Actes du huitième défi fouille de texte, DEFT2012, Grenoble, France, 8 juin 2012. Pages 53-64.
Proceedings of the Eight DEFT Workshop, DEFT2012, Grenoble, France, 8th July 2012. Pages 53-64.

Participation de l'IRISA à DeFT2012 : recherche d'information et apprentissage pour la génération de mots-clés

Vincent Claveau, Christian Raymond
IRISA-CNRS IRISA-INSA
Campus de Beaulieu, 35042 Rennes, France
vincent.claveau@irisa.fr
christian.raymond@irisa.fr

RÉSUMÉ

Dans cet article, nous décrivons notre participation au Défi Fouille de Texte (DeFT) 2012. Ce défi consistait en l'attribution automatique de mots-clés à des articles scientifiques en français, selon deux pistes pour lesquelles nous avons employé des approches différentes. Pour la première piste, une liste de mots-clés était fournie. Nous avons donc abordé ce problème comme une tâche de recherche d'information dans laquelle les mots-clés sont les requêtes. Cette approche a donné d'excellents résultats. Pour la seconde piste, seuls les articles étant fournis, nous avons employé une approche s'appuyant sur un extracteur de terme et une réordonnancement par apprentissage.

ABSTRACT

IRISA participation to DeFT 2012 : information retrieval and machine learning for keyword generation

This paper describes the IRISA participation to the DeFT 2012 text-mining challenge. It consisted in the automatic attribution or generation of keywords to scientific journal articles. Two tasks were proposed which led us to test two different strategies. For the first task, a list of keywords was provided. Based on that, our first strategy is to consider that as an Information Retrieval problem in wich the keyword are the queries, which are attributed to the best ranked documents. This approach yielded very good results. For the second task, only the articles were known; for this task, our approach is chiefly based on a term extraction system whose results are reordered by machine learning.

MOTS-CLÉS : Génération de mots-clés, Extraction de termes, Recherche d'information, *Boosting*, arbres de décision, TermoStat.

KEYWORDS: Keyword generation, Term extraction, Information Retrieval, Boosting, Decision tree, TermoStat.

1 Introduction

Dans cet article, nous décrivons notre participation au Défi Fouille de Texte (DeFT) 2012[1]. Ce défi consistait en l'attribution automatique de mots-clés à des articles scientifiques en français, selon deux pistes pour lesquelles nous avons employé des approches différentes. Pour la première piste, une liste de mots-clés était fournie. Nous avons donc abordé ce problème comme une tâche de recherche d'information dans laquelle les mots-clés sont les requêtes. Cette approche a donné d'excellents résultats. Pour la seconde piste, seuls les articles étant fournis, nous avons employé une approche s'appuyant sur un extracteur de terme et un réordonnancement par apprentissage.

La suite de l'article est structurée en trois parties. Nous décrivons tout d'abord brièvement le système d'extraction de termes et les nécessaires prétraitements que nous avons utilisés pour les deux pistes. La section 3 détaille ensuite l'approche que nous avons adoptée pour la piste 1, et les résultats que nous y avons obtenu. Notre contribution pour la piste 2 est quant à elle présentée dans la section 4. Nous terminons enfin par quelques remarques et conclusions sur le défi et les résultats obtenus.

2 Prétraitements et extraction terminologique

2.1 Pré-traitements

Les articles étaient fournis encodés en UTF8 et formaté sous un format XML structurant l'article en un résumé et en paragraphes. Beaucoup de ces articles portant sur la traduction, la linguistique, ou l'ethnologie, ceux-ci contiennent des exemples, phrases et parfois paragraphes complets en langue autre que le français (anglais, grec, inuktitut...). Ces extraits pouvant fausser les processus suivants, il a été nécessaire de les prétraiter. Dans certains cas, pour les plus longs de ces extraits, ils ont été traduits automatiquement par Google Translate quand cela était possible. Dans les autres cas, ils ont été simplement supprimés. Certaines formules mathématiques, notations particulières ou caractères spéciaux (insécables, puces...) ont été aussi supprimés. Les textes ainsi nettoyés peuvent alors être traités par les étapes décrites ci-après.

