse implique que la pertinence d'un document est indépendante de celle d'un autre document, que tous les documents pertinents sont également désirables et enfin que le besoin d'information ne se modifie pas durant l'expérience. La deuxième hypothèse est de considérer qu'un jugement de pertinence d'un expert est représentatif de l'ensemble de la population, donc que la pertinence ne dépend pas de l'utilisateur. Enfin, la troisième hypothèse est celle de la complétude des listes de documents pertinents pour chaque requête, donc que l'on connaît tous les documents pertinents de la collection pour une requête donnée.

Cette dernière hypothèse n'est plus vérifiée pour les collections des campagnes d'évaluation TREC, CLEF et NTCIR. Cependant, Zobel (1998) a montré que des collections dont les jugements de pertinence sont créés à l'aide de la méthode du *pooling* ne défavorisent pas les systèmes qui ne sont pas jugés. Elles permettent donc des comparaisons entre différentes méthodes de recherche d'information.

On ajoute souvent une quatrième hypothèse, celle du choix de pertinence binaire (soit pertinent, soit non pertinent). Cette hypothèse n'était pas utilisée lors des expériences de Cranfield qui utilisaient cinq valeurs pour juger la pertinence (totalement pertinent, haut degré de pertinence, utile en général, d'intérêts minimal, non pertinent). Ce choix simplificateur a par contre l'avantage de permettre une formulation simple des mesures de précision, de rappel et de précision moyenne.

[8] http://www.technolangue.net/article61.html (visité le 21 décembre 2005).

3 Approches proposées

3.1 Objectifs

Les différents systèmes de traduction automatique dont nous disposons ont tous des lacunes, mais aucun n'est totalement mauvais. Nous verrons plus loin, dans le tableau 19, que chacun des systèmes de traduction que nous avons utilisés présente la meilleure traduction possible pour plusieurs requêtes.

En observant toutes les traductions pour différentes requêtes, nous constatons que certains traducteurs sont moins efficaces lorsqu'ils sont en présence de noms propres ou d'acronymes. D'autres sont très sensibles aux mots rares qui leur sont inconnus. Enfin, certains traducteurs "oublient" de traduire des parties de phrases qu'ils ne réussissent pas à analyser. Ces absences de traductions peuvent avoir des effets sensibles lorsqu'on est en présence de faux amis, car ces mots ont un sens différent dans la langue cible et provoquent le dépistage de documents ne correspondant pas à la requête.

Notre première approche va tenter de renforcer la traduction en éliminant des traductions erronées inhérentes au système de traduction (section 3.3). Dans ce dessein, nous proposons de retraduire mot par mot la requête dans la langue d'origine à l'aide d'un dictionnaire bilingue. Les mots dont aucune des traductions données par le dictionnaire bilingue ne se trouvent dans la requête originale sont alors retirés de la requête traduite. Nous espérons ainsi éliminer les erreurs les plus grossières comme les absences de traduction.

Notre deuxième approche (section 3.4) poursuit deux objectifs distinct. En premier lieu, nous voulons déterminer quelle méthode est la meilleure pour combiner plusieurs traductions pour une requête. En particulier, nous souhaitons vérifier si la méthode utilisée par notre équipe jusqu'à présent (concaténation des traductions) peut être surpassée. Ensuite, nous aimerions évaluer si la présence d'un terme dans plus d'une traduction de la même requête a une influence sur sa valeur pour la recherche d'information.

Nous avons remarqué que même les moins bons traducteurs automatiques pour l'ensemble des requêtes pouvaient être les meilleurs pour quelques requêtes spécifiques. Nous souhaitons donc sélectionner le meilleur traducteur possible pour chaque requête en nous basant sur des méthodes d'apprentissage automatique : la méthode des plus proches voisins (section 3.6), la génération automatique d'arbres de décision (section 3.7) et la régression logistique (section 3.8). Ces trois approches utilisent des données que nous avons déterminées pour chaque traduction de chaque requête et qui sont décrites en section 3.5.

Comme dernière approche, nous avons voulu nous affranchir de la préparation des requêtes en effectuant une recherche pour chaque traduction et en fusionnant les

résultats de ces différentes recherche en espérant que les résultats retournés par les meilleures traductions apparaîtront en tête de la liste fusionnée (section 3.9).

3.2 Environnement expérimental

Afin de d'évaluer différentes méthodes visant à améliorer la traduction dans un contexte de recherche d'information, nous avons utilisé des collections-test développées lors des campagnes d'évaluation CLEF, un logiciel d'indexation ainsi que différents traducteurs automatiques. Tous les traducteurs sont disponibles en ligne gratuitement.

3.2.1 Les collections-test (CLEF-2001 et CLEF-2002)

Pour nos différentes expériences, nous avons utilisé les collections-test développées lors des campagnes d'évaluation CLEF-2001 (Peters et *al.* 2002) et CLEF-2002 (Peters et *al.* 2003) pour les langues française, allemande, italienne et espagnole. Pour ces langues, les corpus de documents n'ont pas changé en 2001 et en 2002. Ceci nous permet de disposer d'un plus grand nombre de requêtes avec des jugements de pertinence pour nos expérimentations.

Les corpus de documents sont formés de textes d'informations générales tirées d'articles de journaux ou de dépêches d'agences de presse de l'année 1994. Les sources sont les journaux *Le Monde* (France), *La Stampa* (Italie), *Der Spiegel* et *Frankfurter Rundschau* (Allemagne) et les agences de presse *EFE* (Espagne) et *ATS* (Suisse, disponibles en allemand, français et italien). Quelques données statistiques concernant ces corpus sont présentées dans le tableau 4.

Corpus	Français	Italien	Allemand	Espagnol
Taille en MB	243	278	527	509
nb de doc.	87'191	108'578	225'371	215'738
nb de formes[9]	320'526	503'550	1'507'806	528'382
Nombre de formes par document				
moyenne	130,21	129,91	119,07	111,80
écart-type	109,15	97,60	109,73	55,40
médiane	95	92	89	99
maximum	1'622	1'394	2'420	642
minimum	3	1	1	5
Requêtes				
nb requêtes	99	96	99	99
nb doc. pert.	2'595	2'318	4'068	5'548
nb doc./requ.	26,21	24,15	41,09	56,04
nb max doc.	177	95	212	321
nb min doc.	1	2	1	1

Tableau 4 Quelques statistiques au sujet des corpus

[9] Les formes sont les termes indexés par le système d'information, donc les mots qui sont conservés après l'élimination des mots-outils et la racinisation.

Pour les campagnes CLEF 2001 et CLEF 2002, cent requêtes ont été générées. La langue originale de chaque requête n'est pas connue, mais toutes les requêtes ont été traduites dans toutes les langues des collections-test. Pour chaque collection-test, les requêtes n'ayant obtenu aucun document jugé pertinent sont retirées. Nous disposons donc de 99 requêtes pour le français, l'allemand et l'espagnol, et de 96 requêtes pour l'italien. Les requêtes sont formées sur le modèle de celles de TREC. En plus d'un numéro d'identification, chaque requête comprend un **titre** de quelques mots ne formant pas une phrase, une **description** formée d'une ou deux phrases exprimant plus précisément le besoin d'information ainsi qu'une **narration** décrivant ce qui sera considéré comme pertinent ou non pertinent. Le tableau 5 présente un exemple de requête.

Anglais	<num> C052 <EN-title> Chinese Currency Devaluation <EN-desc> Find documents describing the reasons and effects of the devaluation of Chinese currency. <EN-narr> Relevant documents discuss economic arguments in favour of and against the official reduction of the exchange value of the Chinese currency, and the social and economic consequences of the devaluation.
Français	<num> C052 <FR-title> Dévaluation de la monnaie chinoise <FR-desc> Trouvez des documents décrivant les raisons et les effets de la dévaluation de la monnaie chinoise. <FR-narr> Des documents pertinents discuteront des arguments économiques en faveur ou contre la dévaluation officielle de la valeur de change de la monnaie chinoise et les conséquences sociales et économiques de la dévaluation.
Italien	<num> C052 <IT-title> Svalutazione della moneta cinese <IT-desc> Reperisci documenti che descrivono i motivi e gli effetti della svalutazione della moneta cinese. <IT-narr> I documenti rilevanti discutono degli argomenti economici a favore e contro la riduzione ufficiale del valore di scambio della moneta cinese, e le conseguenze sociali ed economiche della svalutazione.
Allemand	<num> C052 <DE-title> Abwertung der chinesischen Währung <DE-desc> Suche Dokumente, die die Gründe und Folgen der Abwertung der chinesischen Währung beschreiben. <DE-narr> Relevante Dokumente behandeln die wirtschaftlichen Gründe für und gegen eine offizielle Senkung des Wechselkurses der chinesischen Währung sowie die sozialen und ökonomischen Folgen der Abwertung.
Espagnol	<num> C052 <ES-title> Devaluación de la moneda china. <ES-desc> Encontrar documentos que describan las razones y los efectos de la devaluación de la moneda china. <ES-narr> Los documentos relevantes discuten los argumentos económicos a favor o en contra de la devaluación oficial de la moneda china, así como las consecuencias sociales y económicas de esa devaluación.

Tableau 5 Exemple de requête dans les différentes langues

Pour toutes les expériences que nous avons menées, nous avons choisi l'anglais comme langue de départ. Nous avons aussi décidé de n'utiliser que les parties titre et description des requêtes.

3.2.2 L'indexation

Les indexations que nous avons effectuées ont été produites à l'aide du logiciel SMART (Salton 1971) auquel un certain nombre de fonctionnalités ont été ajoutées comme le modèle d'indexation probabiliste Okapi (Robertson et *al.* 2000). Les listes de mots-outils développées à l'Université de Neuchâtel[10] ont toujours été utilisées. Différents modèles d'indexation (voir section 1.2) ont été testés pour les langues des collections-test de CLEF 2001 avec les requêtes de CLEF 2001 et CLEF 2002. Les résultats de ces tests sont résumés dans le tableau 6.

	Précision moyenne (% de changement)				
Langue modèle	Anglais 89 requêtes	Français 99 requêtes	Italien 96 requêtes	Allemand 99 requêtes	Espagnol 99 requêtes
Okapi - npn	**53,21**	**50,74**	**43,93**	**38,97**	**54,78**
Lnu - ltc	51,51 (-3%)	48,01 (-5%)	41,78 (-5%)	37,13 (-5%)	51,69 (-6%)
dtu - dtn	48,28 (-9%)	48,05 (-5%)	41,24 (-6%)	37,01 (-5%)	49,92 (-9%)
atn - ntc	47,24 (-11%)	46,07 (-9%)	40,51 (-8%)	35,72 (-8%)	49,63 (-9%)
ltn - ntc	41,21 (-23%)	45,31 (-11%)	39,08 (-11%)	35,06 (-10%)	49,04 (-10%)
lnc - ltc	33,58 (-37%)	34,99 (-31%)	32,27 (-27%)	29,76 (-24%)	41,68 (-24%)
ltc - ltc	31,87 (-40%)	33,38 (-34%)	31,23 (-29%)	29,02 (-26%)	39,06 (-29%)
ntc - ntc	30,47 (-43%)	32,54 (-36%)	29,53 (-33%)	29,17 (-25%)	35,66 (-35%)
bnn - bnn	22,73 (-57%)	18,73 (-63%)	22,10 (-50%)	19,81 (-49%)	27,37 (-50%)
nnn - nnn	10,44 (-80%)	14,46 (-72%)	14,81 (-66%)	14,98 (-62%)	23,79 (-57%)

Tableau 6 Précision moyenne des différents modèles selon les cinq langues (requêtes « TD »)[11]

Nous constatons que parmi les modèles d'indexation que nous avons testés, le modèle Okapi nous permet d'obtenir les meilleurs résultats pour toutes nos langues. Nous avons donc décidé de n'utiliser que le modèle Okapi pour les différentes expérimentations que nous avons conduites.

[10] Disponibles en ligne sur http://www.unine.ch/info/clef/ (visité le 20 février 2006).

[11] Les formules de pondération des modèles sont données en Annexe 1.

3.2.3 Les systèmes de traduction

Nous nous situons dans le contexte de la recherche bilingue d'information avec traduction automatique des requêtes. Nous utilisons la langue anglaise comme langue d'origine pour les requêtes. Nous nommerons l'anglais "langue source" dans la suite du texte. Les requêtes sont traduites à l'aide de cinq systèmes de traductions automatiques ainsi qu'un dictionnaire bilingue. Tous ces outils sont disponibles gratuitement en ligne, la liste en est donnée dans le tableau 7.

Traducteurs	
REVERSO ONLINE™	http://translation2.paralink.com
SYSTRAN™	http://babel.altavista.com/translate.dyn
GOOGLE™	http://www.google.com/language_tools
FREETRANSLATION™	http://www.freetranslation.com
INTERTRAN™	http://www.tranexp.com:2000/InterTran
Dictionnaire bilingue	
BABYLON™	http://www.babylon.com

Tableau 7 Liste des traducteurs utilisés avec leur adresse

Dans le cas de BABYLON™, nous effectuons une traduction mot à mot de la requête. Pour chaque mot, le dictionnaire nous propose un certain nombre de traductions. Nous avons décidé de procéder à trois évaluations, en retenant :

- le premier mot retourné (évaluation notée « Babylon 1 »);
- les deux premiers mots retournés (évaluation notée « Babylon 2 »);
- ou les trois premiers mots retournés (évaluation notée « Babylon 3 »).

Nous disposons donc de huit traductions automatiques différentes (cinq traducteurs et trois versions du dictionnaire bilingue) que nous allons utiliser dans nos diverses expériences.

	Précision moyenne (% de changement)			
collection modèle	Français 99 requêtes	Italien 96 requêtes	Allemand 99 requêtes	Espagnol 99 requêtes
Manuelle	50,74	43,93	38,97	54,78
Reverso	45,25 (-10,8 %)	—	**30,74** (-21,1 %)	**46,13** (-15,8%)
Systran	44,64 (-12,0 %)	31,62 (-28,0 %)	29,71 (-23,8 %)	40,46 (-26,1%)
Google	44,80 (-11,7 %)	31,63 (-28,0 %)	29,99 (-23,0 %)	40,33 (-26,4%)
FreeTrans	41,05 (-19,1 %)	31,80 (-27,6 %)	26,51 (-32,0 %)	40,98 (-25,2%)
InterTran	38,76 (-23,6 %)	29,81 (-32,1 %)	22,33 (-42,7 %)	38,21 (-30,2%)
Babylon 1	**47,65** (- 6,1 %)	**32,21** (-26,7 %)	27,15 (-30,3 %)	38,97 (-28,9%)
Babylon 2	43,45 (-14,4 %)	28,84 (-34,4 %)	26,74 (-31,4 %)	34,37 (-37,3%)
Babylon 3	42,26 (-16,7 %)	27,50 (-37,4 %)	25,52 (-34,5 %)	32,34 (-40,1%)
Meilleur	54,48 (+ 7,4 %)	42,26 (- 3,8 %)	41,43 (+ 6,3 %)	52,40 (-4,3%)

Tableau 8 Précision moyenne des différents outils de traduction selon les langues (modèle Okapi, requête « TD »)

La précision moyenne obtenue par le modèle Okapi avec les quatre langues et selon les différents systèmes de traduction est indiquée dans le tableau 8. Comme première ligne de ce tableau, nous avons repris la traduction manuelle des requêtes et cette valeur va nous servir de référence afin de calculer les pourcentages de différence. On notera que cette traduction manuelle présente, quelle que soit la langue, la meilleure approche. En effet, aucun des systèmes de traduction automatique ne présente une performance moyenne supérieure à celle obtenue par les traductions manuelles.

Au niveau des outils de traduction automatique, aucune des solutions étudiées n'apporte systématiquement la meilleure performance, quelle que soit la langue considérée. On peut toutefois signaler que le dictionnaire bilingue BABYLON propose le meilleur outil pour le français et l'italien (en prenant uniquement la première traduction proposée). Pour l'espagnol comme pour l'allemand, le système REVERSO occupe la première place au niveau de la performance.

Finalement, si l'on retient pour chaque requête la meilleure traduction automatique, nous obtenons la précision moyenne indiquée dans la dernière ligne du tableau 8 (ligne débutant par « Meilleur »). La performance d'un tel système automatique permet presque d'atteindre celle d'une traduction manuelle pour les langues italienne et espagnole et propose une précision moyenne légèrement meilleure pour les langues française et allemande (puisque nous sommes légèrement au-dessus des 5 % de différence). En se basant sur des connaissances a posteriori, nous pourrions donc atteindre, lors d'une interrogation bilingue, une performance similaire à celle d'une interrogation unilingue, pour les langues française, italienne, allemande ou espagnole.

3.3 Amélioration par traduction aller-retour

Les différentes traductions automatiques que nous avons obtenues comportent parfois des erreurs qui semblent facilement détectables. Certains mots ne sont pas traduits et sèment la confusion parce que ce sont des faux amis. Par exemple, le mot anglais *confidence* signifie *confiance* en français et non *secret*. D'autres mots donnent des traductions qui n'ont aucun sens dans la langue cible dans le contexte de la phrase. Par exemple, la phrase anglaise *Find all information about the tainted blood trials in France including the sentences given by the court and the names of the people found guilty* tirée de la requête C057 est traduite en français par GOOGLE™ de la manière suivante : *Trouvez toutes les informations sur les épreuves corrompues de sang en France comprenant les phrases données par la cour et les noms des personnes trouvées coupables*. Les trois mots soulignés sont ceux qui posent problème. Si on substitue ces trois mots par les traductions correctes (*procès*, *contaminé* et *sentences* dans l'ordre), la précision moyenne pour cette requête seule passe de moins de 2 % à 58 %.

Dans cette première approche, nous voulons éviter ces erreurs en reprenant chaque requête traduite mot à mot et en utilisant un dictionnaire bilingue pour

effectuer la traduction inverse. Si une des traductions proposées par le dictionnaire est présente dans la requête source, nous conservons le mot traduit. Si la traduction inverse ne donne rien, on élimine le mot de la requête.

Afin d'améliorer l'accès au dictionnaire, chaque mot sera lemmatisé avant consultation. La lemmatisation est le processus qui permet d'obtenir à partir de tout mot d'une phrase l'entrée correspondante dans le dictionnaire. Le tableau 9 présente un exemple de lemmatisation d'une requête en anglais. Pour éviter les erreurs dues aux marques du pluriel ou du féminin ainsi qu'à la conjugaison des verbes, nous avons aussi essayé de lemmatiser les requêtes en langue source avant le contrôle. La lemmatisation a été effectuée à l'aide du logiciel TREETAGGER (TreeTagger Projekt).

	Requête source	**Requête lemmatisée**
Titre	Final Four Results	Final Four result
Description	Find documents giving the results of the European Basketball Final Four.	find document give the result of the European basketball Final Four.
Narration	Relevant documents will give details on the results of at least one of the three matches (two semi-finals and one final) of the final phase of the European basketball championship. Documents written prior to the semi-finals that give the names of possible winners are not relevant.	relevant document will give detail on the result of at least one of the three match (two semi-final and one final) of the final phase of the European basketball championship. document write prior to the semi-final that give the name of possible winner be not relevant.

Tableau 9 Exemple de lemmatisation pour la requête C054.

