DEFT2011

Actes du septième DÉfi Fouille de Textes

Proceedings of the Seventh DEFT Workshop

1er juillet 2011

Montpellier, France

DEFT2011

Actes du septième DÉfi Fouille de Textes

1er juillet 2011
Montpellier, France

Préface

Le défi fouille de textes (DEFT) a été créé en 2005 par une équipe de chercheurs et doctorants du Laboratoire de Recherche en Informatique (LRI – CNRS/Université Paris Sud) : Jérôme Azé, Mathieu Roche, et Violaine Prince. La création du défi a notamment été inspirée par les campagnes d'évaluation internationales telles que TREC.

En 2007, l'organisation du défi a été reprise par le Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI–CNRS/Université Paris Sud/UPMC) : Martine Hurault-Plantet, Michèle Jardino, Benoît Habert, Cyril Grouin, Patrick Paroubek, et Pierre Zweigenbaum.

Depuis 2010, le défi est co-organisé par le LIMSI et l'École de Bibliothéconomie et des Sciences de l'Information (EBSI – Université de Montréal) : Cyril Grouin, Dominic Forest, Patrick Paroubek et Pierre Zweigenbaum.

L'objectif du défi consiste à proposer chaque année une campagne d'évaluation, en français, sur des thématiques de fouille de texte régulièrement renouvelées :

- 2005 : identification des ruptures de style et de changement de contexte (insertion de phrases de F Mitterrand dans des discours de J Chirac).
- 2006 : identification de la segmentation thématique de textes dans des corpus de discours politiques, de textes juridiques, et d'ouvrages scientifiques.
- 2007 : fouille d'opinion : détection de l'opinion exprimée dans des textes (valence positive, neutre, ou négative) dans des critiques de livres, de films ou de jeux vidéos, dans des relectures d'articles scientifiques et dans des débats parlementaires.
- 2008 : classification de textes selon le genre parmi deux classes (encyclopédie vs. articles de journaux) et la thématique parmi neuf classes (sports, international, national, science, art, économie, littérature, etc.).
- 2009 : fouille d'opinion en corpus multilingue *(français, anglais, italien)* : déterminer le caractère globalement subjectif ou objectif d'un texte en corpus de presse, identifier les passages subjectifs d'un texte en corpus de presse et dans des débats parlementaires, et identifier le parti politique d'appartenance d'un orateur.
- 2010 : étude des variations diachroniques et diatopiques de textes en corpus de presse : identification de la décennie de publication d'un article de presse ancienne (sur une période de 150 ans parmi cinq journaux français) et identification du pays et du journal de parution d'un article de presse contemporaine (parmi quatre journaux français ou québécois).
- 2011 : étude de la variation diachronique et appariements de résumés et d'articles scientifiques : identification de l'année de publication d'un article de presse ancienne (parmi sept journaux français) et appariements de résumés et d'articles scientifiques.

Pour cette nouvelle édition 2011, nous avons donc proposé aux participants de travailler sur deux tâches distinctes.

La première concerne la variation diachronique en corpus de presse – reprenant en celà une thématique déjà abordée en 2010 – et attend des participants qu'ils identifient l'année de publication d'extraits de 500 mots (première piste) ou de 300 mots (seconde piste) d'articles de presse, parus entre 1801 et 1944.

La seconde tâche traite du résumé d'articles scientifiques et s'articule autour de l'appariement de résumés et d'articles scientifiques complets (première piste) et de l'appariement de résumés et du texte des articles scientifiques (seconde piste, les articles ont été amputés de leur introduction et conclusion).

Ces deux tâches ont porté sur des textes rédigés en français.

Cyril Grouin et Dominic Forest, *co-présidents du Comité de Programme*

Comités

Comité de programme

Cyril Grouin (LIMSI–CNRS, Orsay), *co-président*

Dominic Forest (EBSI, Université de Montréal), *co-président*

Béatrice Daille (LINA, Nantes)

Mathieu Lafourcade (LIRMM, Montpellier)

Patrick Paroubek (LIMSI–CNRS, Orsay)

Mathieu Roche (LIRMM, Montpellier)

Juan Manuel Torres-Moreno (LIA, Avignon)

Pierre Zweigenbaum (LIMSI–CNRS, Orsay)

Comité d'organisation

Cyril Grouin (LIMSI–CNRS, Orsay)

Dominic Forest (EBSI, Université de Montréal)