2.2 Extraction de termes par TermoStat

Aussi bien pour la piste 1 que la piste 2, nous utilisons un extracteur de termes. Ces outils ont pour but de détecter, extraire et normaliser les termes dans des textes de spécialité. Ces termes sont dits soit simples (composés d'un seul mot-forme) ou complexes (plusieurs mots-formes). Deux approches sont usuellement employées : symbolique ou numérique. L'approche symbolique repose sur des patrons morpho-syntaxiques, et est particulièrement utilisée pour extraire des termes complexes. L'approche numérique se base sur les fréquences d'apparition des termes pour décider s'ils sont particuliers au domaine ou non. Ces deux

[1]Ce travail a été en partie effectué dans le cadre du projet Quaero, financé par l'agence pour l'innovation française OSEO.

approches sont habituellement utilisées en conjonction au sein des outils d'extraction les plus connus, dans un ordre variable selon les outils.

Pour ce défi, nous avons utilisé TermoStat (Drouin, 2003), développé par Patrick Drouin à l'OLST, Université de Montréal. Il est librement accessible à `http://olst.ling.umontreal.ca/~drouinp/termostat_web/`. Il appartient au groupe de techniques enchaînant une extraction basée sur des patrons morpho-syntaxiques et un filtrage numérique. Sa particularité réside dans ce dernier traitement : TermoStat compare les fréquences d'apparition des candidats-termes dans le texte spécialisé avec celles d'un très gros corpus généraliste. Cela lui permet de mettre au jour des usage spécifique au texte étudié, aussi bien pour les termes simples que les termes complexes. Le corpus généraliste français est d'environ 28 500 000 occurrences, correspondant à approximativement 560 000 formes différentes. Il est composé d'articles de journaux portant sur des sujets variés tirés du quotidien français Le Monde et publiés en 2002.

TermoStat fonctionne en trois étapes. Le texte est tout d'abord lemmatisé et étiqueté en parties-du-discours à l'aide TreeTagger (Schmid, 1997). Cette première étape permet ainsi à TermoStat d'appliquer une série d'expressions régulières prédéfinies pour extraire les mots ou les ensembles de mots pouvant être des termes. Voici quelques uns de ces patrons syntaxiques tels que donnés dans la notice de TermoStat :
Nom : *définition, dictionnaire*
Nom + adj : *champ sémantique, définition lexicale*
Nom+ prep + nom : *partie du discours, dictionnaire de langue*
Nom+ prep + nom + adj : *complément de objet direct, principe de compositionalité sémantique*
Nom + part pass : *variation liée, langue écrite*
Nom + adj + prep + nom : *structuration sémantique du lexique, approche sémiotique du langage*
Adj : *lexical, syntagmatique*
Adv : *paradigmatiquement, syntagmatiquement*
Verbe : *désambigüiser, lexicaliser*

La dernière étape calcule un score et sélectionne les candidats-termes extraits avec les patrons à l'étape précédente. C'est ce score qui compare les fréquences dans le texte considéré et dans le corpus généraliste. Plusieurs indices sont implémentés dans TermoStat (spécificité, Loglikelihood, χ^2...). Dans notre cas, cet indice a relativement peu d'importance puisqu'il ne sert qu'à limiter la liste des candidats-termes, l'ordre n'étant pas pris en compte (cf. section 3) ou recalculé (voir section 4 pour les résultats avec l'ordonnacement original de TermoStat et avec réordonnancement).

Outre la capacité à extraire les termes simples, Termostat a l'avantage de gérer les phénomènes de variation simples comme la flexion. Les listes de termes obtenues sont finalement filtrées pour ôter quelques candidats erronés dus à quelques erreurs récurrentes de TreeTagger ou à la présence de mots de langues étrangères qui seraient restés dans les textes.

3 Piste 1 : un problème de recherche d'information

Pour cette première piste, une liste contenant tous les mots-clés des articles à traiter était fournie en plus des articles eux-mêmes. Comme nous l'avons expliqué précédemment, nous

avons abordé ce problème d'attribution des mots-clés comme un problème de recherche d'information. Nous décrivons ci-dessous cette approche, et notamment la prise en compte de la morphologie, et les résultats obtenus.