Nous avons évalué cette approche avec le système bilingue anglais-espagnol (requêtes en anglais, collection en espagnol). Le dictionnaire utilisé pour le contrôle est celui proposé par BABYLON que nous avons aussi utilisé pour la traduction anglais-espagnol. Nous nommerons « contrôle normal » les expériences utilisant les requêtes source sans traitement pour la traduction inverse et « contrôle lemmatisé » lorsque nous utilisons les requêtes lemmatisées. Le tableau 10 présente quelques statistiques concernant les nombres de mots retirés ou conservés pour chaque traducteur et pour les cent requêtes. Les colonnes contiennent les données suivantes :

- Mots : le nombre total de mots dans les cent requêtes traduites.
- Absents : le nombre de mots qui n'ont pas d'entrée en espagnol dans notre dictionnaire.
- Retirés : le nombre de mots retirés parce qu'aucune traduction anglaise du mot espagnol se trouve dans la version source.
- Conservés : le nombre de mots dont une correspondance a été trouvée.

Ces deux dernières colonnes existent en deux versions, l'une avec le fichier source sans prétraitement, la deuxième avec le fichier source lemmatisé.

Les pourcentages indiqués dans le tableau représentent la proportion du nombre de mots qui se trouvent dans chaque catégorie traducteur par traducteur.

			Contrôle normal		Contrôle lemmatisé	
Traducteur	**Mots**	**Absents**	**Retirés**	**Conservés**	**Retirés**	**Conservés**
Reverso	1947	138 (7,1%)	638 (32,8%)	1171 (60,1%)	371 (19,0%)	1438 (73,9%)
Systran	2032	149 (7,3%)	656 (32,3%)	1227 (60,4%)	388 (19,1%)	1495 (73,6%)
Google	2072	149 (7,2%)	669 (32,3%)	1254 (60,5%)	400 (19,3%)	1523 (73,5%)
FreeTrans	2146	149 (6,9%)	705 (32,8%)	1292 (60,2%)	420 (19,6%)	1577 (73,5%)
InterTran	1775	171 (9,6%)	522 (29,4%)	1082 (61,0%)	242 (13,6%)	1362 (76,7%)
Babylon 1	1751	111 (6,3%)	678 (38,7%)	962 (54,9%)	378 (21,6%)	1262 (72,1%)
Babylon 2	3364	186 (5,5%)	1587 (47,2%)	1591 (47,3%)	1163 (34,6%)	2015 (59,9%)
Babylon 3	4705	230 (4,9%)	2328 (49,5%)	2127 (45,6%)	1801 (38,3%)	2674 (56,8%)

Tableau 10 Nombre de mots conservés par la traduction inverse pour chaque traducteur (TD).

En étudiant les statistiques des mots absents ou conservés, nous constatons qu'en moyenne, aucun traducteur automatique ne se démarque de manière importante que ce soit en utilisant le contrôle normal ou le contrôle lemmatisé. Le cas de BABYLON est un peu particulier puisque le même traducteur est utilisé dans les deux sens. Mais même pour celui-ci, plus de la moitié des mots absents sont des mots espagnols corrects qui ont été traduits depuis l'anglais et qui ne se trouvent pas comme entrée dans le dictionnaire espagnol-anglais.

En termes de précision, les résultats sont présentés dans le tableau 11. La base représente la précision moyenne obtenue par les requêtes traduites sans traitements supplémentaires, telle que présentée dans le tableau 8. Pour les traducteurs automatiques, si nous ne gardons que les mots dont une correspondance a été trouvée en anglais, nous perdons plus de 50 % de la précision moyenne en comparaison avec la base dans le cas du contrôle normal. Le cas du contrôle lemmatisé est légèrement meilleur que le précédent mais représente tout de même une baisse de la précision moyenne de plus de 36 %. Le dictionnaire bilingue donne des résultats similaires bien que la baisse soit plus faible. La précision de base de BABYLON étant plus basse, la précision obtenue ne dépasse pas celle des meilleurs traducteurs automatiques (REVERSO et FREETRANSLATION).

	Précision moyenne (différence avec la base)				
		Contrôle normal		**Contrôle lemmatisé**	
Traducteur	**Base**	**Absents retirés**	**Avec Absents**	**Absents retirés**	**Avec Absents**
Reverso	46,13	23,00 (-50,1%)	35,64 (-22,7%)	28,86 (-37,4%)	40,69 (-11,8%)
Systran	40,46	19,10 (-52,8%)	31,13 (-23,1%)	24,95 (-38,3%)	36,50 (- 9,8%)
Google	40,33	18,97 (-53%)	31,01 (-23,1%)	24,30 (-39,7%)	36,19 (-10,3%)
FreeTrans	40,98	20,22 (-50,7%)	34,25 (-16,4%)	24,02 (-41,4%)	37,65 (- 8,1%)
InterTran	38,21	18,60 (-51,3%)	29,50 (-22,8%)	24,47 (-36%)	34,30 (-10,2%)
Babylon 1	38,97	21,01 (-46,1%)	33,28 (-14,6%)	26,00 (-33,3%)	37,48 (- 3,8%)
Babylon 2	34,37	21,05 (-38,7%)	33,07 (- 3,8%)	24,55 (-28,6%)	35,76 **(+ 4%)**
Babylon 3	32,34	20,01 (-38,1%)	31,61 (- 2,3%)	23,12 (-28,5%)	33,40 **(+ 3,3%)**

Tableau 11 Précision moyenne pour la traduction inverse (TD)

En analysant les mots absents, nous avons constaté qu'environ la moitié d'entre eux sont des noms propres ("Miguel Indurain") ou des locutions étrangères ("Tour de France") qui n'ont pas été traduits d'anglais en espagnol et qui ont probablement une grande importance dans la recherche. Nous avons donc essayé de les conserver. Les résultats de ces expériences sont donnés dans les colonnes "Avec Absents" du tableau 11. La détérioration de la précision est moindre, mais elle reste tout de même notable avec environ -20 % pour le contrôle normal et -10 % pour le contrôle lemmatisé. La seule exception reste BABYLON, mais comme pour la méthode précédente, sa meilleure performance est compensée par son score de base plus faible.

En cherchant à comprendre les raisons de cette détérioration, nous avons constaté qu'une des faiblesses principales de ce système consiste en sa tendance à supprimer des mots. En effet, dans certains cas, les mots restants dans la requête sont tous des mots-outils, ce qui détériore la précision puisque aucun document ne sera retourné. Par exemple, la requête C077 avec le traducteur FREETRANS et le contrôle normal se comporte de cette manière. Elle est présentée dans le tableau 12.

	Anglais	Traduit	Filtré
Titre	Teenage Suicides	Los Suicidios de Teenage	(vide)
Description	What information is available concerning teenage suicides?	Qué está disponible con respecto a teenage ¿Los suicidios?	Qué con

Tableau 12 Traduction et filtrage de la requête C077 (TD).

Un autre problème est l'usage de mots espagnols dans les requêtes anglaises comme "El Niño" dans le sens de perturbation climatique. Le traducteur conserve ce mot tel quel, mais on le supprime lors de la traduction inverse puisque le dictionnaire bilingue proposera *kid*, *baby* ou encore *boy* comme possibilités ce qui ne correspond pas au contenu de la requête source. La requête amputée de "El Niño" (C043) devient tellement générale que tout document parlant de phénomène météorologique est retourné, ce qui ne permet pas de dépister correctement les explications d'un de ces phénomènes.

Ensuite, les noms propres posent un réel problème au dictionnaire bilingue. Si les pays sont habituellement traduits correctement dans les deux sens, certains noms de personnes ou de villes sont traduits de l'anglais à l'espagnol, mais le dictionnaire bilingue n'est pas capable de faire la traduction inverse. C'est par exemple le cas entre l'anglais et le français pour Gorbatchov traduit par Gorbatchev et qui reste Gorbatchev dans la traduction inverse.

Enfin, le dictionnaire bilingue est perfectible. Certains mots correctement traduits mais relativement rares en espagnol ne sont pas trouvés par le dictionnaire espagnol-anglais et par conséquent éliminés de la requête. D'autres ont plusieurs significations et certaines ont été omises par le dictionnaire.

En conclusion, cette méthode dépend trop du dictionnaire bilingue pour donner des résultats probants. D'une manière générale, les quelques améliorations obtenues pour certaines requêtes sont largement compensées par la détérioration de la précision moyenne des autres requêtes.

3.4 Divers systèmes de combinaison de traductions

Les traducteurs automatiques peuvent nous fournir des résultats très différents pour une même requête. Or, chaque traducteur a ses points forts et ses points faibles et aucun ne permet d'obtenir le meilleur résultat pour toutes les requêtes[12]. Notre deuxième approche cherche donc à combiner les différentes traductions d'un même jeu de requêtes. Une requête étant par essence une description imprécise du besoin d'information, la méthode usuelle pour combiner les différentes traductions d'une même requête est la concaténation simple (ajout bout à bout des différentes traductions). Le tableau 13 propose un exemple de cette technique pour trois traducteurs.

Babylon 1	Titre	noruego referendum en funcionamiento union europea
	Description	estar el reaccion interior el descanso de europa a el negativa resultado de el noruego referendum interior cual noruega decidido contra membresia interior el europeo union union europea
Google	Titre	referendum noruego en el eu
	Description	cuales eran las en el resto de europa a los resultados negativos del referendum noruego en el cual noruega decidia contra calidad de miembro en la union europea (eu).
Reverso	Titre	referendum noruego a union europea
	Description	que era las en el resto de europa a el los resultados negativos del referendum noruego en el que noruega decidio contra socios en la union europea (union europea).
Concaté-nation	Titre	noruego referendum en funcionamiento union europea referendum noruego en el eu referendum noruego a union europea
	Description	estar el reaccion interior el descanso de europa a el negativa resultado de el noruego referendum interior cual noruega decidido contra membresia interior el europeo union union europea cuales eran las en el resto de europa a los resultados negativos del referendum noruego en el cual noruega decidia contra calidad de miembro en la union europea (eu). que era las en el resto de europa a el los resultados negativos del referendum noruego en el que noruega decidio contra socios en la union europea (union europea).

Tableau 13 Exemple de concaténation de traductions (Requête C073, TD)

Notre but est d'essayer différentes autres techniques de combinaison pour déterminer si elles permettent d'améliorer la performance de recherche. Ayant constaté que certains mots se trouvent dans plusieurs traductions, nous souhaitons particulièrement vérifier si le nombre de traductions dans lesquelles un mot

[12] Nous développerons ce point dans la section 3.5 *Disparité entre les requêtes*.

apparaît a une importance pour le résultat de la recherche d'information. Dans ce dessein, nous allons fixer un seuil, nommé *limite*, qui représente le nombre minimal de traductions différentes contenant un terme donné pour que celui-ci soit intégré dans la requête combinée. Lors du décompte des traductions, le titre et la description sont considérés séparément.

Ensuite, comme le modèle probabiliste Okapi que nous utilisons pour nos expérimentations tient compte de la fréquence des termes dans la requête, nous allons essayer plusieurs méthodes pour décider du nombre de répétition de chaque terme dans la requête combinée. Nous décrirons dans les paragraphes suivants l'idée générale de chacune de ces méthodes. Le tableau 14 en présente un résumé succinct. Dans ce tableau, lorsqu'on parle de rang, on considère que les traductions sont triées, pour chaque terme, dans l'ordre décroissant du nombre d'occurrences de ce terme. Dans le cas des moyennes, les valeurs obtenues sont arrondies vers le haut.

Nom	**Nombre de répétitions du terme dans la requête combinée**
one	Une seule répétition de chaque terme.
min	Le nombre minimal de répétitions parmi les traductions proposant ce terme.
max	Le nombre maximal de répétitions.
lim	Le nombre de répétitions contenues dans la traduction de rang *limite*.
avg	La moyenne de tous les nombres de répétitions y compris si le terme est absent.
avg_lim	La moyenne de tous les nombres de répétitions parmi les traductions de rang inférieur à *limite*.
avg_gz	La moyenne de tous les nombres de répétitions parmi les traductions qui contiennent le terme.
add	La somme de tous les nombres de répétitions (simple concaténation des requêtes).
add_lim	La somme de tous les nombres de répétitions parmi les traductions de rang inférieur à *limite*.

Tableau 14 Méthodes de combinaison

Le tableau 15 présente un exemple fictif de combinaisons à partir de trois requêtes : "Europe Europe Europe", "Union Europe Europe" et "Union Europe Suisse". Le tableau 16 présente un exemple plus complet basé sur une requête réelle.

	Limite		
	1	2	3
min	Europe Union Suisse	Europe Union	Europe
max	Europe Europe Europe Union Suisse	Europe Europe Europe Union	Europe Europe Europe
lim	Europe Europe Europe Union Suisse	Europe Europe Union	Europe
avg	Europe Europe Union Suisse	Europe Europe Union	Europe Europe
avg_lim	Europe Europe Europe Union Suisse	Europe Europe Europe Union	Europe Europe
avg_gz	Europe Europe Europe Union Suisse	Europe Europe Europe Union	Europe Europe
add	Europe Europe Europe Union Europe Europe Union Europe Suisse	Europe Europe Europe Union Europe Europe Union Europe	Europe Europe Europe Europe Europe Europe
add_lim	Europe Europe Europe Union Suisse	Europe Europe Europe Union Europe Europe Union	Europe Europe Europe Europe Europe Europe

Tableau 15 Exemple fictif de combinaisons

Dans notre première méthode, on considère que la seule chose importante est la présence d'un terme dans un nombre de traductions correspondant au moins à la *limite*. En conséquence, chaque terme n'est inclus qu'une seule fois dans la requête combinée et les duplicata sont supprimés. Nous avons nommé cette méthode *one*.

La méthode *max* part du principe qu'un terme doit avoir autant d'importance que celle que lui donne la traduction qui le contient le plus de fois. On inclut donc dans la requête combinée autant de répétitions de ce terme que le nombre maximal d'occurrences parmi toutes les traductions. À l'inverse, la méthode *min* considère qu'un terme ne doit pas être plus important que le poids attribué par la traduction qui lui donne le moins d'importance. Le nombre d'occurrences de ce terme correspond donc au nombre minimal de répétitions parmi les traductions proposant ce terme.

Après le minimum et le maximum qui évaluent les extrêmes, nous voulions une mesure de tendance centrale. Nous avons choisi la moyenne arithmétique. La méthode *avg* considère la moyenne des occurrences du terme dans toutes les traductions, y compris en cas d'absence. Ceci nous semble entrer quelque peu en contradiction avec le principe du seuil introduit par la *limite*, car même si un terme dépasse le seuil, son importance est péjorée s'il n'est présent que dans peu de traductions. Pour éviter cet effet, nous avons utilisé deux techniques. D'une part, ne calculer la moyenne que sur les traductions qui contiennent le terme (*avg_gz*), d'autre part, limiter le nombre de traductions incluses dans la moyenne à la valeur de *limite* (*avg_lim*). Dans ce dernier cas, si le terme est présent dans un plus grand nombre de traductions, on choisit les traductions qui contiennent le plus de fois le terme.

La concaténation étant la méthode la plus courante de combinaison, nous avons décidé d'y recourir dans les méthodes *add* et *add_lim*, cette dernière fonctionnant de manière analogue à *avg_lim* en utilisant la concaténation au lieu de la moyenne.

Enfin, avec notre dernière méthode nommée *lim*, nous choisissons les répétitions présentes dans la traduction de rang *limite* en considérant pour chaque terme les traductions dans l'ordre décroissant du nombre d'occurrences de ce terme.

Le tableau 16 présente le résultat de nos différentes méthodes de combinaison pour la requête C073 (exemple donné dans le tableau 13) pour des valeurs de *limite* de 1, 2 et 3. Comme le modèle d'indexation que nous utilisons dans nos expérimentations (Okapi) ne tient pas compte de l'ordre des termes dans la requête, ceux-ci sont donnés dans l'ordre alphabétique.

		Limite		
		1	2	3
one	T	a el en eu europea funcionamiento noruego referendum union	en europea noruego referendum union	noruego referendum
	D	a calidad contra cual cuales de decidia decidido decidio del descanso el en era eran estar eu europa europea europeo interior la las los membresia miembro negativa negativos noruega noruego que reaccion referendum resto resultado resultados socios union	a contra cual de del el en europa europea la las los negativos noruega noruego referendum resto resultados union	a contra de el europa europea noruega noruego referendum union
min	T	a el en eu europea funcionamiento noruego referendum union	en europea noruego referendum union	noruego referendum
	D	a calidad contra cual cuales de decidia decidido decidio del descanso el el en en en era eran estar eu europa europea europeo interior interior interior la las los membresia miembro negativa negativos noruega noruego que que reaccion referendum resto resultado resultados socios union	a contra cual de del el el en en en europa europea la las los negativos noruega noruego referendum resto resultados union	a contra de el el europa europea noruega noruego referendum union
max	T	a el en eu europea funcionamiento noruego referendum union	en europea noruego referendum union	noruego referendum
	D	a calidad contra cual cuales de de decidia decidido decidio del descanso el el el el en en en era eran estar eu europa europea europea europeo interior interior interior la las los membresia miembro negativa negativos noruega noruego que que reaccion referendum resto resultado resultados socios union union	a contra cual de de del el el el el en en en europa europea europea la las los negativos noruega noruego referendum resto resultados union union	a contra de de el el el el el europa europea europea noruega noruego referendum union union
lim	T	a el en eu europea funcionamiento noruego referendum union	en en europea europea noruego referendum union union	noruego referendum
	D	a calidad contra cual cuales de de decidia decidido decidio del descanso el el el el el en en en era eran estar eu europa europea europea europeo interior interior interior la las los membresia miembro negativa negativos noruega noruego que que reaccion referendum resto resultado resultados socios union union	a contra cual de de del el el el el en en en europa europea europea la las los negativos noruega noruego referendum resto resultados union union	a contra de el el europa europea noruega noruego referendum union
avg	T	a el en eu europea funcionamiento noruego referendum union	en europea noruego referendum union	noruego referendum
	D	a calidad contra cual cuales de de decidia decidido decidio del descanso el el el el en en en era eran estar eu europa europea europea europeo interior la las los membresia miembro negativa negativos noruega noruego que reaccion referendum resto resultado resultados socios union union	a contra cual de de del el el el el en en en europa europea europea la las los negativos noruega noruego referendum resto resultados union union	a contra de de el el el el europa europea europea noruega noruego referendum union union
avg_lim	T	a el en eu europea funcionamiento noruego referendum union	en en europea europea noruego noruego noruego referendum referendum referendum union union	noruego referendum
	D	a calidad contra cual cuales de de decidia decidido decidio del descanso el el el el el en en en era eran estar eu europa europea europea europeo interior interior interior la las los membresia miembro negativa negativos noruega noruego que que reaccion referendum resto resultado resultados socios union union	a contra cual de de del el el el el en en en europa europea europea la las los negativos noruega noruego referendum resto resultados union union	a contra de el el europa europea noruega noruego referendum union

Tableau 16 Exemple de nos méthodes de combinaison pour la requête C073 (TD)

		Limite		
		1	2	3
avg_gz	T	a el en eu europea funcionamiento noruego referendum union	en europea noruego referendum union	noruego referendum
	D	a calidad contra cual cuales de de decidia decidido decidio del descanso el el el el en en en era eran estar eu europa europea europea europeo interior interior interior la las los membresia miembro negativa negativos noruega noruego que que reaccion referendum resto resultado resultados socios union union	a contra cual de de del el el el el en en en europa europea europea la las los negativos noruega noruego referendum resto resultados union union	a contra de de el el el el europa europea europea noruega noruego referendum union union
add	T	a el en en eu europea europea funcionamiento noruego noruego noruego referendum referendum referendum union union	en en europea europea noruego noruego noruego referendum referendum referendum union union	noruego noruego noruego referendum referendum referendum
	D	a a a calidad contra contra contra cual cual cuales de de de de de decidia decidido decidio del del descanso el el el el el el el el el en en en en en era eran estar eu europa europa europa europea europea europea europea europeo interior interior interior la la las las los los membresia miembro negativa negativos negativos noruega noruega noruega noruego noruego noruego que que reaccion referendum referendum referendum resto resto resultado resultados resultados socios union union union union union union	a a a contra contra contra cual cual cual de de de de de de del del el el el el el el el el el en en en en en en europa europa europa europea europea europea europea la la las las los los negativos negativos noruega noruega noruega noruego noruego noruego referendum referendum referendum resto resto resultados resultados union union union union union union	a a a contra contra contra de de de de de de el el el el el el el el el el el el europa europa europa europea europea europea europea noruega noruega noruega noruego noruego noruego referendum referendum referendum union union union union union union
add_lim	T	a el en eu europea funcionamiento noruego referendum union	en en europea europea noruego noruego referendum referendum union union	noruego noruego noruego referendum referendum referendum
	D	a calidad contra cual cuales de de decidia decidido decidio del descanso el el el el el en en en era eran estar eu europa europea europea europeo interior interior interior la las los membresia miembro negativa negativos noruega noruego que que reaccion referendum resto resultado resultados socios union union	a a contra contra cual cual de de de de del del el el el el el el el el el en en en en en en en europa europa europea europea europea la la las las los los negativos negativos noruega noruega noruego noruego referendum referendum resto resto resultados resultados union union union union	a a a contra contra contra de de de de de de el el el el el el el el el el el el europa europa europa europea europea europea europea noruega noruega noruega noruego noruego noruego referendum referendum referendum union union union union union union

Tableau 16 Exemple de nos méthodes de combinaison pour la requête C073 (TD)

Nous constatons que certaines de ces méthodes sont les mêmes pour certaines valeurs de *limite*. Ainsi, pour une *limite* de valeur un, les méthodes *max*, *lim* (le plus grand de un élément) et *avg_lim* (la moyenne du plus grand élément) sont égales. Si la *limite* est égale au nombre de traductions différentes, la méthode *lim* donne la même requête que *min*, de même qu'*avg*, *avg_lim* et *avg_gz* d'une part et *add* et *add_lim* d'autre part. Enfin, la méthode *add* avec une *limite* de un est égale à la concaténation simple.

