Similarités entre textes basées sur les noms propres

N. Friburger[1], D. Maurel[1]
[1]*LI (Laboratoire d'Informatique de Tours)*
64, avenue Jean Portalis, 37000 Tours
{friburger, maurel}@univ-tours.fr

RÉSUMÉ. *Les noms propres représentent environ 10 % du texte d'un article de journal. Leur quantité et leur qualité informationnelle sont déjà utilisées dans les systèmes d'extraction d'informations (conférences MUC). Nous avons créé un outil basé sur une description linguistique sous forme de transducteurs à nombre finis d'états. Les noms propres extraits sont alors utilisés dans un but de recherche d'information : il s'agit de présenter aux utilisateurs des textes journalistiques sous la forme d'une hiérarchie et de fournir une description des sujets traités dans les textes. Dans cet article, nous présentons une mesure de similarité permettant de mettre en valeur les noms propres et d'améliorer ainsi la classification automatique de textes. Cette mesure fusionne une similarité sur tous les mots du texte avec une similarité avec les mots seuls.*

ABSTRACT. Proper names represent about 10% of newspaper articles in English or French texts. Their quantity and informational quality are already used in different Information Extraction systems. Proper names have widely been studied in the MUC conferences designed to promote research in Information Extraction. We have created our own named entity extraction tool based on a linguistic description with automata. The extracted names are used in an information retrieval process: we want to cluster journalistic texts with a high precision level and to provide a topic description of the clusters. We verify the interest of the use of proper names in a similarity measure to improve clustering. This measure merge a similarity based on all the words with a similarity based on the proper names.

MOTS-CLÉS : *similarité, classification hiérarchique, noms propres.*

KEYWORDS: *similarity, hierarchic clustering, proper names.*

1. Introduction

Le traitement automatique des langues peut apporter des améliorations non négligeables à la fouille de textes. Les noms propres ont été largement étudiés dans le domaine de l'extraction d'information ; nous pensons qu'ils peuvent aussi jouer un rôle dans les systèmes de recherche d'information. C'est pourquoi nous proposons de faire ressortir les qualités sémantiques des noms propres à travers une mesure de similarité pour classer des articles de journaux entre eux. La quantité de noms propres et leur qualité informationnelle dans ce type de textes les rend pertinents pour améliorer la classification non supervisée grâce à une mesure de similarité qui les met en valeur par rapport aux autres mots d'un texte.

L'hypothèse de l'importance des noms propres présents dans un texte pour le classer semble prometteuse. Dans cet article, nous présentons la tâche MUC relative à l'extraction des entités nommées (dont les noms propres font parti) et notre système d'extraction et de catégorisation des noms propres. Dans une deuxième partie, nous proposons les mesures de similarité à base de noms propres que nous avons créées et testées. Enfin, nous expliquons comment nous avons procédé pour comparer les différentes classifications hiérarchiques obtenues avec nos mesures de similarité et nous décrivons les résultats obtenus.

2. L'extraction des noms propres

L'extraction des entités nommées est une tâche définie pour un usage d'extraction d'information. Avant le programme américain d'évaluation MUC[1], les recherches basées sur les noms propres existaient mais étaient insuffisantes. MUC propose de faire concourir des systèmes à la tâche d'extraction d'information ou à des sous-tâches liées à ce domaine, notamment l'extraction des entités nommées (NET – Named Entity Task).

La 6^{éme} édition de MUC, en 1995, propose pour la première fois cette tâche d'extraction des entités nommées. MUC-6 structure cette tâche et distingue trois types d'entités à reconnaître et catégoriser : ENAMEX, TIMEX et NUMEX (Chinchor, 1997). TIMEX contient les expressions de temps et NUMEX les nombres et pourcentages. Les entités de type ENAMEX sont composées par les noms propres ou assimilés et par les acronymes. Les entités ENAMEX sont de trois sortes :

- Organisations : noms de sociétés, gouvernements et autres entités organisationnelles,
  Ex : *Organisation des Nations Unies* ou *Microsoft*

[1] Message Understanding Conference

- Personnes : noms de personnes Ex : *Bill Clinton,* ou de familles Ex : *les Kennedy*
- Noms de lieux : lieux définis politiquement ou géographiquement (villes, départements, régions internationales, hydronymes, montagnes, …)
  Ex : *Allemagne*, *Paris, Silicon Valley*

La tâche d'extraction des entités nommées définie par MUC a donné de très bons résultats dès sa première édition (Grishman et al.,.1996). La plupart des systèmes participant à MUC-6 ont obtenu des scores de rappel et précision supérieurs à 90%. Il est intéressant de souligner qu'un expert humain obtient un rappel de 97% à cette tâche.