3.1 Principe

Le principe adopté est relativement simple : les mots-clés sont tour à tour considérés comme des requêtes et les articles comme les documents d'une collection. Pour une requête donnée, ces documents sont ordonnés du plus pertinent au moins pertinent à l'aide d'un système de recherche d'information classique qui assigne un score à chaque document. À partir de cet ordonnancement, différentes stratégies peuvent être mises-en-œuvre : le mot-clé considéré peut par exemple être attribué aux n premiers documents retournés, ou à tous les documents obtenant un score supérieur à un certain seuil, ou autre.

Le système de recherche d'information que nous avons implémenté pour cette tâche repose sur des techniques standard du domaine de la RI. Nous avons en particulier utilisé un modèle vectoriel et testé différents types de pondérations. Dans ce type de modèle, chaque document est représenté comme un sac de mots. Les mots outils sont ôtés à l'aide d'un anti-dictionnaire (*stop-list*). Avec une telle représentation, un document contenant la phrase « *le président du parti vote contre la proposition* » sera représenté par { président, parti, proposition, vote }. Il faut noter que la phrase « *le parti du président vote pour la proposition* » obtient la même représentation. Cette déséquencialisation du texte ne permet donc pas de prendre en compte les termes complexes qui permettraient ainsi de distinguer *parti du président* et *président du parti*. Pour les besoins du défi, nous ajoutons donc à cette description classique les termes complexes extraits par TermoStat.

Différentes pondérations utilisées en RI ont été expérimentées. Celles-ci ont toutes pour but de donner plus ou moins d'importance aux termes apparaissant dans les documents selon leur représentativité pour décrire le contenu du document. Cette pondération est un élément essentiel de la qualité des calculs de similarité ; le TF-IDF est l'un des plus anciens (Luhn, 1958; Spärck Jones, 1972). Il est habituellement défini par :

$$w_{TF-IDF}(t,d) = tf(t,d) * \log(N/df(t)) \quad (1)$$

avec tf est le nombre d'occurrence ou la fréquence du terme t dans le document considéré, df sa fréquence documentaire, c'est-à-dire le nombre de documents dans lequel il apparaît, N est le nombre total de documents

Mais le TF-IDF n'est pas le seul choix possible et, de fait, rarement le meilleur (Claveau, 2012). Dans le cadre de ce défi, nous avons principalement utilisé la pondération Okapi-BM25, dont la formule est donnée dans l'équation 2 qui indique le poids du terme t dans le document d ($k_1 = 2$ and $b = 0.75$ sont des constantes, dl la longueur du document, dl_{avg} la longueur moyenne des documents).

$$\begin{aligned} w_{BM25}(t,d) &= TF_{BM25}(t,d) * IDF_{BM25}(t) \\ &= \frac{tf(t,d) * (k_1+1)}{tf(t,d) + k_1 * (1 - b + b * dl(d)/dl_{avg})} * \log \frac{N - df(t) + 0.5}{df(t) + 0.5} \end{aligned} \quad (2)$$

Cette pondération classique peut être interprétée comme une version moderne du TF-IDF.

Il faut noter que les techniques de type LSI, LDA ou vectorisation, permettant d'associer des requêtes et des documents même s'ils n'ont pas de termes en commun sont peu adaptées à notre tâche. En effet, plutôt que de favoriser le rappel, on cherche au contraire à trouver le document contenant la formulation la plus proche de la requête. Pour la même raison, on n'utilise pas de racinisation (*stemming*). Cette technique aveugle de normalisation morphologique est jugée trop agressive pour notre tâche puisqu'elle ne permet plus de distinguer entre *social* et *socialisme* de manière définitive (i.e. quelle que soit la requête). Nous proposons une technique plus fine pour prendre en compte ces variations morphologiques dans la sous-section ci-dessous que s'applique différemment selon la requête

Enfin l'assignation d'un mot-clé peut se faire selon différente stratégie une fois les calculs de RI effectués. Nous en avons testé deux. Dans la première, notée *run1*, nous assignons tous les mots-clés pour lesquels le document est classé premier, sans tenir compte du nombre de mots-clés attendus par article. La deuxième stratégie, notée *run2* dans les résultats ci-dessous, assigne exactement le nombre de mots-clés attendus, en retenant ceux pour lesquels le document considéré est le mieux classé.