Afin d'évaluer ces différentes méthodes, nous avons travaillé avec les collections-tests espagnoles et allemandes en partant des requêtes anglaises. Ces

collections sont décrites dans la section 3.2.1. Toutes les expériences ont toujours été menées avec tous nos systèmes de traduction, soit huit au total.

Le tableau 17 présente les valeurs de précision moyenne obtenues pour différentes valeurs de *limite* avec le corpus espagnol. Les pourcentages d'améliorations par rapport au meilleur système de traduction, REVERSO (précision moyenne de 46,13 dans le tableau 8) sont indiquées entre parenthèses. La valeur en **gras** est la meilleure combinaison pour une valeur de *limite* donnée.

limite	1	3	4	5	8
one	38,37 (-16,8)	43,08 (- 6,6)	43,21 (- 6,3)	38,10 (-17,4)	24,48 (-46,9)
min	37,48 (-18,7)	42,55 (- 7,8)	43,23 (- 6,3)	38,17 (-17,3)	24,45 (-47,0)
max	38,07 (-17,5)	42,53 (- 7,8)	43,19 (- 6,4)	38,07 (-17,5)	**24,65 (-46,6)**
lim	38,07 (-17,5)	42,47 (- 7,9)	43,16 (- 6,4)	38,10 (-17,4)	24,45 (-47,0)
avg	25,48 (-44,8)	25,48 (-44,8)	25,84 (-44,0)	25,84 (-44,0)	24,45 (-47,0)
avg_lim	38,07 (-17,5)	42,50 (- 7,9)	43,21 (- 6,3)	38,10 (-17,4)	24,45 (-47,0)
avg_gz	37,48 (-18,7)	42,50 (- 7,9)	43,19 (- 6,4)	38,15 (-17,3)	24,45 (-47,0)
add	**47,31 (+ 2,6)**	**46,82 (+ 1,3)**	**43,71 (- 5,2)**	37,79 (-18,1)	24,61 (-46,6)
add_lim	38,07 (-17,5)	42,83 (- 7,1)	43,58 (- 5,5)	**38,38 (-16,8)**	24,61 (-46,6)

Tableau 17 Précision moyenne selon les valeurs de limite et les méthodes de combinaison (espagnol)

La figure 6 présente graphiquement pour chaque méthode l'évolution de la précision moyenne avec l'augmentation de la valeur de la *limite*.

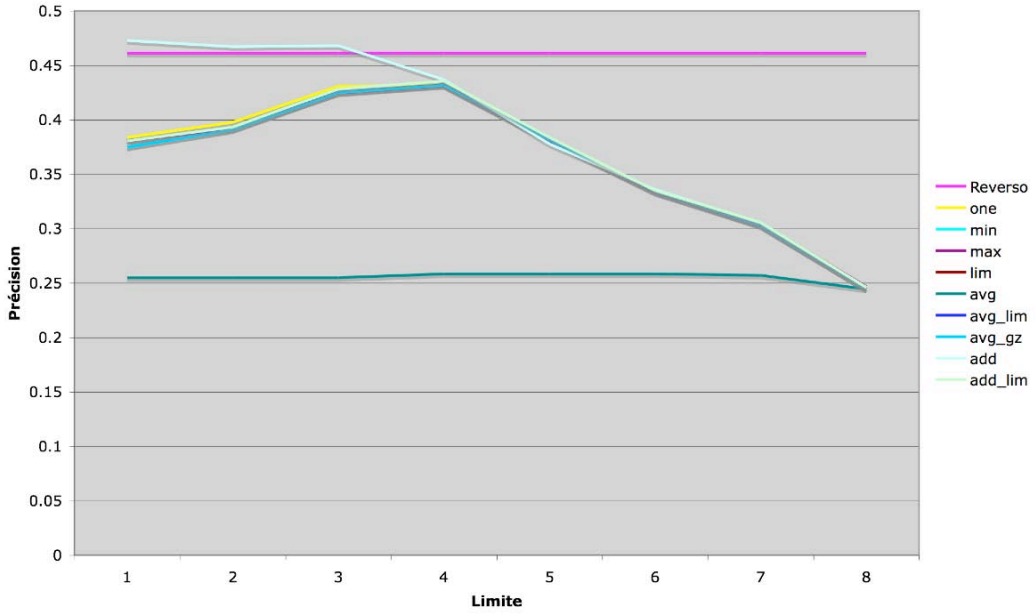

Figure 6 Précisions moyennes pour l'espagnol

On constate que la méthode *add* se distingue des autres. C'est la seule qui ne détériore pas la précision moyenne, en tout cas pour ses meilleurs résultats. C'est aussi la seule dont la meilleure performance est obtenue avec une valeur *limite* de un, donc en opérant une concaténation simple des différentes traductions obtenues. Mais même si cette méthode ne détériore pas la précision moyenne, l'amélioration reste en deçà de 5 %, pourcentage qui nous permettrait de considérer cette augmentation comme significative.

Tous les autres systèmes détériorent la précision moyenne de manière significative. On remarque tout de même que leurs meilleurs résultats se situent avec une limite de quatre, donc lorsque la moitié au moins des traductions contient le mot. Concernant la méthode *avg*, on constate que la valeur de *limite* n'a que peu d'influence sur la performance. En comparaison, les deux autres méthodes intégrant la moyenne se comportent de la même manière que les autres méthodes.

Le tableau 18 présente pour l'expérience en allemand les valeurs de précision moyenne obtenues pour différentes valeurs de *limite* ainsi que les pourcentages d'améliorations par rapport au meilleur système de traduction, REVERSO qui obtient une précision moyenne de 30,74 (voir le tableau 8). La valeur en **gras** est la meilleure combinaison pour une valeur de *limite* donnée.

limite	1	2	3	4	5	8
one	29,90 (- 2,7 %)	31,05 (+ 1,0 %)	30,84 (+ 0,3 %)	29,83 (- 3,0 %)	**26,36 (-14,2 %)**	**17,18 (-44,1 %)**
min	29,81 (- 3,0 %)	30,94 (+ 0,7 %)	30,75 (+ 0,1 %)	29,82 (- 7,9 %)	26,35 (-14,3 %)	**17,18 (-44,1 %)**
max	28,95 (- 5,8 %)	29,94 (- 2,6 %)	29,42 (- 4,3 %)	28,31 (- 3,2 %)	25,38 (-17,4 %)	17,16 (-44,2 %)
lim	28,95 (- 5,8 %)	30,39 (- 1,1 %)	30,68 (+ 0,2 %)	29,76 (- 2,1 %)	26,25 (-14,6 %)	**17,18 (-44,1 %)**
avg	17,90 (-41,8 %)	17,90 (- 41,8 %)	17,90 (-41,8 %)	17,90 (-41,8 %)	18,15 (-40,9 %)	17,13 (-44,3 %)
avg_lim	28,95 (- 5,8 %)	30,40 (- 1,1 %)	30,44 (- 1,0 %)	29,78 (- 3,1 %)	26,15 (-14,9 %)	17,13 (-44,3 %)
avg_gz	29,78 (- 3,1 %)	30,92 (+ 0,6 %)	30,72 (- 0,1 %)	**29,84 (- 2,9 %)**	26,34 (-14,3 %)	17,13 (-44,3 %)
add	**34,73 (+13,0 %)**	**34,31 (+11,6 %)**	**33,01 (+ 7,4 %)**	29,29 (- 4,7 %)	25,55 (-16,9 %)	17,12 (-44,3 %)
add_lim	28,95 (- 5,8 %)	30,20 (- 1,8 %)	30,15 (- 1,9 %)	29,25 (- 4,8 %)	25,67 (-16,5 %)	17,12 (-44,3 %)

Tableau 18 Précision moyenne selon les valeurs de limite et les méthodes de combinaison (allemand)

La figure 7 présente graphiquement pour chaque méthode l'évolution de la précision moyenne par rapport à la valeur de la *limite*.

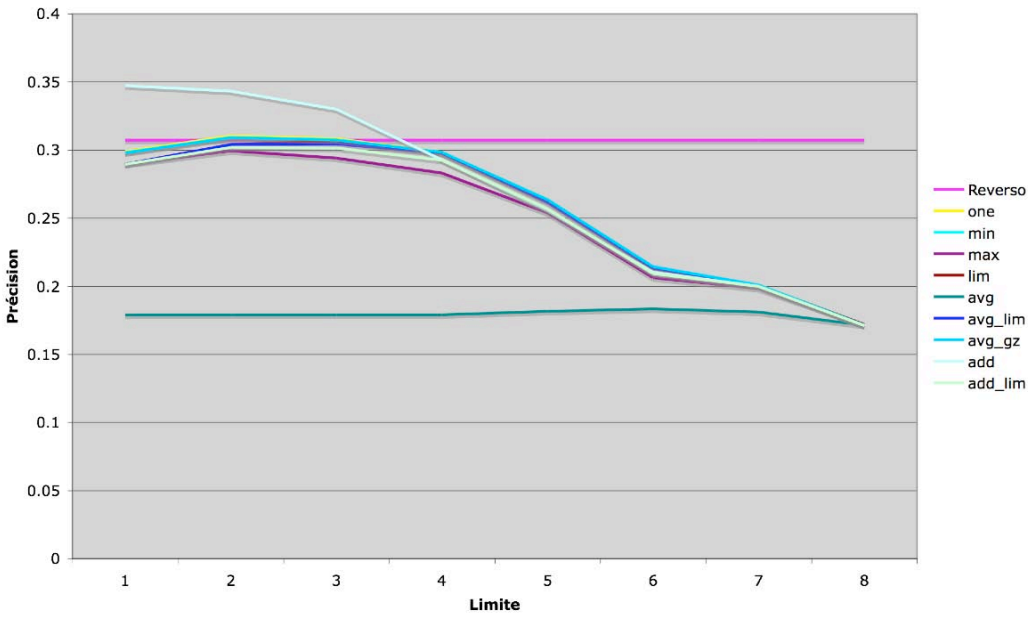

Figure 7 Précisions moyennes pour l'allemand

Contrairement au cas de l'espagnol, aucune des méthodes, à l'exception de *avg*, ne détériore significativement la précision moyenne, du moins pour de faibles valeurs de *limite* (moins de cinq). Pour le reste, on constate que c'est à nouveau la méthode *add* qui fournit les meilleurs résultats, mais cette fois avec une amélioration significative par rapport au meilleur outil de traduction pour les valeurs de *limite* inférieures à quatre.

En conclusion, nos expériences montrent que la meilleure méthode de combinaison parmi celles que nous avons testées pour ces langues reste la concaténation simple des différentes traductions. De plus, en comparant les figure 6 et figure 7, on remarque que ces graphes présentent des tendances similaires même si les valeurs sont plus basses pour les expériences sur l'allemand que celles sur l'espagnol. Cette dernière constatation était prévisible, les valeurs de précision moyenne étant plus faibles pour l'allemand que pour l'espagnol comme le montre le tableau 8.

3.5 Disparité entre les requêtes

Dans les sections 3.3 et 3.4, nous avons toujours travaillé sur l'ensemble des requêtes comme un tout. Chaque opération était effectuée sur chaque requête de la même manière. Or la mesure de performance indiquée dans le tableau 8 est une moyenne qui peut cacher une forte variabilité entre les divers outils de traduction. Afin de quantifier ces variations, nous avons indiqué dans le tableau 19 deux valeurs par langue et outil de traduction. La première indique le nombre de requêtes pour lesquelles le système correspondant propose la meilleure

alternative, la seconde le nombre de requêtes pour lesquelles le système de traduction propose l'unique meilleure traduction.

	Nombre de requêtes			
collection	Français	Italien	Allemand	Espagnol
Reverso	**17,1** / 14	–	**20,3** / **19**	**23,2** / **20**
Systran	8,3 / 0	13,1 / 0	12,7 / 8	8,8 / 1
Google	11,3 / 3	14,1 / 3	11,2 / 6	10,3 / 2
FreeTransla	11,8 / 10	**21,6** / 10	14,8 / 14	18,3 / 16
InterTran	13,8 / 13	20,1 / 13	7,3 / 7	13,3 / 12
Babylon 1	17,0 / **15**	19,8 / **15**	9,3 / 9	13,8 / 13
Babylon 2	9,8 / 4	3,8 / 4	12,8 / 5	6,1 / 6
Babylon 3	9,8 / 4	3,3 / 4	10,8 / 3	5,1 / 5

Tableau 19 Nombre de requêtes pour lesquelles le système de traduction propose la meilleure traduction.

Pour la première valeur, lorsque deux ou plusieurs outils possèdent la même performance optimale, la valeur accordée équivaut à l'inverse du nombre d'outils proposant cette meilleure précision. Ainsi, si deux systèmes de traduction obtiennent la même performance maximale, on attribue à chacun une valeur de 1/2, et si quatre systèmes arrivent au même résultat optimal, le score attribué à chacun sera de 1/4. Avec ce mode de calcul, le système REVERSO propose 20,3 fois la meilleure traduction en langue allemande, tandis que « Babylon 2 » apporte la meilleure traduction pour 12,8 requêtes.

Nous avons calculé la seconde valeur car face à une requête courte (en utilisant le titre uniquement), les systèmes de traduction automatique peuvent proposer des traductions identiques et obtiennent donc la même performance. Ainsi, pour la langue française, le système REVERSO propose la meilleure traduction pour 17,1 requêtes, dont 14 fois où il s'avère être le seul. Pour la langue espagnole, et uniquement pour cette langue, le système REVERSO offre la meilleure performance moyenne (voir tableau 8) et obtient la meilleure performance pour 23,2 requêtes, nombre le plus élevé pour cette langue.

3.5.1 Apprentissage automatique

Nous avons constaté dans la section 3.2.3 que la traduction automatique pourrait nous permettre d'atteindre, voire de dépasser, la performance de la traduction manuelle si on choisissait la bonne traduction pour chaque requête. Dans les sections suivantes, nous allons donc tenter de sélectionner la meilleure traduction pour chaque requête. Dans ce dessein, nous allons explorer trois méthodes d'apprentissage automatique, qui, à partir de données statistiques, cherchent à classifier la requête entre deux ensembles possible : « c'est une bonne traduction » ou « ce n'est pas une bonne traduction ». Nous décrirons nos évaluations à l'aide de la méthode des « plus proches voisins » en section 3.6, des « arbres de décision » en section 3.7 et de la « régression logistique » en section 3.8.

Les données statistiques que nous avons récoltées sur les requêtes traduites sont résumées dans le tableau 20. Elles sont de trois types et nous allons les décrire de manière plus complète ci-dessous.

Nom	Description
source	Le système de traduction automatique ayant traduit la requête.
concepts	Le nombre de termes indexés inclus dans la requête.
avg_idf	La moyenne des valeurs de l'*idf* des termes de la requête.
max_idf	La valeur maximale de l'*idf* de tous les termes de la requête.
max_idf2	La deuxième valeur maximale de l'*idf* de tous les termes de la requête.
min_idf	La valeur minimale de l'*idf* de tous les termes de la requête.
min_idf2	La deuxième valeur minimale de l'*idf* de tous les termes de la requête.
sigle	La présence d'un acronyme dans la requête.
personne	La présence ou l'absence d'un nom propre de personne dans la requête.
géo	La présence ou l'absence d'un nom propre géographique dans la requête.
autre_nom	La présence ou l'absence d'un nom propre n'appartenant pas aux deux catégories précédentes comme l'*Orient Express* ou le groupe *Nirvana* dans la requête.
date	La présence ou l'absence d'une date dans la requête.

Tableau 20 Récapitulatif des données statistiques récoltées sur les requêtes

La première donnée qui nous intéresse, nommée *source*, identifie le système de traduction. Nous aimerions déterminer si un système d'apprentissage automatique choisit systématiquement un traducteur s'il dispose de cette information.

Nous avons ensuite quelques données dérivées de notre système d'indexation. En premier lieu, la valeur *concepts* nous indique le nombre de termes inclus dans la requête et qui se retrouvent dans la collection de documents que nous voulons traiter. Cette valeur ne tient pas compte du nombre d'occurrences du terme dans la requête. On remarque que les mots outils ne sont pas compris dans le dictionnaire et n'apparaissent donc pas dans cette valeur.