2.1. Les noms propres en Traitement Automatique des Langues

Les noms propres apportent une information importante sur le sens et le contenu des textes. Par exemple, un texte parlant de *Bill Clinton* et de *Monica Lewinski* évoque rapidement une affaire qui a défrayé la chronique il y a quelques années. Nous voulons utiliser cette capacité des noms propres à suggérer le sujet d'un article. Pour pouvoir utiliser les noms propres, il faut d'abord les reconnaître. Or ils sont trop nombreux pour être placés dans un dictionnaire et un grand nombre d'entre eux sont inconnus (Mani et al., 1996) et fugaces (les noms d'entreprises, par exemple, apparaissent et disparaissent au gré des créations, fusions et faillites). Les noms propres appartiennent à une classe très ouverte de mots.

(Mikheev et al., 1999) a étudié l'usage et l'impact des très gros dictionnaires et a montré que les noms de lieux nécessitent un dictionnaire pour être extraits : ils sont relativement stables et peu nombreux. Le projet Prolex, présenté dans (Maurel et al., 1999), propose un dictionnaire de noms de lieux assez complet que nous utilisons.

Les noms de personnes ou d'organisations peuvent être trouvés grâce à des indices externes que sont leurs contextes d'apparitions dans les textes : *M. Bush* ou *la société Microsoft (M.* et *société* indiquent respectivement un nom de personne et un nom de société) et grâce à des indices internes (comme un prénom pour un nom de personne). Ce découpage en indices externes et internes est présenté par (MacDonald, 1993).

Les journaux sont intéressants pour ce travail autour des noms propres car ils contiennent une très grande quantité de noms propres. (Coates-Stephens, 1993) précise que les noms propres représentent 10% des textes des journaux en anglais. En étudiant le journal *Le Monde* et le journal *Ouest France*, nous concluons que les noms propres représentent de 8 à 11% des textes journalistiques en français.

Nous avons limité notre étude aux noms propres les plus nombreux : noms de personnes, de lieux et d'organisations. Ces trois types de noms propres ne sont pas égaux les uns par rapport aux autres :

- Leur repérage est plus ou moins difficile selon leur type : les noms de personnes ont souvent un contexte d'apparition qui permet de les trouver et de les catégoriser comme personnes (Ex : *le* ***roi*** *Hassan II*, *le* ***juge*** *Van Ruymbeke*). Les noms de lieux sont rarement accompagnés d'un tel contexte et sont trouvés grâce à des dictionnaires. Les noms d'organisation sont souvent trouvés grâce à des mots clefs qu'ils contiennent et à quelques indices externes (Ex : ***Organisation*** *Mondiale de la Santé*).
- Leur quantité dans les articles de journaux dépend de leur type : une étude que nous avons réalisée sur 130 articles de journaux montre que les noms propres les plus nombreux sont les noms de lieux (43,9%), suivis de près par les noms de personnes (39,8%), puis par les noms d'organisations (16,3%).

L'extraction et la catégorisation des noms propres vont être réalisées par des outils prenant en compte le contexte lexical ainsi que leur structure interne. Dans la partie suivante, nous expliquons le fonctionnement de notre système d'extraction de noms propres.

2.2. Outil d'extraction des noms propres

Il y a deux méthodes principales pour extraire les noms propres :

- La description de règles utilisant des indices linguistiques, surtout lexicaux (contextes, présence d'un indice interne).
- L'apprentissage.

(Coates-Stephens, 1993) décrit le système d'extraction de noms propres *Funès*. Ce système utilise des règles très complètes prenant en compte la morphologie, la syntaxe et la sémantique des noms propres. Comme Coates-Stephens, nous avons créé un outil d'extraction utilisant tous les indices linguistiques possibles.