3.2 Prise en compte de la variation morphologique

L'approche que nous avons adoptée pour acquérir les variantes morphologiques des mots contenus dans les requêtes s'appuie sur une technique que nous avons développée initialement à des fins terminologiques (Claveau et L'Homme, 2005) puis adaptée au cas de la RI (Moreau *et al.*, 2007). Le principe de cette technique d'acquisition morphologique est relativement simple et s'appuie sur la construction d'analogies. En toute généralité, une analogie peut être représentée formellement par la proposition A : B $\doteq$ C : D, qui signifie « A est à B ce que C est à D » ; c'est-à-dire que le couple A-B est en analogie avec le couple C-D. Son utilisation en morphologie, assez évidente, a déjà fait l'objet de plusieurs travaux (Hathout, 2001; Lepage, 2003) : par exemple, si l'on postule l'analogie *connecteur* : *connecter* $\doteq$ *éditeur* : *éditer*
et si l'on sait par ailleurs que *connecteur* et *connecter* partagent un lien morpho-sémantique, on peut alors supposer qu'il en est de même pour *éditeur* et *éditer*.

Le préalable essentiel à l'utilisation effective de l'apprentissage par analogie est la définition de la notion de similarité qui permet de statuer que deux paires de propositions – dans notre cas deux couples de mots – sont en analogie. La notion de similarité que nous utilisons, notée Sim, est simple mais adaptée aux nombreuses autres langues dans lesquelles la flexion et la dérivation sont principalement obtenues par préfixation et suffixation. Intuitivement, Sim vérifie que, pour passer d'un mot m_3 à un mot m_4, les mêmes opérations de préfixation et de suffixation que pour passer de m_1 à m_2 sont nécessaires. Plus formellement, notons $\mathrm{lcss}(\mathrm{X}, \mathrm{Y})$ la plus longue sous-chaîne commune à deux chaînes de caractères X et Y (*e.g.* $\mathrm{lcss}($*installer*, *désinstallation*$)$ = *install*), et $\mathrm{X} +_{suf} \mathrm{Y}$ (respectivement $+_{pre}$) la concaténation du suffixe (resp., préfixe) Y à X, et $\mathrm{X} -_{suf} \mathrm{Y}$ (respectivement $-_{pre}$) la soustraction du suffixe (resp., préfixe) Y à X. La mesure de similarité Sim est alors définie de la manière suivante :

$$\mathrm{Sim}(m_1\text{-}m_2,\, m_3\text{-}m_4) = 1 \quad \text{si} \quad \begin{cases} m_1 = \mathrm{lcss}(m_1, m_2) +_{pre} \mathrm{Pre}_1 +_{suf} \mathrm{Suf}_1, \text{ et} \\ m_2 = \mathrm{lcss}(m_1, m_2) +_{pre} \mathrm{Pre}_2 +_{suf} \mathrm{Suf}_2, \text{ et} \\ m_3 = \mathrm{lcss}(m_3, m_4) +_{pre} \mathrm{Pre}_1 +_{suf} \mathrm{Suf}_1, \text{ et} \\ m_4 = \mathrm{lcss}(m_3, m_4) +_{pre} \mathrm{Pre}_2 +_{suf} \mathrm{Suf}_2 \end{cases}$$

Sim(m_1-m_2, m_3-m_4) = 0 sinon

où $\mathrm{Pre_i}$ et $\mathrm{Suf_i}$ sont des chaînes de caractères quelconques. Si Sim(m_1-m_2, m_3-m_4) = 1, cela signifie que l'analogie $m_1 : m_2 \doteq m_3 : m_4$ est vérifiée et donc on suppose que la relation morpho-sémantique entre m_1 et m_2 est la même qu'entre m_3 et m_4.

Notre processus de détection de variantes morphologiques consiste ainsi à vérifier, au moyen de la mesure Sim, si un couple de mots inconnus est en analogie avec un ou plusieurs exemples de couples connus. En pratique, pour des raisons d'efficacité lors de la recherche d'analogies, plutôt que les couples-exemples, ce sont les opérations de préfixation et suffixation à l'œuvre dans la mesure de similarité Sim qui sont stockées. Ainsi, le couple-exemple *désinstaller* ↔ *réinstallation* n'est pas stocké en tant que tel, mais on conserve la règle : $m_2 = m_1 -_{pre}$ *dés* $+_{pre}$ *ré* $-_{suf}$ *er* $+_{suf}$ *ation*
Montrer l'analogie *déshydrater* : *réhydratation* $\doteq$ *désinstaller* : *réinstallation* revient alors simplement à tester que *déshydrater* ↔ *réhydratation* vérifie la règle précédente.