Les autres données de ce type sont basées sur la fréquence documentaire, plus précisément sur l'*idf* défini comme :

$$idf = \ln\left(n/df\right) \tag{5}$$

avec *n* le nombre de documents dans la collection et *df* le nombre de documents comprenant le terme, les termes absents de la collection ainsi que les mots-outils ayant par définition une valeur nulle. Les données qui nous intéressent sont la moyenne des valeurs d'*idf* des termes de la requête notée *avg_idf*, la valeur maximale pour tous les termes de la requête (donc la valeur du terme le moins fréquent de la collection présent dans la requête) notée *max_idf* et la valeur minimale notée *min_idf*. De plus, afin d'obtenir une statistique plus robuste et atténuer un éventuel effet de pic dû à une valeur exceptionnelle, nous avons aussi calculé la valeur immédiatement inférieure à la valeur maximale, *max_idf2*, et la valeur immédiatement supérieure à la valeur minimale, *min_idf2*. On notera que la valeur moyenne de l'*idf* s'avère être un bon prédicteur de la performance d'une

requête selon Cronen-Townsend *et al*. (2002), hypothèse que nous désirons vérifier dans notre contexte bilingue.

Le dernier type de données est constitué de valeurs binaires (0 ou 1) indiquant la présence ou l'absence de propriétés qui nous semblent avoir une influence sur la traduction automatique et qui peuvent être identifiées facilement. En effet, certains traducteurs ont des difficultés particulières pour les traiter. Nous avons considéré les sigles, les noms propres ainsi que les dates. La présence d'acronymes ou de sigles, comme O.N.U., OCDE ou UE, valeur nommée *sigle*, est déterminée à l'aide de reconnaissance de motifs, puisqu'ils apparaissent dans nos requêtes comme suite de majuscules séparées ou non par des points. Ensuite, les noms propres sont séparés en trois catégories, les noms de personnes comme « Miguel Indurain », les noms géographiques comme « Madrid », « France » ou « Alps » et les noms propres qui ne font pas partie des précédentes comme le groupe de musique « Nirvana » ou le train « Orient Express ». Ces valeurs sont nommées respectivement *personne*, *géo* et *autre_nom* et sont déterminées à l'aide de listes. Notre dernière propriété qui indique la présence d'une date et est à nouveau déterminée à l'aide de reconnaissance de motifs.

Enfin, afin de pouvoir apprendre, il nous faut la classification entre « bonne traduction » et « mauvaise traduction » pour chaque requête traduite qui fera partie de l'ensemble d'apprentissage.

3.6 Plus proches voisins

La première technique d'apprentissage dont nous parlerons est celle des *k*-plus proches voisins (*k-nearest neighbours* en anglais). Pour cette méthode, nous considérons que chaque donnée statistique représente une dimension dans un espace en acceptant le postulat que si deux points de cet espace sont proches, ils appartiennent à la même classe. Notre représentation d'une traduction d'une requête est donc un point dans cet espace. Afin de classifier un nouveau point avec notre système, nous comparons la distance euclidienne entre ce point et chacun des points déjà existants (ensemble d'apprentissage). Si *k* vaut un, nous attribuons la classe du point minimisant cette distance à notre nouveau point. Si *k* est supérieur à un, nous attribuons la classe de la majorité des *k* points les plus proches à notre nouveau point.

Une de nos variables, la *source*, est une variable catégorielle et doit être transformée en plusieurs variables indicatives. La variable ayant huit valeurs possibles, nous aurons besoin de huit variables binaires pour la remplacer. Le tableau 21 présente les valeurs des huit variables de remplacement pour chaque valeur de la variable *source*.

Source	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Babylon 1	1	0	0	0	0	0	0	0
Babylon 2	0	1	0	0	0	0	0	0
Babylon 3	0	0	1	0	0	0	0	0
Google	0	0	0	1	0	0	0	0
Free	0	0	0	0	1	0	0	0
Reverso	0	0	0	0	0	1	0	0
Systran	0	0	0	0	0	0	1	0
InterTran	0	0	0	0	0	0	0	1

Tableau 21 Valeurs des variables de remplacement pour la variable source.

Avec nos 11 autres données statistiques, nous avons donc un espace à 19 dimensions comme espace de travail. Afin d'éviter qu'une donnée soit privilégiée par rapport à d'autres à cause de son échelle, toutes les données sont normalisées selon la formule (6).

$$x_n = \frac{x - \mu}{\sigma} \qquad (6)$$

où x_n est la valeur normalisée, x la valeur à normaliser, μ la moyenne des valeurs de x pour toutes les requêtes de toutes les traductions et σ son écart-type.

Nous avons donc commencé par classifier toutes les traductions de chaque requête en deux classes :

- la traduction est l'une des meilleures traductions (classe 1);
- la traduction n'est pas l'une des meilleures traductions (classe 0).

Pour déterminer si une traduction est la meilleure, nous avons calculé la précision moyenne de chaque requête pour chaque traduction. Nous considérons qu'une traduction est la meilleure si sa précision moyenne s'écarte au plus de 5 % de la précision moyenne maximale pour cette requête.

Pour nos expériences, nous avons utilisé la méthode *leaving-one-out*. Notre ensemble d'apprentissage se compose de toutes les requêtes sauf une et notre ensemble de test est le singleton formé de la requête écartée de l'ensemble d'apprentissage.

Afin de vérifier si certaines variables décrites à la section 3.5.1 ont plus d'importance que d'autres pour cette technique d'apprentissage automatique, nous avons évalué chacun des jeux de requêtes résultant des 4095^{13} combinaisons de variables possibles (chaque variable prise en compte ou non) pour la méthode du plus proche voisin et des trois plus proches voisins. Nous avons travaillé avec la collection espagnole décrite dans la section 3.2.1.

La méthode des plus proches voisins ne nous garantit pas que nous aurons pour chaque requête une traduction unique qui appartiendra à la classe 1. Nous avons

[13] La variable source n'est considérée que comme une seule variable. Il n'y a donc que 2^{12} possibilités. La 4096^{e} combinaison (aucune variable prise en compte) n'a aucun intérêt.

donc décidé que si, pour une requête, plusieurs traductions appartenaient à la classe 1, nous concaténerions les différentes versions. Dans le cas où aucune n'appartiendrait à la classe 1, nous les concaténerions toutes.

En analysant les résultats de nos évaluations, nous constatons que les précisions moyennes obtenues sont très différentes selon les variables retenues. Le tableau 22 présente les dix meilleures et les dix moins bonnes combinaisons de variables pour la méthode du plus proche voisin. La précision moyenne est indiquée avec entre parenthèses le pourcentage d'amélioration par rapport à la meilleure traduction automatique, celle de REVERSO, qui obtient une précision moyenne de 46,13. Le tableau 24 contient les mêmes informations pour la méthode des trois plus proches voisins.

En détaillant les résultats pour la méthode du plus proche voisin, nous constatons que la précision moyenne est supérieure ou égale à celle obtenue par la meilleure traduction automatique pour 839 combinaisons de variables d'apprentissage. Le meilleur résultat, avec 48,44, atteint juste une amélioration de 5 % ce qui nous permet de le considérer comme réellement meilleur. Ceci est confirmé par un test statistique, nous avons choisi le test du signe avec des valeurs considérées comme égales si la différence est de moins de 10 % en valeur relative (Conover 1980). Les valeurs de précision qui présentent une amélioration significative sont soulignées dans le tableau 22. Le moins bon résultat obtient une précision moyenne de 39,18 et présente donc une détérioration de 15,1 %.

Rang	Variables	Précision
1	*avg_idf, max_idf, min_idf2, personne, autre_nom*	48,44 (+ 5,0 %)
2	*avg_idf, max_idf, min_idf2, personne*	48,15 (+ 4,4 %)
3	*avg_idf, max_idf, min_idf2, autre_nom*	48,05 (+ 4,2 %)
4	*avg_idf, max_idf, min_idf2, sigle, personne, autre_nom*	48,01 (+ 4,1 %)
5	*avg_idf, max_idf, min_idf2, personne, autre_nom, date*	47,99 (+ 4,0 %)
6	*avg_idf, max_idf, min_idf, min_idf2, sigle, personne, géo*	47,95 (+ 3,9 %)
7	*avg_idf, max_idf, min_idf2, sigle, personne, autre_nom, date*	47,92 (+ 3,9 %)
8	*concepts, avg_idf, max_idf, min_idf2, personne, autre_nom*	47,89 (+ 3,8 %)
9	*avg_idf, max_idf, min_idf2, personne, date*	47,87 (+ 3,8 %)
10	*avg_idf, max_idf, min_idf2*	47,87 (+ 3,8 %)
...		
4086	*source, géo*	40,91 (-11,3 %)
4087	*source, sigle, géo, autre_nom, date*	40,70 (-11,8 %)
4088	*concepts, personne, autre_nom*	40,67 (-11,8 %)
4089	*source, personne, géo*	40,65 (-11,9 %)
4090	*source, géo, autre_nom*	40,60 (-12,0 %)
4091	*source, personne, géo, autre_nom, date*	40,53 (-12,1 %)
4092	*source, personne, géo, autre_nom*	40,29 (-12,7 %)
4093	*source, sigle, géo, date*	40,27 (-12,7 %)
4094	*source, sigle, géo, autre_nom*	40,07 (-13,1 %)
4095	*source, sigle, géo*	39,18 (-15,1 %)

Tableau 22 Variables utilisées et précision moyenne pour les meilleurs et moins bons résultats avec la méthode du plus proche voisin.

Les dix meilleures combinaisons comportent toutes les variables *avg_idf*, *max_idf* et *min_idf2*. Aucune ne contient les variables *source* et *max_idf2*. Si on continue le classement, pour les 839 combinaisons dont nous parlions plus haut, la variable *source* n'apparaît que 85 fois (10 %) alors que la suivante, *min_idf*, est présente dans 333 combinaisons (40 %). Le tableau 23 présente les fréquences d'apparition des différentes variables d'apprentissage pour ces 839 combinaisons.

Variable	*source*	*concepts*	*avg_idf*	*max_idf*	*max_idf2*	*min_idf*	*min_idf2*	*sigle*	*personne*	*géo*	*autre_nom*	*date*
Nb	85	368	473	499	350	333	486	446	426	439	422	413
%	10 %	44 %	56 %	59 %	42 %	40 %	58 %	53 %	51 %	52 %	50 %	49 %

Tableau 23 Fréquence d'apparition des variables d'apprentissage dans les 839 meilleures combinaisons.

Si l'on s'intéresse aux dix pires combinaisons, on constate la présence des variables *source* et *géo* dans neuf cas. Aucune ne contient l'une des variables basées sur l'*idf*. Nous en concluons donc que dans le cas du plus proche voisin, les variables basées sur l'*idf* sont plutôt un facteur d'amélioration pour l'apprentissage automatique alors que de connaître la *source* tend à nous donner des résultats médiocres.

Nous avons aussi constaté que les meilleures combinaisons proposent d'inclure en moyenne entre 3,4 et 3,8 traductions différentes pour chaque requête alors que les moins bonnes n'en sélectionnent qu'entre 1,7 et 2,3. Le choix des traductions est aussi plus diversifié dans le cas des meilleures combinaisons, certaines (GOOGLE, FREETRANSLATION, SYSTRAN et INTERTRAN) étant systématiquement boudées dans les moins bonnes. Ceci est probablement dû à la présence de la variable *source*.

Les mêmes analyses ont été menées pour la méthode des trois plus proches voisins, avec des constatations fort différentes. La précision moyenne n'est supérieure ou égale à celle obtenue par la meilleure traduction automatique que pour 101 combinaisons de variables d'apprentissage et la précision moyenne est en général plus basse qu'avec la méthode du plus proche voisin. Le meilleur résultat, avec 47,56, atteint une amélioration de 3,1 % que nous ne considérons pas comme significative. Ceci est confirmé par notre test statistique (test du signe avec des valeurs considérées comme égales si la différence est de moins de 10 % en valeur relative). Le moins bon résultat obtient une précision moyenne de 32,66 et présente donc une détérioration de 29,2 %.

Rang	Variables	Précision
1	*source, min_idf, sigle, autre_nom, date*	47,56 (+ 3,1 %)
2	*source, concepts, min_idf, sigle, personne, autre_nom, date*	47,43 (+ 2,8 %)
3	*source, min_idf, sigle, personne, autre_nom, date*	47,41 (+ 2,8 %)
4	*date*	47,05 (+ 2,0 %)
5	*personne*	47,04 (+ 2,0 %)
6	*source, concepts, min_idf, min_idf2, géo, date*	46,97 (+ 1,8 %)
7	*source, max_idf, max_idf2, min_idf, min_idf2, sigle, personne, géo, autre_nom, date*	46,97 (+ 1,8 %)
8	*sigle*	46,96 (+ 1,8 %)
9	*géo*	46,96 (+ 1,8 %)
10	*géo, date*	46,95 (+ 1,8 %)
	...	
4086	*max_idf2, min_idf, personne, géo, autre_nom, date*	38,91 (-15,7 %)
4087	*min_idf, sigle, personne, autre_nom*	38,72 (-16,1 %)
4088	*max_idf2, min_idf, sigle, personne, date*	38,72 (-16,1 %)
4089	*max_idf2, min_idf, min_idf2, sigle, personne, autre_nom, date*	38,69 (-16,1 %)
4090	*source, personne, géo, date*	38,57 (-16,4 %)
4091	*source, géo, date*	37,26 (-19,2 %)
4092	*source, personne, géo*	37,20 (-19,4 %)
4093	*source, géo*	35,07 (-24,0 %)
4094	*source, autre_nom, date*	34,27 (-25,7 %)
4095	*source, autre_nom*	32,66 (-29,2 %)

Tableau 24 Variables utilisées et précision moyenne pour les meilleurs et moins bons résultats avec la méthode des trois plus proches voisins.

Ici, aucune variable ne se démarque réellement des autres, *avg_idf* et *max_idf* étant absentes tant des meilleurs que des moins bons résultats alors que les autres sont présentes aux deux extrêmes. Nous avons tout de même constaté que la variable *source* était très présente dans les meilleurs résultats pour cette méthode.

Variable	*source*	*concepts*	*avg_idf*	*max_idf*	*max_idf2*	*min_idf*	*min_idf2*	*sigle*	*persomne*	*géo*	*autre_nom*	*date*
Nb	76	44	26	25	33	47	53	51	48	62	69	57
%	75 %	44 %	26 %	25 %	33 %	46 %	52 %	50 %	47 %	61 %	68 %	56 %

Tableau 25 Fréquence d'apparition des variables d'apprentissage dans les 101 meilleures combinaisons.

En analysant les traductions sélectionnées, nous obtenons un début de réponse au sujet des résultats obtenus. En effet, dans cinq des dix meilleures combinaisons, le système ne choisit des traductions que pour une ou deux requêtes, toutes les autres n'obtenant que des classes 0. Ceci explique une différence de précision moyenne de moins de 0,3 % par rapport à la concaténation simple (voir tableau 17). La nature de l'ensemble d'apprentissage qui comprend beaucoup plus de classe 0 que de classe 1 est probablement la source de cet effet. C'est d'ailleurs la raison qui nous a poussé à analyser en premier "le plus-proche-voisin". Si nous n'avons pas

des groupes de points très bien séparés, il y a donc de bonnes chances que, même si le point le plus proche est dans la bonne classe, les deux suivants ne le seront pas.

La méthode du plus proche voisin nous donne donc des résultats encourageants, mais si on augmente le nombre de voisins nécessaires à la décision, le système n'est plus à même de décider correctement.

3.7 Génération d'arbres de décision

Un *arbre de décision* est une méthode de classification des données. Un *arbre de décision* est soit une *feuille*, soit un *nœud de décision*. Un *nœud de décision* décrit un test à effectuer pour un attribut de classification et comprend une branche (ouvrant sur un sous-arbre) pour chaque résultat possible du test. Une *feuille* désigne la classe à laquelle appartiennent les données correspondant à la décision explicitée par le chemin de la racine au nœud en question.

La figure 8 présente en exemple une partie d'arbre de décision construit automatiquement sur la base de nos données. Les feuilles contiennent la décision de classification, dans le cas présent, la traduction est « bonne » ou « mauvaise ».

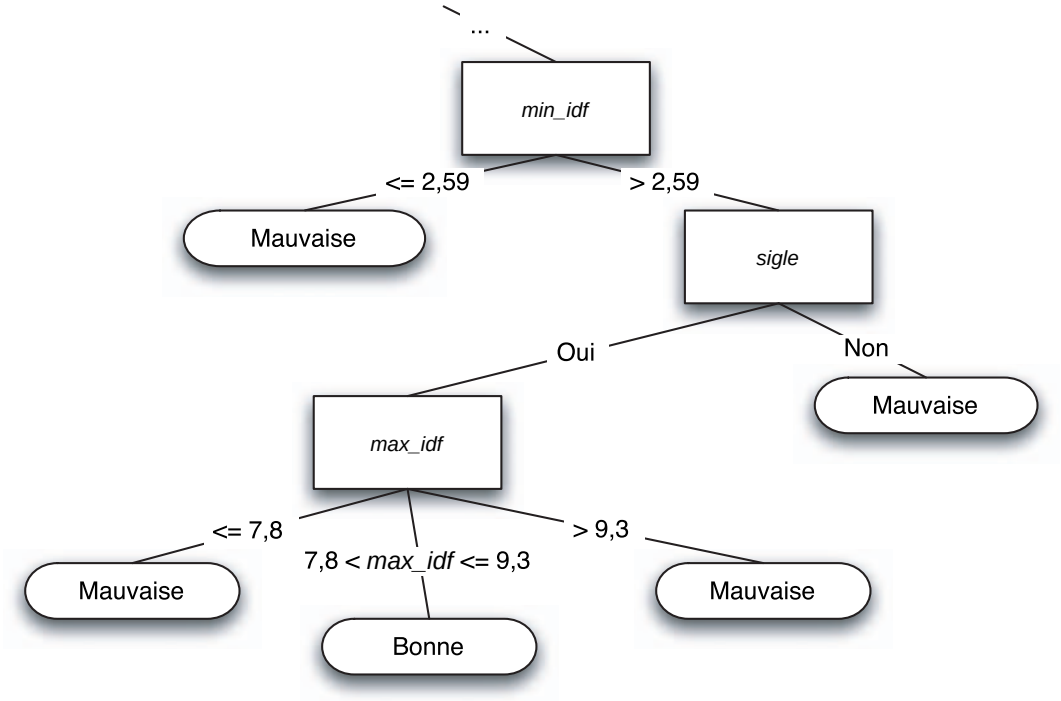

Figure 8 Exemple d'arbre de décision

Ce même arbre de décision peut être représenté sous la forme des règles suivantes :

- si *min_idf* <= 2,59,
  - alors la traduction est mauvaise;
- si *min_idf* > 2,59,
  - si la requête contient un *sigle*,
    - si *max_idf* <= 7,8,
      - alors la traduction est mauvaise;
    - si 7,8 < *max_idf* <= 9,3,
      - alors la traduction est bonne;
    - si *max_idf* > 9,3,
      - alors la traduction est mauvaise;
  - si la requête ne contient pas de *sigle*,
    - alors la traduction est mauvaise.

On sait qu'un *idf* faible (un mot apparaît dans un ou quelques documents) peut être le signe d'une faute d'orthographe ou, dans le cas présent, d'une traduction peu usuelle (voire également d'une erreur dans le dictionnaire bilingue). De même, sur une requête, on devrait, normalement, rencontrer un terme ayant un *idf* élevé (soit un mot courant dans la langue). Si ce n'est pas le cas, les règles indiquent qu'il s'agit, peut-être, d'une mauvaise traduction.