Notre outil utilise des transducteurs à nombre fini d'états[2] : ils permettent de décrire de manière facile et robuste les noms propres et leurs contextes (Silberztein, 2000). En effet, les transducteurs sont des automates particuliers qui ont un alphabet d'entrée et un alphabet de sortie. L'alphabet d'entrée décrit les motifs que nous voulons localiser dans les textes : les noms propres. L'alphabet de sortie contient des informations à insérer concernant le type du nom propre localisé.

[2] Les automates à états finis et les transducteurs sont de plus en plus utilisés dans le traitement automatique des langues (Roche et al., 1997). L'utilisation de ces transducteurs présente l'avantage d'une très grande robustesse, précision et rapidité.

Notre outil d'extraction (Friburger et al., 2001), permet de passer une cascade de transducteurs (Abney, 1996) sur le texte en deux étapes :

– La première étape extrait les noms propres grâce à une série de règles (cascade de transducteurs) et utilise des dictionnaires de prénoms, lieux et noms de métiers. Notre première phase d'extraction préfère une très grande précision plutôt qu'un très grand rappel.

– La seconde étape utilise les noms propres trouvés dans la première étape comme s'il s'agissait d'un dictionnaire. Ce système permet de trouver une grande quantité de noms propres qu'aucun indice dans le texte ne permet de repérer. Cette seconde étape a un inconvénient : les erreurs d'identification de la première étape se répercutent sur elle.

Ce système extrait environ 91% de noms propres avec une précision de 97%. La quantité de noms propres extraits semble suffisante pour pouvoir réaliser des calculs de similarités à base de noms propres.

3. Similarités basées sur les noms propres

Pour calculer leur similarité, les techniques standards représentent les textes comme des vecteurs de termes (les termes étant des mots ou groupes de mots du texte). (Voorhees, 1999) précise que les systèmes de recherche d'information courants consistent en deux traitements principaux :

- *Indexation* : le texte est découpé en unités lexicales (tokens). Les mots vides (stopwords) sont éliminés pour ne conserver que les mots potentiellement intéressants. Les mots restants sont lemmatisés ou suffixés (stemming).

- *Matching* : Une mesure de similarité entre textes est calculée.

Notre système suit ce schéma : les noms propres sont extraits et les mots restants du texte sont lemmatisés, puis nous représentons chaque texte par un vecteur composé de deux sous-vecteurs différents (Salton et al., 1983) :

- Le premier sous-vecteur est composé soit de tous les mots du texte, soit des mots du texte qui ne sont pas des noms propres. Ces mots sont lemmatisés à l'aide des dictionnaires très complets du système Intex (Courtois et al., 1990).

- Le second sous-vecteur est composé des noms propres du texte catégorisés selon leur type (lieux, personnes, organisations).

A partir de ces deux sous-vecteurs, il est possible de calculer une similarité entre les textes. La similarité du sous-vecteur de mots (resp. du sous-vecteur de noms propres) est appelée sim_{mots} (resp. sim_{np}). La fusion des similarités des deux sous-

vecteurs, donnée par la formule [1], permet de pondérer les similarités des deux sous-vecteurs en discernant les noms propres des autres mots.

$$sim(d_i, d_j) = \alpha.sim_{mots}(d_i, d_j) + \beta.sim_{np}(d_i, d_j) \quad [1]$$

où α et β sont les pondérations des deux sous vecteurs.

3.1. Mesure de la similarité de deux sous-vecteurs

Etant donné une collection de textes C, chaque texte i est représenté par un vecteur de termes Di. Les termes présents dans ce texte i sont définis par le poids w_{ij} de chacun de ces termes [2].

$$D_i = (w_{i1}, w_{i2}, \ldots, w_{ij}, \ldots, w_{it}) \quad [2]$$

Pour vérifier que les noms propres peuvent améliorer la qualité des mesures de similarité, nous avons choisi la mesure TF.IDF très utilisée et reconnue, décrite dans (Salton et al., 1988). Nous avons aussi testé la mesure de Jaccard qui nous semblait intéressante pour des raisons évoquées plus loin. Jaccard est simplement le nombre de mots communs de deux textes d_i et d_j divisé par le nombre de mots de leur union [3].