La technique de détection de dérivés morphologiques par analogie présentée ci-avant requiert des exemples de couples de mots morphologiquement liés pour pouvoir fonctionner. Cet aspect supervisé n'est pas adapté à une utilisation en RI où l'on souhaite au contraire une totale autonomie du système. Pour répondre à ce problème, nous remplaçons cette phase de supervision humaine par une technique d'amorçage simple permettant de constituer automatiquement un ensemble de paires de mots pouvant servir d'exemples.

Cette première phase de recherche de couples-exemples se déroule de la façon suivante :

1 – choisir un article au hasard dans la collection ;

2 – constituer tous les couples de mots possibles (issus de l'article) ;

3 – ajouter aux exemples les couples m_1-m_2 tels que lcss(m_1,m_2) > l ;

4 – retourner en 1.

Dans les expériences rapportées ci-dessous, ces étapes ont été répétées pour tous les documents à traiter.

Cette phase de constitution d'exemples repose donc sur la même hypothèse que précédemment : la dérivation et la flexion se font principalement par des opérations de préfixation et suffixation. Il n'est pas grave lors de cette phase de ne pas repérer des couples de mots morphologiquement liés ; cependant, pour le bon fonctionnement des analogies qui vont en être tirées, il faut éviter de constituer des couples qui ne seraient pas des exemples valides. Dans notre approche simple, deux précautions sont prises. D'une part, la longueur minimale de la sous-chaîne commune l est fixée à un chiffre assez grand (dans nos expériences, $l = 7$ lettres), ce qui réduit le risque de réunir deux mots ne partageant aucun lien. D'autre part, rechercher les variantes morphologiques au sein d'un même document maximise les chances que les deux mots soient issus d'une même thématique et donc d'un vocabulaire cohérent.

Une fois cette première phase accomplie, il nous est maintenant possible de vérifier si un couple de mots inconnus est en analogie avec une paire connue et de déduire ainsi si les deux mots inconnus sont en relation de dérivation ou de flexion. Dans le cadre de notre application, les mots dont on souhaite récupérer les variantes morphologiques sont ceux constituants les requêtes (les mots-clés). Pour ce faire, chaque mot-forme des requêtes est

confronté à chaque mot de la collection ; si le couple ainsi formé est en analogie avec un des couples-exemples, il est alors utilisé pour l'extension de la requête. En pratique, pour des questions de rapidité, les règles d'analogies sont utilisées de manière génératives : des mots sont produits à partir du terme de la requête en suivant les opérations de préfixation et suffixation indiquées dans les règles et ils sont conservés s'ils apparaissent dans l'index de la collection. L'apprentissage des règles se faisant hors-ligne, seule la recherche des variantes morphologiques des termes des requêtes dans l'index est faite en ligne ; en pratique, dans les expériences reportées ci-après, cela prend quelques dixièmes de seconde.

Ainsi, pour une requête « *pollution des eaux souterraines* », la requête étendue finalement utilisée dans le SRI sera « *pollution des eaux souterraines polluants dépollution anti-pollution pollutions polluées polluent eau souterraine souterrains souterrain* ». Il est important de noter que, lors de l'extension, seuls les mots directement liés aux termes de la requêtes sont ajoutés ; les mots eux-mêmes liés aux extensions ne sont pas pris en compte. Cette absence volontaire de transitivité doit ainsi éviter de propager des erreurs (*vision* → *provision* → *provisions* → *provisionner* → *approvisionner* → *approvisionnement*...).

Enfin, comme nous l'avons déjà expliqué, pour cette application, il est important de privilégier la précision. Si le terme présent dans la requête apparaît dans un ou peu de documents, nous n'utilisons pas d'extensions morphologiques. Nous préférons en effet les documents contenants exactement l'expression utilisée comme mot-clé. En revanche, l'extension morphologique est déclenchée dans deux cas opposés. Si le terme n'apparaît dans aucun document, cette extension de requête permet éventuellement de ramener des documents. Et si le terme apparaît dans beaucoup de documents, l'extension permet de privilégier les documents contenant beaucoup plus le terme et ses variantes. Ce déclenchement de l'extension morphologique des requêtes est donc guidé par l'IDF.