Nous avons utilisé le programme C4.5 (Quinlan 1993) pour créer un arbre de décision à partir de nos données d'apprentissage et de la classification correspondante. Ce programme part de l'ensemble d'entraînement complet et va chercher à le partitionner à partir d'un attribut, puis recommence pour chacun des ensembles résultants. Lorsqu'il traite un ensemble, trois possibilités existent :

- tous les éléments appartiennent à la même classe, il n'y a donc rien à partitionner et le *nœud* est une *feuille* désignant la classe des éléments;
- les éléments appartiennent à des classes différentes et peuvent être discriminés par un attribut, il partitionne donc l'ensemble à l'aide de cet attribut et relance le processus pour chaque sous-ensemble créé;
- les éléments appartiennent à des classes différentes et ne peuvent pas être discriminés par un attribut, il n'y a donc pas de partition possible. Dans ce cas, le *nœud* est une *feuille* et désigne la classe la plus fréquente dans l'ensemble.

Afin de choisir quel attribut il faut utiliser à chaque étape, C4.5 fonctionne selon une méthode basée sur le gain d'information. Pour cela, il calcule la quantité d'information contenue dans un ensemble *S,* sur la base de l'entropie de *S* définie comme suit :

$$Entropie(S) = \sum_{i \in C} -\frac{|S_i|}{|S|} \log_2 \left(\frac{|S_i|}{|S|} \right) \qquad (7)$$

avec C l'ensemble des classes possibles et S_i l'ensemble des éléments de S appartenant à la classe i. À partir de l'entropie, le gain d'information pour un attribut est calculé ainsi :

$$Gain(S,\text{attribut}) = Entropie(S) - \sum_{a \in A} \frac{|S_a|}{|S|} Entropie(S_a) \quad (8)$$

avec A l'ensemble des valeurs possibles de cet attribut et S_a l'ensemble des éléments pour lesquels l'attribut vaut a. Le système va alors choisir l'attribut pour lequel le gain d'information est maximal comme nœud de décision. Cette méthode était utilisée par le prédécesseur de C4.5 nommé ID3. Mais un biais important se dessine, le système choisissant plus facilement des attributs ayant beaucoup de valeurs possibles au risque d'avoir des règles tellement spécifiques que les candidats à la classification n'entrent pas dans l'arbre de décision. Pour rectifier ce biais, C4.5 va calculer la quantité d'information engendrée en divisant l'ensemble selon les valeurs de l'attribut :

$$SplitInfo(S,\text{attribut}) = -\sum_{a \in A} \frac{|S_a|}{|S|} \log_2\left(\frac{|S_a|}{|S|}\right) \quad (9)$$

Cette valeur est alors utilisée pour mesurer le rapport de gain d'information :

$$RapportGain(S,\text{attribut}) = \frac{Gain(S,\text{attribut})}{SplitInfo(S,\text{attribut})} \quad (10)$$

qui nous donne la proportion de l'information engendrée par la partition qui paraît utile à la classification. Ce rapport ayant tendance à augmenter si la division est presque triviale, C4.5 se base sur deux conditions pour choisir l'attribut :

- le gain doit être plus grand que la moyenne des gains calculés pour les différents attributs possibles;
- parmi ces attributs, choisir celui qui obtient le plus grand rapport de gain.

Après cette première étape et dans la situation idéale, nous avons un arbre de décision permettant de classifier correctement tous les éléments de l'ensemble d'apprentissage. Ceci n'est probablement pas désirable car on risque d'obtenir un arbre trop ajusté à l'ensemble d'apprentissage qui ne sera pas performant lors de l'utilisation avec de nouveaux éléments à classifier.

Le programme va donc procéder à une deuxième étape nommée élagage. Pour chaque nœud de décision, il essaie de le remplacer par une feuille désignant la classe la plus fréquente pour l'ensemble contenu à ce niveau et il calcule la proportion d'erreurs que commet l'arbre de décision sur l'ensemble d'entraînement après cette modification. Si cette proportion est inférieure à une valeur donnée (d'habitude 25 %) et si l'arbre contient toujours au moins une feuille désignant chaque classe présente dans l'arbre avant l'élagage, alors cette modification est conservée. Dans le cas contraire, le nœud est réintroduit. Dans tous les cas,

l'élagage se poursuit tant qu'il reste un nœud à tester. C4.5 essaye aussi de remplacer le nœud candidat par le sous-arbre correspondant au choix le plus fréquent pour ce nœud. Enfin, l'arbre est transformé en un ensemble de règles et nous pouvons utiliser le système pour classifier de nouveaux éléments.

Cette méthode d'apprentissage a été testée avec deux collections, en espagnol et en allemand, à partir des requêtes en anglais. Nos traductions, requête par requête, sont distribuées en deux classes :

- la traduction est l'une des meilleures traductions (classe 1);
- la traduction n'est pas l'une des meilleures traductions (classe 0).

Pour l'apprentissage, nous avons considéré trois possibilités pour le choix de la classe :

- la méthode stricte, où seuls les traducteurs ayant la meilleure performance appartiennent à la classe 1;
- la méthode normale, où les traducteurs appartiennent à la classe 1 si leur performance s'écarte au plus de 5 % de la performance du meilleur traducteur;
- la méthode assouplie, où les traducteurs appartiennent à la classe 1 si leur performance s'écarte au plus de 10 % de la performance du meilleur traducteur.

Comme dans le cas des plus proches voisins (section 3.6), nous n'avons pas la garantie d'obtenir une unique traduction dans la classe 1 pour chaque requête. Nous avons à nouveau décidé que si, pour une requête, plusieurs traductions appartiennent à la classe 1, nous concaténerions les différentes versions. Dans le cas où aucune n'appartiendrait à la classe 1, nous les concaténerions toutes.

Le tableau 26 présente quelques informations concernant les règles engendrées par C4.5 selon la langue et la méthode de choix de la classe. La colonne *Nb* indique le nombre de règles finales (après élagage). Les tailles représentent la longueur du chemin de la racine au nœud feuille ce qui correspond au nombre de conditions que contiennent les règles. Nous présentons cette valeur au minimum, au maximum et en moyenne. Le taux d'erreur est celui qui est calculé par le programme en fonction du nombre d'éléments de l'ensemble d'apprentissage qui ne seront pas correctement classifiés suite à la procédure d'élagage. Enfin le type d'erreur sépare les éléments selon la classification par le système et l'indication de la correction de cette classification. Ainsi, *vp* indique une classification 1 correcte, *vn* une classification 0 correcte, *fp* une classification 1 incorrecte et *fn* une classification 0 incorrecte.

		Taille			Taux d'erreur	Type d'erreur			
	Nb	min	moy	max		vp	vn	fp	fn
ES strict	12	1	3	5	12,8 %	43	655	6	96
ES normal	21	1	3,6	6	25 %	118	482	54	146
ES assoupli	34	1	4,3	7	22 %	210	414	50	126
DE strict	6	1	2,7	7	14,5 %	20	664	9	107
DE normal	6	1	4,3	8	18 %	42	614	0	144
DE assoupli	19	1	3,7	7	18,9 %	94	555	13	138

Tableau 26 Statistiques à propos des règles induite par C4.5

Nous constatons qu'un grand nombre d'éléments appartenant à la classe 1 sont mal classifiés par le système. En espagnol, malgré la limite de taux d'erreur de 25 %, plus d'un tiers de ces éléments n'obtiennent pas la classe correcte, alors que seuls un huitième des éléments de classe 0 sont mal classifiés. En allemand, cette différence est encore plus importante malgré un taux d'erreur calculé plus faible. De plus, un nombre important de requêtes ne comprendront aucune traduction classifiée comme bonne traduction. Pour ces requêtes, c'est donc la concaténation de toutes les traductions qui sera utilisée. Dans le cas de l'espagnol (méthode stricte) et de l'allemand (méthode stricte ou normale), plus de la moitié des requêtes se trouvent dans cette situation.

Le tableau 27 présente les valeurs de précision moyenne obtenues avec les requêtes construites à l'aide de l'apprentissage ainsi que l'amélioration par rapport à la précision du meilleur traducteur automatique. On constate qu'en espagnol, la précision est très légèrement améliorée. Même si elle ne semble guère importante, un test statistique (test du signe) permet de vérifier que cette amélioration est tout de même significative (valeurs soulignées dans le tableau). D'autre part, le choix de la méthode de définitions des classes pour l'apprentissage ne change que très peu les résultats. En allemand, les différences sont aussi statistiquement significatives. De plus, la méthode assouplie donne de bons résultats avec une augmentation de la précision de 12 %. On remarque tout de même que la marge de progression potentielle en allemand est plus importante (voir tableau 8).

	précision
ES strict	47,76 (+ 3,5 %)
ES normal	47,97 (+ 4 %)
ES assoupli	47,76 (+ 3,5 %)
DE strict	31,47 (+ 2,4 %)
DE normal	31,28 (+ 1,8 %)
DE assoupli	34,42 (+ 12 %)

Tableau 27 Précision moyenne pour l'apprentissage C4.5

Puisque nous avons utilisé le même ensemble comme ensemble d'apprentissage et comme ensemble de test, nous voulions vérifier que les arbres de décision générés ne sont pas biaisés. Nous avons donc construit les cent arbres correspondant à la méthode du *leaving one out* pour l'expérience espagnole stricte. Pour cette technique, nous écartons une requête de l'ensemble d'apprentissage et nous

construisons l'arbre résultant, puis nous utilisons la requête écartée comme test. Sur les cent arbres générés, seuls deux étaient différents de l'arbre général après la procédure d'élagage et aucune différence dans la classification des requêtes écartées n'est constatée. L'élagage permet donc d'obtenir un arbre suffisamment général pour réduire, voire éviter, le biais.

Nous constatons donc que dans toutes nos expériences, les résultats de la recherche sont améliorés après l'usage de l'apprentissage automatique à l'aide d'arbres de décisions pour la sélection de traduction.

3.8 Régression logistique

Nos différentes informations statistiques, décrites en section 3.5.1, nous permettent de considérer la classification automatique entre « bonne » et « mauvaise » traduction avec une autre méthodologie. Dans le cas de la régression logistique, la variable expliquée (« bonne » ou « mauvaise » traduction) dépend des données statistiques comme variables explicatives. La variable expliquée n'ayant que deux valeurs possibles, la régression logistique va estimer la probabilité que prenne la valeur un. Comme exemple, nous traçons dans la figure 9 le graphe de dispersion de la variable expliquée par rapport à une variable explicative. Nous obtenons deux lignes de points plus ou moins concentrés, une au niveau zéro et l'autre au niveau un (losanges bleus). Utiliser une droite de régression linéaire (ligne pointillée) n’est pas forcément une solution pour estimer la probabilité d’une variable binaire. En effet, nous aimerions obtenir une valeur bornée entre zéro et un. Or, il est possible que pour certaines valeurs de la variable explicative, la valeur calculée par la régression linéaire soit supérieure à un ou inférieure à zéro, ce qui en rend l'interprétation absurde.

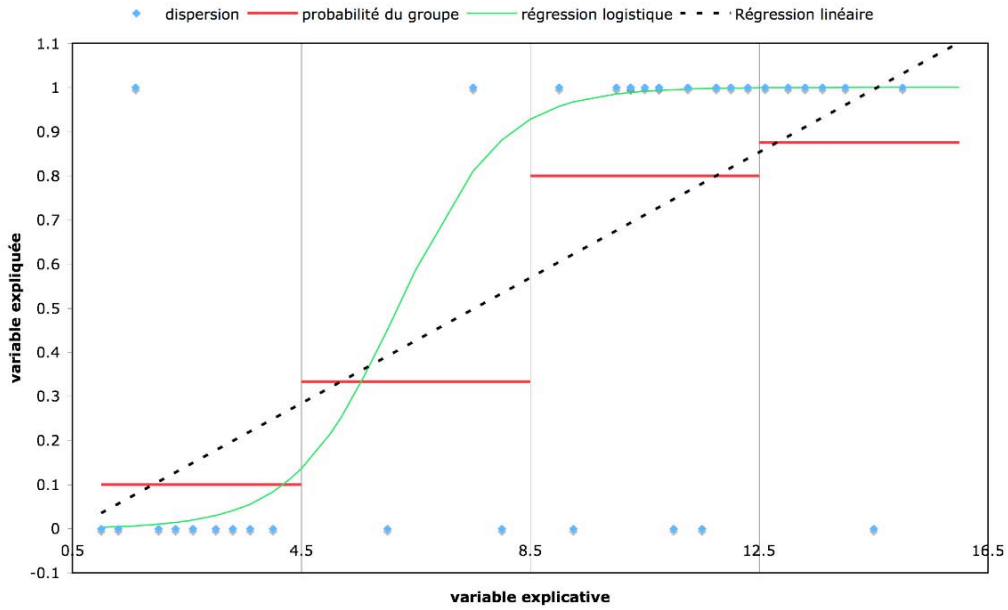

Figure 9 Exemple de graphe de dispersion pour une variable expliquée par rapport à une variable explicative, probabilité par groupe, régression linéaire et régression logistique.

Si nous créons des groupes en fonction de la valeur de la variable explicative, nous pouvons calculer la proportion de chaque groupe ou probabilité que ce groupe appartienne à la classe « un » de la variable expliquée (lignes rouges). Cette valeur représente aussi la probabilité de réalisation de la variable expliquée si la variable explicative appartient au groupe. La courbe que forment ces valeurs, appelée *courbe en S*, se rapproche de la distribution cumulative d'une variable aléatoire. Hosmer *et al*. (2000) proposent d'utiliser le modèle de distribution logistique pour approximer la probabilité de réalisation lorsque la variable expliquée est binaire. La fonction de régression logistique prend la forme :

$$\pi(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}} \tag{11}$$

avec α et β les coefficients à estimer. Dans le cas multivarié, nous regroupons nos différentes variables dans le vecteur $\mathbf{X} = [x_1, x_2, \ldots x_k]$. La fonction de régression logistique s'écrit alors selon la formule 12.

$$\pi(X) = \frac{e^{\alpha+\sum_{i=1}^{k}\beta_i x_i}}{1+e^{\alpha+\sum_{i=1}^{k}\beta_i x_i}} \tag{12}$$

Les coefficients α et $\beta_1, \beta_2 \ldots \beta_k$ de la formule 12 sont estimés à l'aide du logiciel R (Venaples *et al.* 1999) selon la méthode du maximum de vraisemblance, à partir

de notre échantillon d'apprentissage, c'est-à-dire nos différentes informations statistiques et la bonne décision.

La méthode de régression logistique a déjà été utilisée dans différents contextes de prédiction en bibliothéconomie (Bookstein *et al*. 1992), comme stratégie de dépistage (Gey 1994) ou pour la fusion de listes de résultats dans le cadre de métamoteurs de recherche (Le Calvé *et al.* 2000).

Afin d'évaluer cette méthode, nous avons travaillé avec la collection en espagnol à partir des requêtes en anglais. Nos traductions, requête par requête, sont distribuées en deux classes :

- la traduction est l'une des meilleures traductions (classe 1);
- la traduction n'est pas l'une des meilleures traductions (classe 0).

Pour déterminer si une traduction est la meilleure, nous avons calculé la précision moyenne de chaque requête pour chaque traduction. Nous considérons qu'une traduction est la meilleure si sa précision moyenne s'écarte au plus de 5 % de la précision moyenne maximale pour cette requête.

Comme pour les autres expériences, nous allons comparer les résultats de notre apprentissage avec la meilleure traduction automatique, soit dans notre cas celle fournie par le système REVERSO (précision moyenne de 46,13 %). Afin de dépasser la performance de cette stratégie de référence, nous avons imaginé cinq modèles basés sur la régression logistique. Nous avons choisi un ensemble de variables pour chacun de ces modèles, puis nous avons utilisé la procédure `stepAIC` de R (minimisation du critère AIC (Venaples *et al.* 1999)) pour sélectionner les variables à utiliser dans la régression. Le tableau 28 présente les choix de variables de départ pour nos différents modèles et résume les variables sélectionnées par la procédure `stepAIC`.

Modèle	Variables choisies	Variables sélectionnées
A	Toutes	*source*, *min_idf, personne, géo* et *autre_nom*
B	Toutes sauf la source	*concepts*, *min_idf, personne, géo* et *autre_nom*
C	Données du système d'indexation uniquement	*concepts*, *max_idf, avg_idf, min_idf2* et *min_idf*
D	Même que C sans *max_idf2* et *min_idf2*	*concepts*, *max_idf, avg_idf* et *min_idf*
E	*avg_idf* seul.	Pas de sélection effectuée

Tableau 28 Variables retenues pour nos différents modèles de régression logistique

Dans le modèle A, nous avons tenu compte de toute l'information disponible et les variables retenues sont : *source*, *min_idf*, *personne*, *géo* et *autre_nom*. Selon nos attentes, l'origine de la traduction est incluse dans cet ensemble (variable *source*), et le fait de connaître que telle ou telle traduction a été produite par tel système (par exemple Reverso) s'avèrera un élément important dans la décision de retenir cette traduction comme la meilleure. Une mesure liée à l'*idf* a également été retenue (le *min_idf* dans ce cas) ainsi que les informations binaires

notant la présence ou non de noms propres de personne (*personne*), de géographie (*géo*) ou autres (*autre_nom*).

Pour le modèle B, nous avons tenu à masquer l'origine de la traduction afin de permettre une sélection uniquement sur les autres variables. Dans ce cas de figure, nous proposons de retenir les variables *concepts*, *min_idf*, *personne*, *géo* et *autre_nom*. On notera que dans ce deuxième modèle, la variable *source* est remplacée par la variable *concepts* (soit la longueur en nombre de mots indexés de la requête traduite).

Dans nos trois derniers modèle, nous désirons mettre à l'épreuve le modèle de prédiction de Cronen-Townsend *et al*. (2002) qui indique que l'*idf* moyen d'une requête s'avère être un bon prédicteur de la précision moyenne. Pour le modèle C, toutes les données du système d'indexation étaient choisies et les variables que nous avons sélectionnées sont : *concepts*, *min_idf*, *min_idf2*, *av_gidf*, et *max_idf*. Pour le modèle D, nous avons abandonné les deuxièmes minimum et maximum de l'*idf*. Les variables sélectionnées sont alors *concepts*, *min_idf*, *avg_idf*, et *max_idf*. Enfin, le modèle E ne contient que la variable *avg_idf* et la régression logistique a été appliquée sans sélection.

Afin d'évaluer les cinq modèles retenus, nous pouvons utiliser les 99 requêtes disponibles pour estimer les coefficients α, β_1, β_2, … β_k de la régression logistique et reprendre le même jeu de requêtes pour l'évaluation. Cette méthode d'évaluation, nommée rétrospective, est certes biaisée, mais elle nous permet d'avoir une idée de la performance sous-jacente du modèle. Comme méthode alternative d'évaluation, nous avons opté pour le *leaving-one-out* proposant une estimation sans biais. Cette stratégie autorise un entraînement sur l'ensemble des observations moins une et cette dernière est utilisée pour évaluer le modèle. En itérant sur les requêtes disponibles, nous pouvons ainsi obtenir une évaluation faite sur les 99 requêtes avec un apprentissage fait sur 98 requêtes.