$$\frac{|d_i \cap d_j|}{|d_i \cup d_j|} \quad [3]$$

La mesure TF.IDF est composée de deux parties : TF_{ik} (Term Frequency) est la fréquence du terme T_k dans le texte d_i, tandis que IDF_k (Inverse Term Frequency) du terme T_k dans la collection C est décrite par :

$$idf_k = \log\left(\frac{N}{n_k}\right) \quad [4]$$

où N est le nombre de textes dans la collection C et n_k est le nombre de documents contenant au moins une occurrence du terme t_k.

Le poids w_{ik} d'un terme est normalisé comme suit [5] :

$$w_{ik} = \frac{tf_{ik} \cdot idf_k}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 \cdot (idf_k)^2}} \quad [5]$$

Finalement, la similarité entre deux textes est donnée par la formule [6] :

$$sim(d_i, d_j) = \sum_{k=1}^{t} w_{ik} \cdot w_{jk} \quad [6]$$

La similarité *sim(di, dj)* varie entre 0 et 1.

3.2. Connaissances pragmatiques liées aux noms propres

Nous avons utilisé deux heuristiques spéciales pour calculer les poids des noms propres : l'une concerne les noms de personnes, l'autre les noms d'organisations.

Notre système d'extraction permet d'extraire les noms de personnes en repérant le prénom si celui si est précisé. Dans un article de journal, les personnes sont le plus souvent nommées au moins une fois avec leur prénom, pour ne plus être précisé dans la suite de l'article. Nous considérons que dans un même article, pour un même nom de personne, si le prénom n'est plus précisé il s'agit de la même personne : un journaliste ne citerait pas deux personnes différentes dans le même article sans les distinguer par leur prénom. De plus, si deux personnes ayant le même nom de famille mais ayant deux prénoms différents sont citées dans un même texte (ex: *Jacques Chirac* et *Bernadette Chirac*), leur fréquence est partagée pour la raison suivante : si dans le texte où *Jacques Chirac* et *Bernadette Chirac* sont cités ensemble, il se trouve une référence à un *président Chirac*, on ne peut savoir, sans connaissances pragmatiques, si ce *président Chirac* réfère *Jacques* ou *Bernadette*.

Entre deux textes, les noms de personnes sont comparés en tenant compte des heuristiques décrites ci-dessus. Lorsqu'on compare les noms de personnes pris dans deux textes dont on est en train de calculer la similarité, on compare les noms de famille puis les prénoms (s'ils existent) si les noms de sont égaux. Si les prénoms sont différents, ce n'est pas de la même personne dont on parle et le poids de son nom entre les deux textes est égal à 0.

La seconde heuristique est appliquée aux noms d'organisations. Notre système d'extraction reconnaît les noms d'organisations et leur forme abrégée si elle est précisée. Par exemple, le terme *Organisation des Nations Unies* est équivalent à

ONU : les deux formes sont synonymes et sont reconnues l'une pour l'autre. Quand deux textes contiennent des noms d'organisations synonymes, les poids sont calculés en conséquence.

Nous avons testé les résultats de cette heuristique en calculant une mesure de similarité avec des noms propres seuls et nous avons comparé avec les résultats obtenus sans cette heuristique. Le résultat est mitigé : en fait, avec heuristique ou sans, les résultats de classification sont les mêmes sur des jeux d'essais classiques. Par contre, nous avons obtenu de meilleurs résultats sur des jeux d'essais "piégés" (c'est-à-dire composés d'articles portant sur des personnes homonymes) en utilisant nos heuristiques.

4. Résultats

4.1. Corpus de tests

Le premier problème de notre évaluation est d'avoir un corpus avec une classification connue. Nous avons pu créer un tel corpus en utilisant l'hypothèse des groupes ("cluster hypothesis") formulée par (van Rijsbergen, 1979). Cette hypothèse dit que les documents proches tendent à être pertinents pour la même requête.