3.3 Résultats

Le tableau 1 présente les résultats selon les mesures d'évaluation définies pour le challenge : précision, rappel et f-mesure[2]. Nous y indiquons les résultats obtenus par notre système utilisant Okapi. À des fins de comparaison, les valeurs obtenues avec le même système et différentes pondérations sont également présentées : TF-IDF, LSI (Dumais, 2004), Hellinger (Escoffier, 1978; Domengès et Volle, 1979).

	Précision (%)	Rappel (%)	F-mesure (%)
TF-IDF	73.86	57.36	64.57
Hellinger	76.25	59.78	67.01
LSI	72.79	56.80	63.81
Okapi *run1*	80.36	64.80	71.75
Okapi sans extension morphologique	81.38	57.67	67.50
Okapi liste *run2*	69.03	69.05	69.04

TABLE 1 – Résultats sur la piste 1 de l'approche par recherche d'information

[2]Ces valeurs, calculées par notre propre programme d'évaluation, diffèrent très légèrement de celles obtenues par les organisateurs.

4 Piste 2 : extraction et réordonnancement de termes

4.1 Principe

L'affectation de mots-clés à un article peut être vu comme un problème de classification binaire. Ainsi, à partir d'une liste de mots-clés candidats potentiels, ce problème d'apprentissage se pose sous la forme suivante : on cherche à apprendre quelles sont les caractéristiques qui font qu'un mot ou un syntagme, extrait d'un document, est ou non un mot-clé de ce document. On dispose de données d'apprentissage : pour un document du jeu d'entraînement donné, chaque mot-clé/syntagme candidat est décrit par un ensemble d'attributs et un label informant si ce candidat est un mot-clé (le label est noté 'CLEF' ci-après) ou non dans ce document (le label est alors 'NON_CLEF').

Un algorithme de classification supervisé peut alors être appliqué sur ces données. Pour chaque document de test, l'ensemble des mots-clés ayant le meilleur score au sens de l'algorithme de classification est conservé. La classifieur que nous avons choisi est bonzaiboost (Raymond, 2010) une implémentation de l'algorithme de boosting AdaBoost.MH (Schapire et Singer, 2000) sur des arbres de décision à un niveau (2 feuilles), les résultats soumis ont été obtenus avec 100 tours de boosting sur la tâche 1 comme la tâche 2.

Notre système a utilisé les attributs suivants :

- la liste de mots-clés candidats est fournie pour la tâche 1. Pour la tâche 2, elle a été produite avec l'utilisation de TermoStat et enrichie avec les noms issus des citations de l'article, les mots dont le suffixe est « isme » ainsi que les noms de pays.
- à chaque mot-clé candidat sont attachés les descripteurs suivants :
  - le patron morpho-syntaxique extrait par TreeTagger (Schmid, 1997)
  - la proportion de paragraphes du document dans lesquels il apparaît
  - sa fréquence dans le document complet (TF)
  - sa fréquence dans le résumé
  - sa fréquence okapi dans le document complet (TF_{BM25})
  - sa fréquence okapi dans le résumé
  - son score IDF (IDF)
  - son score IDF selon okapi (IDF_{BM25})
  - le score TFIDF des mots composants le syntagme (w_{TFIDF})
  - le score okapi des mots composants le syntagme (w_{BM25})

4.2 Résultats

Les résultats obtenus sur la tâche 1 suivant ce principe obtiennent 0.67 (run 3 de la piste 1) de f-mesure ce qui est moins performant que notre approche basé RI mais nous laisse à la seconde position du classement des participants. Sur la tâche 2, la méthode est appliquée

le patron morpho-syntaxique extrait par TreeTagger	17
la proportion de paragraphes du document dans lesquels il apparait	15
la fréquence dans le document complet	5
la fréquence hors-résumé	3
la fréquence dans le résumé	1
la fréquence okapi dans le document complet	13
la fréquence okapi dans le résumé	4
l'IDF	10
l'IDF okapi	5
le score tf*idf des mots composants le syntagme	10
le score okapi des mots composants le syntagme	17

TABLE 2 – Nombre de sélection de chaque descripteur lors de l'apprentissage.

pour réordonner une liste de mots-clés candidats générée par TermoStat. L'utilisation seule de TermoStat obtient un score 0.1699 (run 2 dans l'evaluation officielle) qui augmente à 0,2087 après ré-ordonnancement (run 1). Ce ré-ordonnancement nous permet de nous classer troisième avec peu d'écart avec le second.