Dans une première série d'expériences, nous avons exigé de la machine qu'elle retourne pour chaque requête la meilleure des traductions parmi les huit disponibles. En cas d'égalité, les différentes meilleures traductions sont concaténées. La performance moyenne obtenue sous ces conditions est indiquée dans le tableau 29 à la ligne notée « tolérance 0 % ». Pour le modèle A, la décision est relativement simple ; il faut toujours prendre la traduction obtenue par le modèle Reverso (évaluation rétrospective ou *leaving-one-out*). Disposant de moins d'information, les autres modèles présentent une précision moyenne nettement inférieure au modèle A ou à une stratégie simple retournant toujours la solution proposée par Reverso.

	Précision moyenne (% de changement)				
Modèle	A	B	C	D	E
Reverso	46,13	46,13	46,13	46,13	46,13
Rétrospective					
tolérance de 0 %	46,13 (0,0%)	42,91 (-7,0%)	41,04 (-11%)	41,29 (-11%)	42,97 (-6,9%)
tolérance de 5 %	46,13 (0,0%)	46,29 (+0,3%)	43,01 (-6,8%)	45,35 (-1,7%)	**47,35 (+2,6%)**
tolérance de 10 %	46,13 (0,0%)	47,99 (+4,0%)	46,59 (+1,0%)	47,15 (+2,2%)	47,31 (+2,6%)
tolérance de 15 %	46,20 (+0,2%)	**48,84 (+5,9%)**	47,30 (+2,5%)	47,53 (+3,0%)	47,31 (+2,6%)
tolérance de 20 %	47,27 (+2,5%)	48,52 (+5,2%)	**48,30 (+4,7%)**	48,12 (+4,3%)	47,31 (+2,6%)
tolérance de 25 %	**47,65 (+3,3%)**	48,64 (+5,4%)	48,28 (+4,7%)	**48,21 (+4,5%)**	47,31 (+2,6%)
Leaving-one-out					
tolérance de 0 %	46,13 (0,0%)	40,85 (-11%)	39,41 (-15%)	40,16 (-13%)	N/A
tolérance de 5 %	46,13 (0,0%)	44,06 (-4,5%)	42,37 (-8,2%)	43,39 (-5,9%)	N/A
tolérance de 10 %	46,13 (0,0%)	47,80 (+3,6%)	45,49 (-1,4%)	46,58 (+1,0%)	N/A
tolérance de 15 %	46,09 (-0,1%)	**48,75 (+5,7%)**	46,89 (+1,6%)	47,13 (+2,2%)	N/A
tolérance de 20 %	45,31 (-1,8%)	48,51 (+5,2%)	47,82 (+3,7%)	47,90 (+3,8%)	N/A
tolérance de 25 %	**46,22 (+0,2%)**	48,58 (+5,3%)	**47,91 (+3,9%)**	**48,24 (+4,6%)**	N/A

Tableau 29 Précision moyenne de nos cinq modèles de prédiction (méthode normale) espagnol

Au lieu d'être strict et de retenir uniquement la traduction maximisant l'équation 12 (régression logistique), nous pouvons admettre que si la probabilité estimée s'écarte de, par exemple, 5 % de la meilleure traduction, nous pouvons également considérer la traduction sous-jacente comme une bonne traduction de la requête courante. En admettant cette tolérance, nous ne limitons pas notre procédure de sélection à la recherche de l'unique traducteur optimum, mais notre système vise à trouver les bonnes traductions d'une requête écrite en anglais.

Sur cette base d'une marge de tolérance de 5 % (respectivement 10 %, 15 %, 20 % ou 25 %), la machine peut indiquer que deux ou plusieurs traductions automatiques peuvent être considérées comme excellentes et donc être retenues dans le dépistage final. Dans ce cas, la machine concatène les termes provenant de toutes les requêtes sélectionnées.

Les évaluations rétrospectives ou *leaving-one-out* reprises dans le tableau 29 indiquent que c'est seulement en combinant plusieurs traductions automatiques que nous arrivons à dépasser la performance moyenne du meilleur outil de traduction automatique, soit la valeur de 46,13 % obtenue par le système Reverso. De plus, comme nous l'attendions, l'évaluation rétrospective est légèrement meilleure que *leaving-one-out*, mais toutes deux ont un comportement similaire.

La meilleure performance en évaluation *leaving-one-out* est obtenue avec le modèle B et avec une marge de tolérance de 15 %. La précision moyenne s'élève à 48,75 % en concaténant, pour ce cas, en moyenne 5,16 traductions par requête. Avec le modèle A et une tolérance de 25 %, on atteint une précision moyenne de 46,22 % avec 3,19 traductions en moyenne. Pour les modèles C et D, la meilleure performance s'obtient aussi avec une tolérance de 25 %. Nous arrivons ainsi à une précision moyenne de 47,91 % avec 5,47 traductions en moyenne pour le modèle C et une précision moyenne de 48,24 % avec, en moyenne, 5,88 traductions par requête pour le modèle D.

D'une manière générale, il faut être tolérant. L'utilisation de l'unique meilleure traduction proposée par le système (tolérance 0 %) n'apporte jamais la meilleure performance. Il est aussi un peu surprenant de constater que notre modèle A n'arrive pas au même niveau de performance bien que disposant, a priori, d'une quantité d'information plus importante. Plus de précision ne s'avère donc pas toujours la meilleure approche. Mais dans ce modèle, le choix est fortement influencé par le système Reverso, bien qu'il soit en moyenne le meilleur, il ne fournit la meilleure traduction « que » pour 23,2 requêtes sur 99.

Concernant le modèle E, nous avons constaté à l'aide de l'évaluation rétrospective qu'il ne permettait de choisir des traductions que pour un niveau de tolérance de 0 % ou 5 %. Si on augmente ce niveau au-delà de 5 %, le système propose la concaténation de toutes les traductions pour chaque requête. Il n'arrive donc pas à choisir correctement de bonnes traductions avec si peu de données. L'évaluation *leaving-one-out* étant très coûteuse en temps machine, nous n'avons pas jugé nécessaire de l'effectuer pour ce modèle.

Dans nos différentes expériences, nous n'avons pas présenté de résultat avec une valeur de tolérance excédant 25 % car nous avons constaté qu'au-delà de cette valeur, nous obtenions très vite la concaténation de toutes les traductions.

Les différences de performances constatées par rapport à Reverso ne semblant pas très importantes, nous avons utilisé un test statistique (test du signe) pour vérifier si elles sont significatives. Dans le tableau 29, les valeurs sont soulignées lorsque la différence entre la précision moyenne de Reverso et celle de notre modèle est statistiquement significative d'après notre test. Nous constatons que pour les modèles B et D, les meilleurs résultats présentent une différence de précision moyenne statistiquement significative.

Comme dans le cas de la génération d'arbres de décision présentée en section 3.7, nous avons considéré deux autres possibilités pour le choix de la classe pour l'apprentissage :

- la méthode stricte, où seuls les traducteurs ayant la meilleure performance appartiennent à la classe 1;
- la méthode assouplie, où les traducteurs appartiennent à la classe 1 si leur performance s'écarte au plus de 10 % de la performance du meilleur traducteur.

Pour ces évaluations, nous avons abandonné le modèle E qui ne nous semblait pas assez intéressant. Le tableau 30 présente les variables sélectionnées par `stepAIC` pour nos différents modèles et selon nos deux méthodes de choix de classe pour l'apprentissage.

Modèle	Variables choisies	Variables sélectionnées	
		Méthode stricte	Méthode assouplie
A	Toutes	*concepts, max_idf2, avg_idf* et *autre_nom*	*source*, *min_idf2, personne, géo* et *autre_nom*
B	Toutes sauf la source	*concepts, max_idf2, avg_idf* et *autre_nom*	*concepts*, *min_idf2, personne, géo* et *autre_nom*
C	Données du système d'indexation uniquement	*concepts, max_idf* et *avg_idf*	*concepts*, *max_idf, avg_idf* et *min_idf2*
D	Même que C sans *max_idf2* et *min_idf2*	*concepts, max_idf* et *avg_idf*	*concepts* et *min_idf*

Tableau 30 Variables retenues pour nos différents modèles de régression logistique pour les méthodes stricte et assouplie, espagnol

On remarque que dans le cas de la méthode stricte, les variables sélectionnées pour les modèles A et B ainsi que C et D sont identiques. De plus, les variables sélectionnées selon nos deux méthodes sont proches de celles de la méthode normale. L'évaluation des différents modèles proposés est faite à l'aide de la méthode rétrospective car notre but est de vérifier le comportement de l'apprentissage selon la procédure de création de l'ensemble d'apprentissage.

Le tableau 31 présente la précision moyenne obtenue pour chacune de nos expérimentations ainsi que le pourcentage d'amélioration par rapport à notre meilleur traducteur, Reverso. Les valeurs sont soulignées lorsque la différence entre la précision moyenne de Reverso et celle de notre modèle est statistiquement significative d'après notre test (test du signe).

	Précision moyenne (% de changement)			
Modèle	A	B	C	D
Reverso	46,13	46,13	46,13	46,13
Méthode stricte				
tolérance de 0 %	42,60 (-7,7 %)		42,62 (-7,6 %)	
tolérance de 5 %	43,96 (-4,7 %)		44,07 (-4,5 %)	
tolérance de 10 %	44,39 (-3,8 %)		44,37 (-3,8 %)	
tolérance de 15 %	45,79 (-0,7 %)		46,10 (-0,1 %)	
tolérance de 20 %	**46,65 (+1,1 %)**		46,60 (+1,0 %)	
tolérance de 25 %	46,65 (+1,1 %)		**46,76 (+1,4 %)**	
Méthode assouplie				
tolérance de 0 %	46,13 (0,0 %)	41,60 (-9,8 %)	40,12 (-13 %)	42,39 (-8,1 %)
tolérance de 5 %	46,13 (0,0 %)	46,14 (+0,0 %)	43,60 (-5,5 %)	47,73 (+3,5 %)
tolérance de 10 %	46,24 (+0,2 %)	47,86 (+3,8 %)	44,84 (-2,8 %)	48,56 (+5,3 %)
tolérance de 15 %	46,53 (+0,9 %)	**48,35 (+4,8 %)**	46,47 (+0,7 %)	**48,68 (+5,5 %)**
tolérance de 20 %	**47,71 (+3,4 %)**	48,26 (+4,6 %)	47,05 (+2,0 %)	48,55 (+5,3 %)
tolérance de 25 %	47,69 (+3,4 %)	48,30 (+4,7 %)	**47,72 (+3,5 %)**	48,64 (+5,4 %)

Tableau 31 Précision moyenne de nos quatre modèles de prédiction (régression logistique, méthodes stricte et assouplie), espagnol

Nous constatons que les différences de précision moyenne avec Reverso pour la méthode stricte ne sont jamais statistiquement significatives. La meilleure

précision moyenne pour le modèle A, 46,65 %, s'obtient avec une tolérance de 20 % et combine en moyenne 4,3 traductions par requête. Le modèle C obtient 46,76 % en choisissant une tolérance de 25 % avec 5,01 traductions en moyenne.

Pour la méthode assouplie, les résultats sont similaires à ceux de la méthode normale. Le meilleur modèle est le D avec une tolérance de 15 % qui nous apporte une précision moyenne de 48,68 %. Le nombre moyen de traductions par requête est de 5,96. Le modèle A obtient une précision moyenne de 47,71 % avec une tolérance de 20 % et 2,31 traductions par requête. Ces deux derniers résultats sont admis comme statistiquement différents de Reverso par notre test au contraire des meilleures performances des modèles B et C. Le modèle B obtient 48,35 % pour une tolérance de 15 % avec 5,61 traductions en moyenne, alors que le modèle C atteint 47,72 % pour une tolérance de 25 % avec 6,23 traductions par requête en moyenne.

Fort de ces résultats encourageants pour la langue espagnole, nous avons voulu vérifier le comportement de la régression logistique et de nos différents modèles pour la collection-test allemande. Les variables sélectionnées par la procédure `stepAIC` sont présentées dans le tableau 32. On remarque que les variables sélectionnées pour les modèles C et D sont identiques. Cela correspond à nos attentes puisque les variables retirées dans D par rapport à C avant la procédure `stepAIC` étaient déjà retirées par cette procédure dans C.

En comparant les résultats pour l'allemand par rapport à l'espagnol, nous constatons d'une manière générale que la modification de l'ensemble de variables choisies au départ provoque moins de différences dans les variables sélectionnées par la procédure `stepAIC`.

Modèle	Variables choisies	Variables sélectionnées
A	Toutes	*source*, *max_idf2*, *avg_idf* et *personne*
B	Toutes sauf la source	*concepts*, *avg_idf* et *personne*
C	Données du système d'indexation uniquement	*concepts* et *avg_idf*
D	Même que C sans *max_idf2* et *min_idf2*	*concepts* et *avg_idf*
E	*avg_idf* seul.	Pas de sélection effectuée

Tableau 32 Variables retenues pour nos différents modèles de régression logistique (allemand)

En observant chaque modèle, nous remarquons qu'avec le modèle A, l'origine de la traduction (*source*) reste une variable importante. Ceci nous a un peu surpris car la différence entre les précisions moyennes des divers traducteurs est beaucoup moins importante en allemand (voir tableau 8 et tableau 19). De plus, deux variables liées à l'*idf* (*max_idf2* et *avg_idf*) ont été retenues, ainsi que l'information indiquant la présence ou l'absence de noms propres de personnes (*personne*).

Comme pour la langue espagnole, si on masque l'origine, le nombre de termes indexés dans la requête traduite (*concepts*) prend de l'importance. De plus, les variables *avg_idf* et *personne* sont conservées. Pour les modèles C et D, seules les variables *concepts* et *avg_idf* sont sélectionnées. Enfin, comme la variable *avg_idf* est présente dans tous les autres modèles, nous nous attendons à une performance du modèle E plus proche des autres modèles qu'en espagnol.

Pour la détermination des classes pour l'apprentissage, nous considérons qu'une traduction est la meilleure si sa précision moyenne s'écarte au plus de 5 % de la précision moyenne maximale pour cette requête. Pour l'évaluation, seule la méthode rétrospective a été utilisée.

Le tableau 33 présente la précision moyenne obtenue pour chacune de nos expérimentations ainsi que le pourcentage d'amélioration par rapport à notre meilleur traducteur, soit le système Reverso. Les valeurs sont soulignées lorsque la différence entre la précision moyenne de Reverso et celle de notre modèle est statistiquement significative d'après notre test (test du signe).

	Précision moyenne (% de changement)			
Modèle	A	B	C	E
Reverso	30,74	30,74	30,74	30,74
Méthode rétrospective				
tolérance de 0 %	30,37 (-1,2 %)	26,25 (-14,6 %)	26,71 (-13,1 %)	25,05 (-18,5 %)
tolérance de 5 %	31,73 (+3,2 %)	28,28 (-8,0 %)	27,88 (-9,3 %)	27,77 (-9,7 %)
tolérance de 10 %	32,16 (+4,6 %)	28,69 (-6,7 %)	28,77 (-6,4 %)	31,09 (+1,1 %)
tolérance de 15 %	32,02 (+4,2 %)	32,00 (+4,1 %)	31,18 (+1,4 %)	32,87 (+6,9 %)
tolérance de 20 %	32,53 (+5,8 %)	33,16 (+7,9 %)	33,30 (+8,3 %)	33,86 (+10,2 %)
tolérance de 25 %	33,09 (+7,6 %)	34,06 (+10,8 %)	33,84 (+10,1 %)	34,16 (+11,1 %)
tolérance de 30 %	33,18 (+7,9 %)	**34,39 (+11,9 %)**	34,30 (+11,6 %)	34,50 (+12,2 %)
tolérance de 35 %	**33,39 (+8,6 %)**	34,35 (+11,7 %)	**34,35 (+11,7 %)**	**34,62 (+12,6 %)**

Tableau 33 Précision moyenne de nos quatre modèles de prédiction (allemand)

Pour l'allemand, notre meilleur modèle est le modèle E, avec une précision moyenne de 34,62 % pour une tolérance de 35 %. Le nombre moyen de traductions par requête est quant à lui plus élevé et se situe à 7,95. Le modèle B obtient une précision moyenne de 34,39 % pour une tolérance de 30 % avec 6,87 traductions en moyenne. Le modèle C atteint 34,35 % pour une tolérance de 35 % avec 7,28 traductions en moyenne. Enfin, le modèle A obtient une précision moyenne de 33,39 % pour une tolérance de 35 % avec en moyenne 3,9 traductions par requête.

On remarque encore qu'avec une tolérance de 45 %, tous nos modèles proposent la concaténation de toutes les traductions, qui obtient une précision moyenne de 34,71 %. Aucun de nos modèle n'est donc en mesure de dépasser cette méthode simple. Par contre, tous nos modèles donnent des résultats dont la différence avec le meilleur outil de traduction pris isolément est statistiquement significative.

Notre système de sélection de bonnes traductions, requête par requête, basé sur la régression logistique, permet donc une amélioration significative par rapport au choix d'un seul outil de traduction tant pour l'espagnol que pour l'allemand. Par contre, notre système donne de meilleurs résultats que la concaténation des différentes traductions pour l'espagnol, mais pas pour l'allemand. Enfin, nous avons partiellement vérifié le modèle de prédiction de Cronen-Townsend *et al*. (2002) qui indique que l'*idf* moyen d'une requête s'avère être un bon prédicteur de la précision moyenne. En effet, si la moyenne de l'*idf* n'est pas toujours conservée, tous nos modèles contiennent au moins une variable basée sur la mesure de l'*idf*.

3.9 Fusion de résultats

Toutes les techniques que nous avons utilisées précédemment opéraient une modification de la requête avant la procédure d'indexation. Notre dernière méthode se positionne après l'indexation et l'appariement.

La fusion de données est une technique utilisée en recherche d'information pour mettre ensemble les résultats de plusieurs recherches différentes. Elle permet de réunir des résultats provenant soit de collections de documents différentes, par exemple dans le cas de la recherche multilingue (Savoy 2004b) ou à l'exemple des métamoteurs (Rasolofo *et al.* 2003), soit de la même collection en utilisant des modèles de pondération différents (Fox *et al.* 1994, Lee 1995) ou des traitements différenciés sur la requête. Dans cette expérience, nous nous situons dans cette dernière approche.

Nous effectuons une recherche pour chacune de nos traductions séparément, puis, une fois les listes de documents dépistés obtenues, nous opérons la fusion. Une liste de documents comprend pour chaque document deux valeurs que nous utiliserons : le *rang* du document attribué par le système et la valeur de similarité du document par rapport à la requête, valeur appelée *RSV*[14] ou *score* et calculée selon notre modèle d'indexation, en l'occurrence Okapi.

Il existe un grand nombre de méthodes pour sélectionner et classer les résultats lors de la fusion. Nous en avons évalué cinq dans nos expérimentations. Un document ne doit être inséré qu'une seule fois dans la liste de résultats retournée. Si un document est à nouveau proposé, il n'est pas inséré la seconde fois.

La première méthode, nommée *round robin*, est basée uniquement sur le rang attribué par le système. Nous prenons d'abord le premier document de chaque liste, puis le deuxième de chaque liste, et ainsi de suite jusqu'à l'épuisement des listes. Pour classer les documents d'un même rang, nous avons déterminé un ordre au préalable basé sur la performance des systèmes de traduction.

---

[14] *RSV* pour *retrieval status value,* la valeur de similarité attribuée à un document par le système de recherche pour une requête donnée, qui est utilisée pour classer les documents.