AMARYLLIS est un programme français de recherche d'information équivalent à TREC : chaque campagne d'évaluation propose un certain nombre de requêtes et des corpus de textes. Les systèmes participant aux campagnes d'évaluations doivent trouver les textes du corpus qui répondent aux requêtes proposées. Afin d'évaluer les résultats des différents systèmes, les réponses de chaque requête sont construites par des experts humains ; les résultats proposés par les systèmes participants sont utilisés ensuite pour réviser les réponses construites à la main.

En utilisant l'hypothèse de van Rijsbergen, nous avons construit des groupes (i.e. classes) avec les résultats pertinents aux requêtes d'Amaryllis sur le corpus OFIL1 et OFIL2 (articles du journal *Le Monde*). Nous avons mélangé au hasard ces groupes pour obtenir 7 jeux de textes à classer, comprenant chacun 200 textes environs ; nous avons aussi créé quelques jeux d’essais ne contenant que 30 à 40 textes.

4.2. Méthode d'évaluation des résultats

La classification automatique est réalisée avec l’algorithme de classification ascendante hiérarchique (CAH). Les textes les plus proches sont dans les mêmes branches de l’arbre. Cette méthode de classification très robuste a fait ses preuves depuis longtemps et fournit en général une sortie de très grande qualité (Voorhees, 1986), (Willet, 1988).

Le principe de la classification hiérarchique ascendante est de regrouper à chaque itération les deux textes qui sont les plus proches. Différents critères de rapprochement des textes existent : le critère Average Link ou la méthode de Ward semblent les meilleurs pour donner une hiérarchie de sujet de bonne qualité. Cependant, nous avons choisi d'utiliser le critère du lien maximal ou Complete Link pour fusionner les groupes entre eux. Ce critère crée des groupes très étroits et limités (voir Figure 1), assez compacts et faciles à distinguer (Zamir et al., 1997).

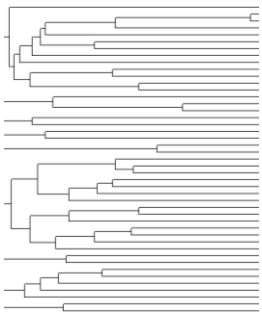

Figure 1: *Arbre obtenu avec le critère du lien maximal*

Le critère Complete Link est le suivant : la similarité d'un nouveau groupe est égale à la similarité des deux membres les moins similaires de ce groupe. Etant donné deux groupes (clusters) c_i et $c_{j,}$, et x, y deux textes, la similarité de ces deux groupes selon la méthode Complete Link est décrite par la formule [7].

$$sim(c_i,c_j) = \min_{x\in c_i, y\in c_j} sim(x,y) \qquad [7]$$

Cette méthode est connue pour être très performante, mais la plus chère en temps, ce qui n'est pas grave puisque, ici, il s'agit juste de vérifier les résultats de notre mesure de similarité.

Pour faire nos calculs, il faut découper l'arbre hiérarchique en groupes (ou clusters); en général, on coupe un arbre en choisissant un seuil de similarité ou un nombre de classes. Nous préférons choisir la deuxième solution qui va nous permettre d'évaluer la qualité des classifications pour un même nombre de classes. Les arbres obtenus par cette classification sur le critère maximal permettent d'obtenir des groupes très étroits sans liens les uns avec les autres : il s'agit en fait d'une forêt d'arbre. Nous entendons par-là que l'arbre hiérarchique est constitué de plusieurs arbres déconnectés les uns des autres car les similarités entre ces arbres

valent 0 lorsqu'on se rapproche de la racine : il y a en fait plusieurs racines qui sont chacune un groupe de textes (voir Figure 1).

Lorsque nous comparons les classifications obtenues, nous le faisons à nombre de classes égales mais ce nombre de classes peut varier suivant les groupes de textes et suivant le type de similarité utilisé. Par exemple, les similarités utilisant les noms propres créent énormément de classes car de nombreux textes ont des similarités qui valent 0 (il y a beaucoup moins de noms propres par textes que de mots, c'est ce qui explique ce phénomène). Pour environ 200 textes classés par les noms propres, notre classification obtient une quarantaine d'arbres déconnectés alors qu'avec tous les mots, on en obtient une dizaine. Nous comparerons donc les résultats des différentes similarités en fonction du nombre d'arbres obtenus par la similarité dont le nombre d'arbres est maximum. Le nombre d'arbres est en fait le nombre de groupes formés par la classification automatique[3].