Le modèle obtenu pour la tâche 2 est résumé dans les tableaux 2 et 3. Le premier montre le nombre de sélections de chaque descripteur. Le second montre pour les 30 premiers tours de boosting, le test sélectionné par l'arbre de décision à un niveau ainsi que son vote selon si on tombe dans la feuille gauche (test positif) ou droite (test négatif) de l'arbre.

4.3 Discussion

L'approche par classification supervisée donne des résultats convaincants, à la fois sur la tâche 1 et la tâche 2 avec pourtant un ensemble très succinct de descripteurs et aucunes connaissances extérieures au corpus de documents, mis à part le corpus de référence utilisé par TermoStat. Étant donné la difficulté de la tâche, le phénomène de sur-apprentissage se fait vite ressentir et augmenter le nombre de tour de boosting ou/et la complexité de l'arbre de décision diminue le pouvoir de prédiction du classifieur. Il est probable que cette approche ait un potentiel d'amélioration important avec l'ajout de nouveaux descripteurs et de connaissances extérieures au corpus, notamment dans le cas où les mots-clés ne sont pas présents dans le document.

5 Conclusion

Les approches utilisées par notre équipe pour les deux pistes du défi relèvent de deux stratégies différentes. Toutes deux ont néanmoins la particularité d'être des techniques éprouvées, mais c'est leur conjonction qui fait l'originalité de notre contribution. D'autre part, les bons résultats obtenus valident ce choix, effectué cette année encore, d'opter pour ces techniques simples.

L'approche par RI se révèle très efficace mais ne peut s'appliquer que lorsque les mots-clés

Test binaire	Tour	oui	non
TF_general<1.5	1	NON_CLEF :3.49	NON_CLEF :1.94
tfresumeokapi<0.730454	2	NON_CLEF :0.27	CLEF :0.71
patron_pos="NOM "	3	NON_CLEF :0.10	CLEF :0.54
IDF<0.488632	4	NON_CLEF :0.81	CLEF :0.05
tfokapi<2.13327	5	NON_CLEF :0.11	CLEF :0.35
paragraphe_apparition<0.00311962	6	CLEF :0.93	NON_CLEF :0.04
score_syntagme<15.4967	7	NON_CLEF :0.10	CLEF :0.29
patron_pos="NOM VER :pper "	8	NON_CLEF :3.38	CLEF :0.01
patron_pos="nom NOM "	9	NON_CLEF :0.86	CLEF :0.03
patron_pos="NOM NOM "	10	NON_CLEF :1.89	CLEF :0.01
IDF<1.17607	11	NON_CLEF :0.33	CLEF :0.06
patron_pos="PRP "	12	NON_CLEF :1.21	CLEF :0.01
tfresumeokapi<1.2873	13	NON_CLEF :0.04	CLEF :0.46
tfokapi<1.65131	15	NON_CLEF :0.13	CLEF :0.12
patron_pos="nom ADJ "	16	NON_CLEF :2.86	CLEF :0.00
IDF<0.00355873	17	NON_CLEF :2.81	CLEF :0.00
paragraphe_apparition<0.00137276	18	CLEF :1.04	NON_CLEF :0.01
tfokapi<0.879093	19	NON_CLEF :0.81	CLEF :0.02
tfokapi<0.998128	20	CLEF :0.29	NON_CLEF :0.05
score_syntagme<134.614	21	NON_CLEF :0.01	CLEF :0.52
patron_pos="VER :pper "	22	NON_CLEF :0.69	CLEF :0.01
score_syntagme_okapi<-10.8857	23	CLEF :0.05	NON_CLEF :0.12
score_syntagme<33.7732	24	NON_CLEF :0.03	CLEF :0.22
score_syntagme_okapi<10.7913	25	CLEF :0.01	NON_CLEF :0.49
IDF<3.24816	26	NON_CLEF :0.09	CLEF :0.06
paragraphe_apparition<0.050569	27	NON_CLEF :0.08	CLEF :0.08
IDF_OKAPI<4.5372	29	CLEF :0.05	NON_CLEF :0.10
score_syntagme_okapi<6.53596	30	NON_CLEF :0.02	CLEF :0.19