La deuxième méthode, appelée *maximum*, se base uniquement sur le score. À chaque itération, nous sélectionnons le document qui a le meilleur score parmi toutes les listes disponibles. Si plusieurs documents candidats sont à égalité, nous avons déterminé un ordre au préalable basé sur la performance des systèmes de traduction.

Ces deux méthodes ne tiennent pas compte de la présence d'un document dans plusieurs listes de résultats. Les trois suivantes vont s'y intéresser. D'abord, dans la méthode *somme*, nous additionnons simplement les scores obtenus par chaque document dans les différentes listes de résultats. Nous pouvons ensuite classer les documents en utilisant cette somme (tri par score décroissant).

Ensuite, nous allons utiliser les rangs en calculant la somme des inverses des rangs du document dans les différentes listes de résultats. Ainsi un document en deuxième position aura une valeur de ½ et un document en cinquième position aura une valeur de 1/5. Si le document est absent d'une liste, l'inverse de rang correspondante est définie égale à zéro. Nous classons à nouveau les documents dans l'ordre décroissant de la valeur trouvée.

Enfin, nous avons combiné ces deux dernières approches en calculant la somme des scores, chacun divisé par le rang correspondant comme indiqué dans la formule 13. Si le document est absent d'une liste de résultats, sa valeur est à nouveau définie comme zéro. Comme pour les dernières méthodes, nous classons les documents par valeur décroissante.

$$\mathrm{Val} = \sum_{t \in \mathrm{Traducteurs}} \frac{\mathrm{RSV}_t}{\mathrm{rang}_t} \tag{13}$$

Afin d'illustrer ces différentes méthodes, nous allons prendre un exemple. Nous disposons de trois listes de cinq résultats présentés dans le tableau 34.

Liste 1		Liste 2		Liste 3	
Doc	Score	Doc	Score	Doc	Score
D1	6,5	D2	5,44	D3	5,21
D4	6	D4	5,43	D5	4,8
D5	5,2	D5	5,42	D6	4,6
D7	4,6	D3	4,8	D1	4,2
D6	3,8	D6	4,6	D4	3,7

Tableau 34 Exemple fictif de listes de résultats

Le résultat de la fusion selon nos différentes méthodes est présenté dans le tableau 35. Les lignes tracées pour les méthodes *round robin* et maximum représentent les documents éliminés parce qu'ils sont déjà dans la liste en meilleure place. La liste finale ne contiendra donc chaque fois que sept documents.

Round robin	Maximum		Somme					
			Score		Inverse rang		Score / rang	
	Doc	Score	Doc	Valeur	Doc	Valeur	Doc	Valeur
D1	D1	6,5	D5	15,42	D1	1,25	D1	7,55
D2	D4	6	D4	15,13	D3	1,25	D4	6,455
D3	D2	5,44	D6	13	D4	1,2	D3	6,41
D4	~~D4~~	~~5,43~~	D1	10,7	D5	1,166…	D5	5,94
~~D4~~	D5	5,42	D3	10,01	D2	1	D2	5,44
D5	D3	5,21	D2	5,44	D6	0,733…	D6	3,2133…
~~D5~~	~~D5~~	~~5,2~~	D7	4,6	D7	0,25	D7	1,15
~~D5~~	~~D3~~	~~4,8~~						
D6	~~D5~~	~~4,8~~						
D7	D7	4,6						
~~D3~~	D6	4,6						
~~D1~~	~~D6~~	~~4,6~~						
~~D6~~	~~D1~~	~~4,2~~						
~~D6~~	~~D6~~	~~3,8~~						
~~D4~~	~~D4~~	~~3,7~~						

Tableau 35 Résultats de la fusion pour l'exemple

Afin d'évaluer ces différentes méthodes, nous avons travaillé avec les collections-tests espagnoles et allemandes en partant des requêtes écrites en langue anglaise. Ces collections sont décrites dans la section 3.2.1. Nous avons utilisé nos huit traducteurs. Comme nous ne voulions pas choisir a priori certains traducteurs plutôt que d'autres, nous avons évalué toutes les combinaisons possibles sans répétition de nos huit traducteurs automatiques avec nos cinq méthodes de fusion.

Le tableau 36 présente les cinq meilleures combinaisons de traductions en espagnol pour chacune de nos méthodes de fusion. Pour chaque combinaison, la précision moyenne ainsi que le pourcentage d'amélioration par rapport à notre meilleure traduction automatique, REVERSO, sont aussi donnés. Les valeurs sont soulignées lorsque la différence entre la précision moyenne de REVERSO et celle de notre modèle est statistiquement significative d'après notre test (test du signe).

Rang	Traducteurs	Précision moyenne
Round robin		
1	Reverso + FreeTranslation	44,59 (- 3,34 %)
2	Reverso + Systran	44,21 (- 4,16 %)
3	Reverso + Google	44,08 (- 4,44 %)
4	Reverso + Google + Systran	43,97 (- 4,68 %)
5	Reverso + Google + FreeTranslation	43,54 (- 5,61 %)
Maximum		
1	Reverso + FreeTranslation	48,48 (+ 5,09 %)
2	Reverso + Google + FreeTranslation	48,05 (+ 4,16 %)
3	Reverso + Systran + FreeTranslation	47,99 (+ 4,03 %)
4	Reverso + Google + Systran + FreeTranslation	47,98 (+ 4,01 %)
5	Reverso + Google + FreeTranslation + InterTran	47,88 (+ 3,79 %)
Somme Score		
1	Babylon 1 + Reverso + Systran + FreeTranslation	48,18 (+ 4,44 %)
2	Babylon 1 + Reverso + Google + FreeTranslation	48,18 (+ 4,44 %)
3	Babylon 1 + Reverso + FreeTranslation	48,07 (+ 4,21 %)
4	Babylon 1 + Reverso + Systran	47,91 (+ 3,86 %)
5	Babylon 1 + Reverso + Google	47,82 (+ 3,66 %)
Somme Inverse rang		
1	Reverso + FreeTranslation	45,32 (- 1,76 %)
2	Babylon 1 + Reverso + Systran + FreeTranslation	44,86 (- 2,75 %)
3	Babylon 1 + Reverso + Google + FreeTranslation	44,86 (- 2,75 %)
4	Babylon 1 + Reverso + FreeTranslation	44,84 (- 2,80 %)
5	Babylon 1 + Reverso	44,76 (- 2,97 %)
Somme score / rang		
1	Reverso + FreeTranslation	46,09 (- 0,09 %)
2	Babylon 1 + Reverso + FreeTranslation	45,79 (- 0,74 %)
3	Babylon 1 + Reverso + Systran + FreeTranslation	45,74 (- 0,85 %)
4	Babylon 1 + Reverso + Google + FreeTranslation	45,73 (- 0,87 %)
5	Reverso + Google	45,57 (- 1,21 %)

Tableau 36 Meilleures combinaisons de traducteurs pour la fusion avec leur précision moyenne (espagnol)

On constate que quelle que soit la méthode de fusion choisie, REVERSO est toujours présent dans les meilleures combinaisons. C'est d'ailleurs toujours le cas si nous prenons les dix meilleurs résultats de chaque modèle. Cela montre bien la prédominance de ce traducteur pour l'espagnol. Babylon 2 et Babylon 3 sont totalement absents alors qu'InterTran n'est présent qu'une seule fois. Or, ces trois derniers traducteurs pris isolément sont ceux qui proposent les moins bonnes performances (voir tableau 8).

Concernant la performance des différentes méthodes, seules les méthodes *Maximum* et *Somme des Scores* permettent d'obtenir une amélioration par rapport à notre meilleure traduction automatique. Mais en effectuant notre test statistique, nous constatons que seuls trois résultats de la méthode *maximum* présentent une amélioration statistiquement significative de la précision moyenne. Tous les autres résultats sont donc trop proches de REVERSO pour que notre test nous permette de les considérer comme différents.

Rang	Traducteurs	Précision moyenne
Round robin		
1	Reverso + Google	31,64 (+ 2,93 %)
2	Reverso + Google + Systran	31,55 (+ 2,64 %)
3	Reverso + Systran	31,42 (+ 2,21 %)
4	Babylon 1 + Reverso + Google	30,99 (+ 0,81 %)
5	Babylon 1 + Reverso + Google + Systran	30,96 (+ 0,72 %)
Maximum		
1	Reverso + Google + Systran + FreeTranslation	34,32 (+ 11,65 %)
2	Reverso + Google + FreeTranslation	34,25 (+ 11,42 %)
3	Babylon 1 + Reverso + Google + Systran + FreeTranslation	34,02 (+ 10,67 %)
4	Babylon 1 + Reverso + Google + FreeTranslation	33,92 (+ 10,34 %)
5	Reverso + Systran + FreeTranslation	33,63 (+ 9,40 %)
Somme Score		
1	Babylon 2 + Babylon 3 + Reverso + Google + Systran + FreeTranslation	34,27 (+ 11,48 %)
2	Babylon 1 + Babylon 2 + Babylon 3 + Reverso + Google + Systran + FreeTranslation	34,25 (+ 11,42 %)
3	Babylon 1 + Babylon 2 + Reverso + Systran + FreeTranslation	34,22 (+ 11,32 %)
4	Babylon 3 + Reverso + Systran + FreeTranslation	34,21 (+ 11,29 %)
5	Babylon 1 + Babylon 3 + Reverso + Systran + FreeTranslation	34,18 (+ 11,19 %)
Somme Inverse rang		
1	Babylon 1 + Babylon 3 + Reverso + Google + Systran + FreeTranslation	32,86 (+ 6,90 %)
2	Babylon 2 + Reverso + Google	32,85 (+ 6,86 %)
3	Babylon 1 + Babylon 2 + Reverso + Google + Systran + FreeTranslation	32,83 (+ 6,80 %)
4	Babylon 3 + Reverso + Google + Systran + FreeTranslation	32,71 (+ 6,41 %)
5	Babylon 2 + Reverso + Google + Systran + FreeTranslation	32,68 (+ 6,31 %)
Somme score / rang		
1	Babylon 2 + Reverso + Google + Systran + FreeTranslation	33,29 (+ 8,30 %)
2	Babylon 2 + Reverso + Google + FreeTranslation	33,24 (+ 8,13 %)
3	Babylon 1 + Reverso + Google + FreeTranslation	33,17 (+ 7,91 %)
4	Babylon 1 + Babylon 2 + Reverso + Google + FreeTranslation	33,17 (+ 7,91 %)
5	Babylon 1 + Babylon 2 + Reverso + Google + Systran + FreeTranslation	33,15 (+ 7,84 %)

Tableau 37 Meilleures combinaisons de traducteurs pour la fusion avec leur précision moyenne (allemand)

Le tableau 37 présente les cinq meilleures combinaisons de traductions en allemand pour chacune de nos méthodes de fusion. Pour chaque combinaison, la précision moyenne ainsi que le pourcentage d'amélioration par rapport à notre meilleure traduction automatique, REVERSO, sont aussi donnés. Les valeurs sont soulignées lorsque la différence entre la précision moyenne de REVERSO et celle de notre modèle est statistiquement significative d'après notre test (test du signe).

Comme pour l'espagnol, le meilleur traducteur, REVERSO, est présent dans les meilleures combinaisons quelle que soit la méthode de fusion et le moins bon, InterTran, est systématiquement absent. Nous constatons aussi que pour

l'allemand, toutes les méthodes de fusion permettent une amélioration par rapport à notre référence, REVERSO. Mais aucune ne permet de dépasser la combinaison simple de toutes les traductions qui obtient une précision moyenne de 34,71 %.

Les méthodes *Maximum* et *Somme des scores* donnent à nouveau les meilleurs résultats, avec une amélioration significative pour les cinq meilleures performances de la deuxième.

Cette technique utilisant la fusion de résultats nous donne donc des résultats intéressants. Par contre, elle nécessite un bon choix de traducteurs à fusionner, l'utilisation de tous les traducteurs provoquant une détérioration de la précision moyenne.

D'autre part, cette technique requiert plusieurs recherches qui sont mises ensemble après coup, ce qui nécessite plus de ressources au moment de la recherche proprement dite que dans le cas des techniques d'apprentissage automatique. En effet, l'apprentissage automatique nécessite un grand travail préparatoire, mais une fois celui-ci terminé, la classification d'un nouvel élément est une opération facile. Comme les précisions moyennes obtenues par nos méthodes de fusion ne sont pas meilleures que celles obtenues à l'aide des techniques d'apprentissage automatique, nous donnerons la préférence à ces dernières.

3.10 Récapitulatif des résultats

Le tableau 38 présente le meilleur résultat obtenu pour chacune des approches évaluées ci-dessus. La première ligne présente la précision moyenne obtenue par la meilleure traduction seule, celle obtenue par le système REVERSO. Ensuite, la précision moyenne du meilleur résultat de chaque approche est indiquée avec le pourcentage d'amélioration par rapport à la meilleure traduction. Les valeurs sont soulignées si cette amélioration est jugée statistiquement significative par le test du signe.

	Précision moyenne (% d'amélioration)	
	ES	DE
Reverso (meilleure traduction seule)	46,13	30,74
Traduction inverse	40,69 (- 11,8 %)	N/A
Combinaisons	47,31 (+ 2,6 %)	34,73 (+ 13,0 %)
Plus proche voisin	48,44 (+ 5,0 %)	N/A
3-plus proches voisins	47,56 (+ 3,1 %)	N/A
Génération d'arbres de décision	47,97 (+ 4,0 %)	34,42 (+ 12,0 %)
Régression logistique (rétrospective)	48,84 (+ 5,9 %)	34,62 (+ 12,6 %)
Régression logistique (*leaving one out*)	48,75 (+ 5,7 %)	N/A
Fusion de résultats (maximum)	48,48 (+ 5,1 %)	34,32 (+ 11,6 %)
Fusion de résultats (somme)	48,18 (+ 4,4 %)	34,27 (+ 11,5 %)

Tableau 38 Récapitulatif des meilleurs résultats de chaque expérience.

Pour la traduction inverse, le meilleur résultat est obtenu avec le traducteur REVERSO, la lemmatisation de la requête d'origine et en conservant tels quels les

mots de la requête traduite qui ne se trouvent pas dans le dictionnaire espagnol-anglais. On constate une détérioration de près de 12 %.

Le meilleur résultat de nos systèmes de combinaison est la concaténation simple des différentes traductions tant pour l'allemand que pour l'espagnol.

Pour la méthode du plus proche voisin, la meilleure méthode nous fait choisir les variables *avg_idf*, *max_idf*, *min_idf2*, *personne* et *autre_nom* pour obtenir une amélioration significative de 5 %. La méthode des 3-plus proches voisins obtient une amélioration non significative de 3,1 % avec les variables *source*, *min_idf*, *sigle*, *autre_nom* et *date*.

Dans le cas de la génération d'arbres de décision, les meilleurs résultats sont obtenus avec une tolérance de 5 % lors de l'apprentissage en espagnol et une tolérance de 10 % en allemand. Ces deux résultats sont statistiquement significatifs. On notera aussi qu'en espagnol, si nous utilisons une tolérance de 10 %, nous obtenons un résultat très légèrement inférieur mais qui présente tout de même une amélioration statistiquement significative.

Les meilleurs résultats pour la régression logistique en espagnol tant en méthode rétrospective que *leaving one out* nous font utiliser les variables *concepts*, *min_idf*, *personne*, *géo* et *autre_nom*. Une tolérance de 5 % est nécessaire sur la détermination de bonnes traductions lors de l'apprentissage, avec une tolérance de 15 % lors du choix de la traduction. Pour l'allemand, nous utilisons les variables *concepts* et *avg_idf* avec une tolérance de 5 % à l'apprentissage et de 35 % lors du choix de la traduction.

Pour la fusion de résultats, avec la méthode *maximum*, nous utilisons la fusion des traducteurs REVERSO et FREETRANSLATION pour obtenir le meilleur résultat en espagnol. Pour l'allemand, nous choisissons les traducteurs REVERSO, GOOGLE, SYSTRAN et FREETRANSLATION. Pour la méthode *Somme des scores*, nous travaillons avec BABYLON 1, REVERSO, SYSTRAN et FREETRANSLATION en espagnol et BABYLON 2, BABYLON 3, REVERSO, GOOGLE, SYSTRAN et FREETRANSLATION en allemand.

Nous constatons enfin que la régression logistique nous permet d'obtenir nos meilleurs résultats toutes approches confondues, les améliorations étant jugées statistiquement significatives par le test du signe par rapport à la meilleure traduction (REVERSO).

4 Conclusion

4.1 Contributions

En premier lieu, ce travail se situe dans le contexte relativement nouveau et récent de la recherche d'information bilingue. Nous avons d'abord présenté un ensemble de six stratégies destinées à améliorer l'utilisation de traducteurs automatiques. De cette étude, les contributions suivantes peuvent être dégagées.

Le renforcement de traduction par l'utilisation de la traduction inverse donne des résultats très limités. Nous y voyons deux raisons. D'abord, les dictionnaires disponibles sont insuffisants, notamment au niveau de la traduction de noms propres ou d'acronymes. Ensuite, cette technique se borne à supprimer des mots et la baisse de performance due au retrait par erreur de mots pertinents est bien plus importante au final que l'amélioration obtenue en retirant des mots non pertinents.

Notre deuxième technique, proposant plusieurs manières alternatives de combiner les diverses traductions d'une requête, nous a permis de montrer que la concaténation simple des traductions, méthode utilisée jusqu'à présent par notre équipe lors des diverses campagnes d'évaluation, était la meilleure que nous ayons testée.

Nos trois techniques d'apprentissage automatique nous permettent d'améliorer la précision moyenne de la recherche lors de l'utilisation de plusieurs traducteurs en les sélectionnant requête par requête. Les expériences effectuées sur l'allemand comme sur l'espagnol ont donné des résultats semblables pour les trois techniques. La méthode des *k*-plus proches voisins donne des résultats probants si *k* vaut un, mais ne peut être utilisée pour d'autres valeurs de *k*. La génération automatique d'arbres de décision nous permet aussi d'améliorer la précision moyenne si la manière de classifier une traduction comme bonne n'est pas trop stricte. De plus, la régression logistique nous a permis d'obtenir nos meilleurs résultats tant en espagnol qu'en allemand.

Enfin, nous avons constaté que l'utilisation de la fusion de résultats après l'indexation permettait aussi d'obtenir une amélioration significative de la précision moyenne. Malheureusement, cette technique est plus gourmande en ressources machines (mémoire et temps de calcul) que les autres au moment de l'appariement. Comme l'amélioration obtenue est équivalente à celle des techniques d'apprentissage automatique, c'est vers celles-ci que nous aurions tendance à nous tourner.

4.2 Limites

Nos différentes techniques ont donné des résultats intéressants, mais nos collections-test ont certains avantages qui ne sont pas forcément reproduits dans tous les cas. En premier lieu, il faut un certain nombre d'outils de traduction automatique efficaces pour pouvoir opérer une sélection. Certaines langues traitées actuellement sont relativement pauvres en outils linguistiques (par exemple le bulgare ou le finnois), ou un outil démontre une efficacité tellement supérieure qu'il est inutile de le combiner avec d'autres dont les performances sont assez moyennes.

Ensuite, afin de pouvoir travailler correctement avec des systèmes basés sur l'apprentissage automatique, un nombre relativement élevé de données déjà évaluées doivent exister. De plus, les requêtes évaluées doivent être suffisamment générales pour que l'apprentissage ne soit pas biaisé et puisse être utile dans l'avenir.