4.3. Mesure de qualité

Nous avons choisi de mesurer la qualité de nos groupes en mesurant l'entropie de l'ensemble des groupes. L'entropie est une mesure du désordre empruntée aux physiciens. Nous utilisons la mesure de l'entropie d'une classification proposée par (Strehl et al., 2000), légèrement différente de celle proposée initialement par (Boley, 1998) : l'entropie d'un cluster, selon Strehl, est pondérée par le nombre k de classes existantes dans la classification de référence. Chaque texte est étiqueté avec le numéro de la classe de référence à laquelle il appartient. Nous comparons cette référence aux résultats que nous obtenons. L'entropie e_c d'un groupe de textes c (k est le nombre de classes de références) est calculée par la formule [8].

$$e_c = -\frac{1}{\log k} \sum_i \left(\frac{c(i,c)}{\sum_i c(i,c)} \log \left(\frac{c(i,c)}{\sum_i c(i,c)} \right) \right) \qquad [8]$$

c(i,c) est le nombre d'occurrence de la classe i dans le cluster c. L'entropie e_c est nulle quand tous les textes appartenant à c appartiennent à la même classe de référence, sinon l'entropie varie est supérieure à 0 ($0 \leq e_c \leq 1$). L'entropie grandit avec le désordre des textes.

L'entropie totale e_T de l'ensemble des groupes de textes est donnée par [9].

$$e_T = \frac{1}{m} \sum_c e_c \cdot n_c \qquad [9]$$

m est le nombre total de groupes et n_c le nombre de textes dans le groupe c.

[3] Nous avons remarquer que la méthode Complete Link donne un nombre d'arbres (ou groupes) supérieur à la classification de référence.

4.4. Evaluation des différentes similarités

Tout d'abord, la mesure sim_{tsmots_TFIDF} est calculée sur tous les mots du texte (sans discriminer les noms propres des autres mots) avec la mesure TF.IDF. Cette mesure servira de repère par rapport aux résultats obtenus avec des similarités mettant en valeur les noms propres. Parmi les 10 mots ayant le plus grand poids avec cette mesure dans chaque texte, il y a 3 ou 4 noms propres. Ceci est une première preuve de l'importance des noms propres par rapport aux autres mots du texte.

Nous rappelons que plus l'entropie est basse, meilleure est la classification.

Nos essais montrent que les noms propres seuls fournissent une bonne classification ; les mots d'un texte (sans les noms propres) donnent des résultats équivalents à ceux des noms propres. Les résultats de la classification avec les noms propres seuls ou les mots seuls sont nettement moins bons que la classification avec tous les mots du texte.

OFIL1 - jeux d'essais	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Nombre de classes	*12*	*13*	*9*	*11*	*12*	*9*	*12*
Entropie sim $_{tsmots\_TFIDF}$	0.106	0.119	0.086	0.136	0.137	0.231	0.065
Entropie sim $_{tsmots\_Jaccard}$	0.279	0.231	0.090	0.159	0.261	0.211	0.130

Tableau 1 : *Différents résultats d'entropie (Jaccard et TF.IDF) avec tous les mots.*

OFIL1 - jeux d'essais	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Nombre de classes	*40*	*29*	*38*	*34*	*39*	*38*	*28*
Entropie sim $_{tsmots\_TFIDF}$	0.019	0.071	0.025	0.057	0.089	0.087	0.022
Entropie sim $_{np\_TFIDF}$	0.063	0.046	0.087	0.102	0.110	0.293	0.037
Entropie sim $_{np\_Jaccard}$	0.078	0.033	0.042	0.107	0.142	0.259	0.068

Tableau 2 : *Comparaison des entropies sur tous les mots avec celles sur les noms propres seules*.

Nous avons fait des essais avec la mesure TF.IDF et la mesure de Jaccard (Tableau 1) : la mesure TF.IDF donne les meilleurs résultats sur les essais avec tous les mots.