TABLE 3 – Tests sélectionnés durant les 30 premiers tours de *boosting*. Pour chaque tour, pour les cas où le test est positif ou négatif, est marqué le label pour lequel l'algorithme vote ainsi que le poids donné à ce vote.

possibles sont connus (piste 1). Sauf à supposer que les mots-clés soient nécessairement tirés d'une terminologie fixée (comme par exemple le MeSH pour les articles du domaine biomédical), cette tâche ne présente qu'un intérêt limité. L'évaluation qui en est faite ne permet d'ailleurs pas de juger parfaitement un tel type d'application puisque tous les mots-clés des articles à traiter étaient donnés, mais seuls ceux-là. Chaque mot-clé devait donc être attribué à au moins un article. Il aurait pu être intéressant de noyer ces mots-clés parmi d'autres et d'ainsi évaluer la capacité réelle des méthodes à trouver les bons mots-clés et non simplement à trouver les bons appariements.

Les résultats obtenus sur la piste 2 par l'approche par réordonnacement sont bien sûr moins bons, mais la tâche est évidemment bien plus compliquée. Elle correspond de fait à une application qui semble plus réaliste mais dont l'évaluation est aussi plus difficile. En effet, un mot-clé prédit par le système mais non donné par l'auteur n'est pas pour autant un mauvais mot-clé. Les habitudes d'indexation, le contexte de l'article (autres articles des mêmes auteurs, autres articles de la revue...) mais aussi hasard et parfois des choix discutables influent sur le résultat. Il serait à ce titre intéressant d'étudier l'accord inter-annotateur d'humains ayant pour tâche de produire ces mots-clés.

Références

CLAVEAU, V. (2012). Okapi, Vectorisation et calcul de similarité pour le TAL : pour oublier enfin le TF-IDF. *In Actes de la 19ème conférence Tratement Automatique du Langage Naturel, TALN'12*, Grenoble, France.

CLAVEAU, V. et L'HOMME, M.-C. (2005). Structuring terminology by analogy machine learning. *In Proceedings of the International conference on Terminology and Knowledge Engineering, TKE'05*, Copenhague, Danemark.

DOMENGÈS, D. et VOLLE, M. (1979). Analyse factorielle sphérique : une exploration. *Annales de l'INSEE*, 35:3–83.

DROUIN, P. (2003). Term extraction using non-technical corpora as a point of leverage. *Terminology*, 9(1):99–117.

DUMAIS, S. (2004). Latent semantic analysis. *ARIST Review of Information Science and Technology*, 38(4).

ESCOFFIER, B. (1978). Analyse factorielle et distances répondant au principe d'équivalence distributionnelle. *Revue de statistique appliquée*, 26(4):29–37.

HATHOUT, N. (2001). Analogies morpho-synonimiques. une méthode d'acquisition automatique de liens morphologiques à partir d'un dictionnaire de synonymes. *In Actes de la 8e conférence Traitement Automatique du Langage Naturel, TALN'01*, Tours, France.

LEPAGE, Y. (2003). *De l'analogie ; rendant compte de la communication en linguistique*. Thèse d'habilitation (HDR), Université de Grenoble 1, Grenoble, France.

LUHN, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal on Research and Development*, 2(2).

MOREAU, F., CLAVEAU, V. et SÉBILLOT, P. (2007). Automatic morphological query expansion using analogy-based machine learning. *In Proceedings of the European Conference on Information Retrieval, ECIR'07*, Rome, Italie.

RAYMOND, C. (2010). Bonzaiboost. http ://bonzaiboost.gforge.inria.fr/.

SCHAPIRE, R. E. et SINGER, Y. (2000). BoosTexter : A boosting-based system for text categorization. *Machine Learning*, 39:135–168. `http://www.cs.princeton.edu/~schapire/boostexter.html`.

SCHMID, H. (1997). *New Methods in Language Processing, Studies in Computational Linguistics*, chapitre Probabilistic part-of-speech tagging using decision trees, pages 154–164. UCL Press, London. `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`.

SPÄRCK JONES, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1).