4.3 Perspectives

Dans plusieurs cas, nous avons utilisé tous les traducteurs en partant du principe que les moins bons seraient automatiquement éliminés par le système. Nous pourrions essayer de ne choisir que les traducteurs qui nous donnent les meilleurs résultats afin d'essayer d'éliminer certaines sources d'erreur. Ceci serait à double tranchant, puisqu'un traducteur pourrait être éliminé alors que c'est lui qui nous donnait le bon résultat pour certaines requêtes, par exemple spécifiques à un domaine.

L'usage d'outils externes, comme des listes de noms propres traduits pourrait aussi être une source alternative de traduction destinée à être combinée avec nos traductions automatiques afin d'améliorer les requête.

Enfin, certaines de nos techniques qui semblent prometteuses devraient être évaluées avec des langues fondamentalement différentes de nos collections-test actuelles, comme les langues asiatiques, le finnois ou le basque.

5 Bibliographie

AllTheWeb Search Engine, http://www.alltheweb.com/info/about/index (visité le 23 mai 2005)

Ballesteros L., Croft, B. W., "Resolving ambiguity for cross-language retrieval", in *Proceedings of the 21st International Conference of the ACM-SIGIR'98*, The ACM Press, New York, 1998, p. 64-71.

Bookstein A., O'Neil E., Dillon M., Stephen D., "Applications of loglinear models for informetric phenomena", *Information Processing & Management*, vol. 28(1), 1992, p. 75-88.

Braschler M., Ripplinger B., Schäuble P., "Experiments with the Eurospider retrieval system for CLEF 2001", in *Evaluation of Cross-Language Information Retrieval Systems*", Lecture Notes in Computer Science, vol. 2046, Springer-Verlag, Berlin, 2002, p. 102-110.

Braschler M., Peters C., "CLEF 2003 methodology and metrics", in *Comparative Evaluation of Multilingual Information Access Systems*", Lecture Notes in Computer Science, vol. 3237, Springer-Verlag, Berlin, 2004, p. 7-20.

Brown P. F., Cocke J., Della Pietra S. A., Della Pietra V. J., Jelinek F., Lafferty J. D., Mercer R. L., Roosin P. S., "A statistical approach to machine translation", *Computational Linguistics*, vol 16(2), 1990, p. 79-85.

Buckley C., Salton G., Allan J., Singhal A., "Automatic Query Expansion Using SMART: TREC 3", in *Overview of the Third Text REtrieval Conference (TREC-3)*, number SP 500-225, NIST Special Publication, Gaithersburg, 1994, p. 69-80.

Buckley C., Singhal A., Mitra M., Salton G., « New retrieval approaches using SMART», *Proceedings of the TREC'4*, Gaithersburg, 1995, p. 25-48.

Buckley C., Mitra M., Waltz J., Cardie C., "Using clustering and superconcepts within SMART : TREC 6" in *Proceedings of TREC'6*, number SP 500-240, NIST Special Publication, Gaithersburg, 1998, p. 107-124.

Calvet L.-J., "La guerre des langues et les chances d'un véritable plurilinguisme", *Panoramiques*, n° 48, Corlet, Condé-sur-Noireau, 2000, p. 10-17.

Chen A., "Multilingual information retrieval using English and Chinese queries", in *Evaluation of Cross-Language Information Retrieval Systems*", Lecture Notes in Computer Science, vol. 2046, Springer-Verlag, Berlin, 2002, p. 44-58.

Chen A., Gey F., "Building an Arabic stemmer for information retrieval", in *The Eleventh Text Retrieval Conference*, number SP 500-251, NIST Special Publication, Gaithersburg, 2002.

Chen A., Gey F., "Combining query translation and document translation in cross-language retrieval", in *Comparative Evaluation of Multilingual Information Access Systems*", Lecture Notes in Computer Science, vol. 3237, Springer-Verlag, Berlin, 2004, p. 108-121.

Cleverdon C. W., "The Cranfield tests on index language devices", *Aslib Proceedings*, volume 19, 1967, p. 173-192, (Réimprimé in *Readings in Information Retrieval*, Spark-Jones K. and Willet P. (ed.), Morgan Kaufmann, San Francisco, 1997, p. 47-59).

Conover, W.J., *Practical nonparametric statistics*, 2nd Ed., John Wiley & Sons, New-York, 1980.

Cronen-Townsend S., Zhou Y., Croft W.B., « Predicting query performance », *Proceedings of the ACM-SIGIR'2002*, Tampere, 11–15 August 2002, The ACM Press, New York, p. 299-306.

Fox C., "A stop list for general text", *ACM SIGIR Forum*, volume 24 (1-2), ACM Press, New-York, 1990, p. 19-35.

Fox E.A., Shaw J.A., "Combination of multiple searches". In: Harman DK, ed., *Proceedings TREC-2*, NIST Publication #500-215, Gaithersburg, 1994, p. 243-249.

Frakes W., Baeza-Bates R. (Eds), *Information retrieval: Data structures and algorithms*, Prentice-Hall, Englewood Cliffs, 1992.

Gauthier G., "L'Inde aux 1652 langues", *Panoramiques*, n° 48, Corlet, Condé-sur-Noireau, 2000, p. 72-75.

Gey F., " Inferring probability of relevance using the method of logistic regression" in *Proceedings of the 17th International Conference of the ACM-SIGIR'94*, New York, 1994, p. 222-231.

Gey F., "Chinese and Korean topic search of Japanese news collections" in *Working Notes of the Fourth NTCIR Workshop on Asian Language Retrieval and Question Answering*, Tokyo, 2004, p. 214-218.

Global Reach, http://glreach.com/globstats/ (visité le 23 mai 2005)

Google Search Engine, http://www.google.ch/intl/fr/corporate/ (visité le 13 avril 2005).

Harman D. K., "How effective is suffixing", *Journal of the American Society for Information Science*, volume 42 (1), 1991, p. 7-15.

Harman D. K., "The TREC Confernces", in Kuhlen R. and Rittberger M. (éd.), *Hypertext – Information Retrieval – Multimedia: Synergieeffekte Elektronischer Informationssysteme, Proceeding of HIM'95*, Universitätsverlag Konstanz, 1995, p. 9-28, (Réimprimé in *Readings in Information Retrieval*, Spark-Jones K. and Willet P. (ed.), Morgan Kaufmann, San Francisco, 1997, p. 247-256).

Hosmer D.W., Lemeshow S., *Applied Logistic Regression*, 2nd Ed., John Wiley & Sons, New-York, 2000.

Hull D.A., "Stemming algorithms: a case study for detailed evaluation", *Journal of the American Society for Information Science*, volume 47 (1), 1996, p. 70-84.

Internet World Stats, http://www.internetworldstats.com/ (visité le 23 mai 2005).

Jones G.J.F., "Beyond English text: Multilingual and multimedia information retrieval", in Tait J.I. (ed.), *Chartering a New Course: Natural Language Processing and Information Retrieval. Essays in Honour if Karen Spärck Jones*, Springer, Dordrecht, 2005, p. 81-97.

Kamps J., Fissaha Adafre S., de Rijke M., "Effective translation, tokenization and combination for cross-lingual retrieval", in *Multilingual Information Access for Text, Sppech and Images*, Lecture Notes in Computer Science, vol. 3491, Springer-Verlag, Berlin, 2005, p. 123-134.

Koehn P., "Europarl: A multilingual corpus for evaluation of machine translation", 2003, http://people.csail.mit.edu/koehn/publications/europarl/ (visité le 14 novembre 2005).

Kraaij W., Pohlmann R., "Viewing stemming as recall enhancement", in Frei H.-P., Harman D., Schauble P, Wilkinson R. (dir.), *Proceedings of ACM-SIGIR 96*, Zürich,1996, p. 40-48.

Kwok K. L., Grunfeld L., Dinstl N., Chan M., "TREC-9 cross-language, Web and question-answering track experiments using PIRCS", in *Proceedings TREC-9*, NIST Publication #500-249, Gaithersburg, 2001, p. 417-426.

Lam-Adesina A.M., Jones G.J.F., "Exeter at CLEF 2003: Experiments with machine translation for monolingual, bilingual and multilingual retrieval", in *Comparative Evaluation of Multilingual Information Access Systems*", Lecture Notes in Computer Science, vol. 3237, Springer-Verlag, Berlin, 2004, p. 271-285.

Le Calvé A., Savoy J., "Database merging strategy based on logistic regression", *Information Processing & Management*, vol. 36(3), 2000, p. 341-359.

Lee J. H., "Combining multiple evidence from different properties of weighting schemes" in *ACM-SIGIR 1995* , Seattle, 1995, p. 180-188

Lovins J.B., "Development of a stemming algorithm", in *Mechanical Translation and Computational Linguistic*, vol. 11, p. 22-33.

McNamee P., Mayfield J., "JHU/APL experiments in tokenization and non-word translation", in *Comparative Evaluation of Multilingual Information Access Systems*", Lecture Notes in Computer Science, vol. 3237, Springer-Verlag, Berlin, 2004, p. 85-97.

Mitchell T.M., *Machine Learning*, McGraw-Hill, Singapour, 1997.

Nielsen J., "When search engines become answer engines", Jakob Nieslen's Alertbox, August 16, 2004, http://www.useit.com/alertbox/20040816.html (visité le 29 mars 2005)

NUA Ltd, http://www.nua.ie/surveys/ (visité le 23 mai 2005)

Porter M.F., "An algorithm for suffix stripping", *Program* vol. 14, 1980, p. 130-137.

Quinlan J. R., *C4.5 : Programs for Machine Learning*, Morgan Kaufman Publishers, San Mateo, 1993.

Rasolofo Y., Hawking D., Savoy J., "Result merging strategies for a current news metasearcher", *Information Processing & Management*, 2003, 39(4), p. 581-609.

Robertson S. E., "The probability ranking principle in IR", *Journal of Documentation*, vol. 33(4), 1977, p. 294-304.

Robertson S. E., Walker S., "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval" in *Proceedings of the 17th International Conference of the ACM-SIGIR'94*, New York, 1994, p. 232-241.

Robertson S.E., Walker S., Beaulieu M., « Experimentation as a way of life: OKAPI at TREC », *Information Processing & Management*, vol. 36(1), 2000, p. 95-108.

Rocchio J.J, "Relevance feedback in information retrieval". in Salton G. (Ed.) *The SMART retrieval system – experiments in automatic document processing*, Prentice-Hall Inc., Englewood Cliffs, 1971, p. 313-323.

Salton G. *The SMART retrieval system – experiments in automatic document processing*, Prentice-Hall Inc., Englewood Cliffs, 1971.

Salton G., McGill M., *Introduction to Modern Information Retrieval*, McGraw-Hill, New-York, 1983.

Salton G., Buckley C., "Term weighting approaches in automatic text retrieval", *Information Processing & Management*, vol. 24(5), 1988, p. 513-523.

Savoy J., "A stemming procedure and stopword list for general French corpora", *Journal of the American Society for Information Science*, volume 50 (10), 1999, p. 944-952.

Savoy J., "Report on CLEF-2001 experiments: Effective combined query translation approach", in *Evaluation of Cross-Language Information Retrieval Systems*", Lecture Notes in Computer Science, vol. 2046, Springer-Verlag, Berlin, 2002, p. 27-43.

Savoy, J., "Report on CLEF-2002 Experiements: Combining multiple sources of evidence", in Peters C., Braschler M., Gonzalo J., Kluck M. (Ed.), *Advances in Cross-Language Information Retrieval*, Lecture Notes in Computer Science, vol. 2785, Springer-Verlag, Berlin, 2003, p. 66-90.

Savoy J., Berger P.-Y., "Recherche bilingue et multilingue d'information, vers une sélection des bonnes traductions", in *Coria'04 Actes des conférences*, IRIT, Toulouse, 2004, p. 271-286.

Savoy J., "Combining multiple strategies for effective cross-language retrieval", in *IR Journal*, 7(1-2), 2004, 121-148.

Savoy J., "Effective multilingual search using Asian languages", *ACM Transactions on Asian Languages Information Processing*, vol 4(3), ACM Press, New-York, 2005.

Singhal A., Choi J., Hindle D., Lewis D.D., Pereira F., « AT&T at TREC-7 », *Proceedings of the TREC-7*, Gaithersburg, 9-11 November 1998, p. 239-251.

TreeTagger Projekt, http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/ (visité le 28 janvier 2006).

Van Rijsbergen C.J., *Information Retrieval*, 2nd ed., Butterworths, London, 1979.

Venaples W.N., Ripley B.D., *Modern Applied Statistics with S-PLUS*, Spinger-Verlag, New York, 1999.

Voorhees E. M., Gupta N. K., Johnson-Laird B., "The collection fusion problem" in *Overview of the Third Text REtrieval Conference (TREC-3)*, number SP 500-225, NIST Special Publication, Gaithersburg, 1995, p. 95-105.

Voorhees E. M., Harman D., "Overview of the sixth text retrieval conference (TREC-6)", *Information Processing & Management*, vol. 36(1), 2000, p. 3-35.

Voorhees E. M., "The philosophy of information retrieval evaluation", in *Evaluation of Cross-Language Information Retrieval Systems*", Lecture Notes in Computer Science, vol. 2046, Springer-Verlag, Berlin, 2002, p. 355-370.

Zobel J., "How reliable are the results of large-scale information retrieval experiments?", in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, Melbourne,1998, p. 307-314.

Annexe 1 Formules de pondération

Afin d'attribuer un poids w_{ij} reflétant l'importance de chaque terme d'indexation T_j, j = 1, 2, ..., t, dans un document D_i, nous pouvons recourir à l'une des formules décrites dans le tableau ci-dessous. Dans cette dernière, tf_{ij} indique la fréquence d'occurrences du terme T_j dans le document D_i (ou dans la requête), n représente le nombre de documents D_i dans la collection, df_j le nombre de documents dans lesquels le terme T_j apparaît (fréquence documentaire), et idf_j l'inverse de la fréquence documentaire ($idf_j = \ln[n/df_j]$). Les constantes ont été fixées aux valeurs suivantes : slope=0,2, pivot =150, b=0,75, k=2, k_1 =1,2, avdl=900. De plus, la longueur du document D_i (ou le nombre de termes d'indexation associé à ce document) est notée par nt_i, la somme de valeurs tf_{ij} par l_i et $K = k \cdot [(1 - b) + b \cdot (l_i/avdl)]$.

bnn	$w_{ij} = 1$	nnn	$w_{ij} = tf_{ij}$
ltn	$w_{ij} = [\ln(tf_{ij}) + 1] \cdot idf_j$	atn	$w_{ij} = \left[0{,}5 + 0{,}5 \cdot \frac{tf_{ij}}{\max tf_{i.}}\right] \cdot idf_j$
dtn	$w_{ij} = \ln[[\ln(tf_{ij}) + 1] + 1] \cdot idf_j$	npn	$w_{ij} = tf_{ij} \cdot \ln\left[\left(n - df_j\right) \Big/ df_j\right]$
Lnu	$w_{ij} = \frac{\left(1 + \ln(tf_{ij}) \Big/ \ln(\text{mean tf}) + 1\right)}{(1 - \text{slope}) \cdot \text{pivot} + \text{slope} \cdot nt_i}$	Okapi	$w_{ij} = \left((k_1 + 1) \cdot tf_{ij}\right) \Big/ \left(K + tf_{ij}\right)$
lnc	$w_{ij} = \frac{\ln(tf_{ij}) + 1}{\sqrt{\sum_{k=1}^{t} \left(\ln(tf_{ik}) + 1\right)^2}}$	ntc	$w_{ij} = \frac{tf_{ij} \cdot idf_j}{\sqrt{\sum_{k=1}^{t} \left(tf_{ik} \cdot idf_k\right)^2}}$
ltc	$w_{ij} = \frac{\left(\ln(tf_{ij}) + 1\right) \cdot idf_j}{\sqrt{\sum_{k=1}^{t} \left(\left(\ln(tf_{ik}) + 1\right) \cdot idf_k\right)^2}}$		
Lnu	$w_{ij} = \frac{\left(1 + \ln(tf_{ij}) \Big/ \ln(\text{mean tf}) + 1\right)}{(1 - \text{slope}) \cdot \text{pivot} + \text{slope} \cdot nt_i}$		

Tableau 39 Formules de pondération

Annexe 2 Nombre d'utilisateurs d'Internet par langue

Le tableau suivant a été compilé à partir des statistiques disponibles au sujet d'Internet sur les sites de (Global Reach), (Internet World Stats) et (Nua Ltd). La figure représente ces même données sous forme de proportion.

	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005
Anglais	40	72	91	148	192	231	234	288	290	300
Chinois	0	1	2	10	31	48	78	103	160	220
Japonais	2	7	9	20	39	48	61	70	85	105
Espagnol	0	1	2	13	21	35	50	66	70	80
Allemand	1	4	6	14	22	37	43	53	62	71
Coréen	0	0	1	5	17	25	28	30	35	40
Français	0	2	3	10	17	18	23	28	40	49
Italien	0	1	2	10	12	20	24	26	35	42
Portuguais	0	0	1	4	11	14	19	26	32	38
Scandinaves	2	2	3	8	9	11	14	15	16	17
Néerlandais	0	1	2	6	7	11	13	12	14	15
Autre	2	11	15	6	29	41	64	89	129	142
Total	47	102	137	254	407	539	651	806	968	1119

Tableau 40 Nombre d'utilisateurs d'Internet par langue entre 1996 et 2005 (en millions)

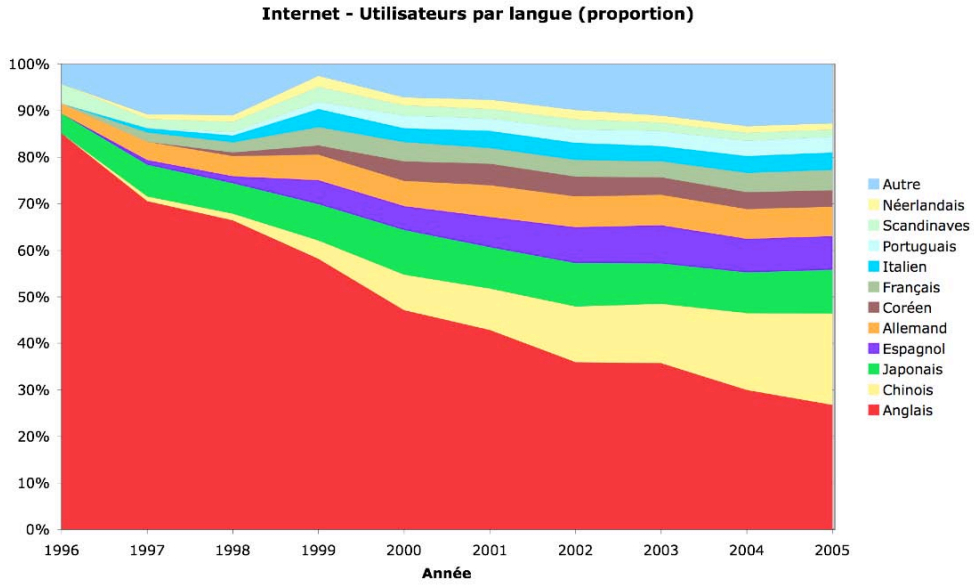

Figure 10 Proportion des utilisateurs d'Internet par langue entre 1996 et 2005

Table des figures

Table des tableaux