Nous attendions un résultat différent en ne considérant que les noms propres (Tableau 2). Nous pensions en effet que, les noms propres étant des mots très particuliers, une mesure ne comptant que les noms propres communs telle que Jaccard (sim $_{np\_Jaccard}$) obtiendrait certainement des résultats équivalents à la mesure TF.IDF (sim $_{np\_TFIDF}$) basée sur les fréquences. Nos expériences le confirment et

montrent même que sur plusieurs essais l'entropie est légèrement plus basse avec Jaccard qu'avec TF.IDF sur les noms propres. Par contre, la mesure de similarité qui utilise tous les mots est clairement meilleure que celles sur les noms propres. Nous pouvons en conclure que les mots communs sont nécessaires pour trouver une bonne classification.

Nous avons ensuite fait varier les coefficients α et β de notre fusion de similarité: le coefficient α est appliqué à une similarité sur les mots seuls ou sur tous les mots du texte (selon le cas), et nous faisons varier le coefficient β sur les noms propres.

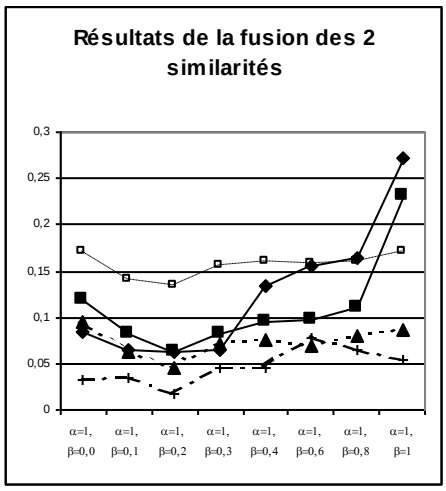

Figure 2 : *Fusion de similarités avec variation des coefficients α et β*

Lorsqu'on applique notre fusion de similarité sur les mots (sans les noms propres) et les noms propres, nous n'arrivons pas, en faisant varier α et β, à obtenir une meilleure classification que celle réalisée avec tous les mots sans discrimination (sim_{tsmots_TFIDFl}).

Lorsque nous testons une fusion de similarité avec tous les mots du texte (α=1) et les noms propres, les résultats sont meilleurs (Figure 2) avec le coefficient β=0.2 que la classification de base obtenue avec tous les mots (sim_{tsmots_TFIDF} avec α=1 et β=0).

Nous avons aussi réalisé des tests sur des corpus assez petits (30-40 textes). Il apparaît que plus le nombre de textes est petit, plus les noms propres donnent d'excellents résultats à nombre de classes égales : ils classent mieux que tous les mots, et la mesure de Jaccard donne de meilleurs résultats que la mesure TFIDF.

4.5. Les noms propres comme sujet de textes

A travers ce travail sur les mesures de similarités, une classification hiérarchique précise est fournie à l'utilisateur pour qu'il puisse balayer rapidement l'ensemble des textes aidé par le sujet des textes. Le sujet d'un groupe de textes est représenté par les noms propres et les mots communs aux textes du groupe et ayant les plus grands poids dans ce groupe. Le type des noms propres est utilisé pour proposer une réponse aux questions *qui ?* (personnes et organisations) et *où ?* (lieux) qu'un utilisateur peut se poser vis-à-vis d'un groupe de textes. Ci-dessous nous présentons un exemple de sujet pour un groupe trouvé par notre système :

Qui? Paula Jones, Monica Lewinsky, Bill Clinton
Où ? Maison Blanche, Etats-unis
Autres mots : plainte, rencontre, divorce, employé, débat, administratif, sondage, amour, république, emploi, femme, démocrate ...

Dans cet exemple, les noms propres à eux seuls évoquent de manière claire le sujet des articles contenus dans ce groupe de textes ; les noms communs sont moins parlants. Cette manière de présenter les résultats d'une classification à l'utilisateur à l'aide de sujets à base de noms propres est prometteuse.

5. Conclusion

Les noms propres sont très précieux et doivent être mis en valeur lors du calcul de similarité entre textes. A travers cet article, nous montrons que les noms propres peuvent améliorer cette mesure. Nous avons créé une mesure de similarité qui prend en compte deux sous-vecteurs pondérés : les noms communs d'une part, et les noms propres d'autre part. La meilleure classification est obtenue pour un schéma de pondération de ces deux sous-vecteurs mettant en valeur les noms propres par rapport aux autres mots. L'intuition que la quantité de noms propres et leur qualité informationnelle les rend plus pertinents que les autres mots d'un texte est ainsi confirmée. On ne peut obtenir une bonne classification en utilisant les noms propres seuls, néanmoins les noms propres améliorent les résultats de la classification s'ils sont mis en valeur par la mesure de similarité.

Nous continuons notre étude de l'importance des noms propres. Nos travaux actuels portent sur les différents types de noms propres pour lesquels nous essayons de voir quels sont ceux qui apportent le plus à la classification.

6. Bibliographie

Abney, S., Partial Parsing via Finite-State Cascades, In *Workshop on Robust Parsing, 8th European Summer School in Logic, Language and Information* Ed., Prague, Czech Republic, 1996, p. 8-15.

Chinchor, N.,. Muc-7 Named Entity Task Definition, http://www.itl.nist.gov /iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html#appendices, 1997.

Coates-Stephens, S., The Analysis and Acquisition of Proper Names for the Understanding of Free Text, Kluwer Academic Publishers, Hingham, MA, 1993.

Courtois, B., Silberztein, M., Dictionnaire électronique des mots simples du français, Larousse, Paris, 1990

Friburger, N., Maurel, D., Finite State Transducer Cascade to Extract Proper Nouns in Franch Text, In *2nd Conference on Implementing and Application of Automata* Ed., Pretoria, South Africa, 2001.

Grishman, R., Sundheim, B., *Message Understanding Conference - 6: a brief history*, In Proceedings of the Sixth Message Understanding Conference. Morgan Kaufmann, 1996.

McDonald, D. D., Internal and External Evidence in the Identification and Semantic Categorisation of Proper Names, In *Corpus processing for lexical acquisition* Ed., 1993, pp.32-43.

Mani I.; MacMillan R. T., *Identifying Unknown Proper Names in Newswire Text*, In Corpus Processing for Lexical Acquisition, MIT Press. Cambridge, MA, 1996, p. 41-59.

Maurel, D., Piton, O., *Un dictionnaire de noms propres pour Intex : Les noms propres géographiques*, Linguisticae Investigationes, (XXII), 1999, p. 277-287.

Mikheev, A., Moens, M., Grover, C., *Named Entity Recognition without Gazetteers*, In *EACL'99* Ed., 1999.

Roche, E., Schabes, Y., *Finite State Language Processing*, MIT Press, Cambridge, Massachussets, 1997.

Salton G., and Buckley C., "*Term-Weighting Approaches in Automatic Text Retrieval*" Information Processing &Management, Vol. 24(5), 1988, p. 513-523.

Salton G., Fox E.A. and H. Wu, *Extended boolean information retrieval*. Commun. ACM , Vol. 26(12), 1983, p. 1022-1036.

Silberztein, M., *INTEX: an FST toolbox*, Theoretical Computer Science, Vol. 234, 2000, p. 33-46.

Strehl, A., Ghosh, J., Mooney, R.: *Impact of similarity measures on web-page clustering*. In Proc. AAAI Workshop on AI for Web Search, 2000, p. 58-64

van Rijsbergen, C.J., *Information Retrieval* (2nd edition), Butterworths, London, 1979.

Voorhees, E.M., *Implementing agglomerative hierarchical clustering algorithms for use in document retrieval*. Information Processing and Management, 22, 1986, p. 465-476.

Voorhees, E.M., "*Natural Language Processing and Information Retrieval*," in Pazienza, MT (ed.), Information Extraction: Towards Scalable, Adaptable Systems, New York: Springer, 1999, p. 32-48.

Willett, P., Recent trends in hierarchic document clustering: a critical review. In Information Processing and Management. 24(5), 1988, p. 577-597.

Zamir, O. Etzioni, O., Madani, O. and R. M. Karp, *Fast and intuitive clustering of Web documents*. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, 1997, p. 287-290.