Mathieu Lafourcade (LIRMM, Montpellier)

Mathieu Roche (LIRMM, Montpellier)

Comité de relecture

Cyril Grouin (LIMSI–CNRS, Orsay)

Dominic Forest (EBSI, Université de Montréal)

Table des matières

Programme

Vendredi 1er juillet 2011, matin

8.45–9.00 *Accueil des participants*

SESSION I — PRÉSENTATION ET RÉSULTATS

9.00–9.30 Présentation et résultats du défi fouille de texte DEFT2011. *Cyril Grouin, Dominic Forest, Patrick Paroubek et Pierre Zweigenbaum*

SESSION II — TÂCHE 1. DIACHRONIE

9.30–10.00 Participation de l'IRISA à DEFT 2011 : expériences avec des approches d'apprentissage supervisé et non-supervisé. *Christian Raymond et Vincent Claveau*

10.00–10.30 Méthodes pour l'archéologie linguistique : datation par combinaison d'indices temporels. *Anne Garcia-Fernandez, Anne-Laure Ligozat, Marco Dinarelli et Delphine Bernhard*

10.30–10.50 *Pause*

10.50–11.20 Comparaison de méthodes sémantique et asémantique pour la catégorisation automatique de documents. *Romaric Boley*

11.20–11.50 Expérimentations autour des espaces sémantiques hybrides. *Adil El Ghali*

SESSION III — TÂCHE 2. APPARIEMENTS

11.50–12.10 Deft 2011 : Appariement de résumés et d'articles scientifiques fondé sur des distributions de chaînes de caractères. *Gaël Lejeune, Romain Brixtel, Emmanuel Giguet et Nadine Lucas*

12.10–12.30 INAOE at DEFT 2011: Using a Plagiarism Detection Method for Pairing Abstracts-Scientific Papers. *Fernando Sánchez-Vega, Esaú Villatoro-Tello, Antonio Juárez-González, Luis Villaseñor-Pineda, Manuel Montes-y-Gómez et Luis Meneses-Lerín*

12.30–14.00 *Repas*

Vendredi 1er juillet 2011, après-midi

SESSION III — TÂCHE 2. APPARIEMENTS (suite)

14.00–14.20 Matching documents and summaries using key-concepts. *Sara Tonelli et Emanuele Pianta*

14.20–14.40 Indexer, comparer, apparier des textes et leurs résumés : une exploration. *Martine Cadot, Sylvain Aubin et Alain Lelu*

14.40–15.00 Matching Texts with SUMMA. *Horacio Saggion*

15.00–15.20 LSVMA : au plus deux composants pour apparier des résumés à des articles. *Yves Bestgen*

15.20–15.50 *Pause*

15.50–16.10 Couplage d'espaces sémantiques et de graphes pour le Deft 2011 : une approche automatique non supervisée. *Yann Vigile Hoareau, Murat Ahat, Coralie Petermann et Marc Bui*

16.10–16.30 One simple formula for losing DEFT with more than 90% of correct guesses. *Daniel Devatman Hromada*

SESSION IV — DISCUSSION ET CONCLUSION

16.30–17.30 Discussion sur l'édition 2011 et pistes pour l'édition 2012.

17.30 *Clôture de l'atelier*

Présentation et résultats

Présentation et résultats du défi fouille de texte DEFT2011 Quand un article de presse a t-il été écrit ? À quel article scientifique correspond ce résumé ?

Cyril Grouin[1] Dominic Forest[2] Patrick Paroubek[1] Pierre Zweigenbaum[1]
(1) LIMSI-CNRS, BP133, 91403 Orsay Cedex, France
(2) École de Bibliothéconomie et des Sciences de l'Information, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, H3C 3J7, Canada
{cyril.grouin, patrick.paroubek, pierre.zweigenbaum}@limsi.fr, dominic.forest@umontreal.ca

Résumé. Dans cet article, nous présentons l'édition 2011 du défi fouille de texte (DEFT). Pour cette édition, nous avons proposé deux tâches, l'une portant sur la variation diachronique (faisant suite à la tâche diachronique instituée lors de DEFT2010), la seconde ayant trait aux appariements entre résumés et articles scientifiques. Nous exposons dans un premier temps les tâches proposées, les modalités de constitution des différents corpus ainsi que les tests humains réalisés. Dans un second temps, nous détaillons les résultats obtenus par chacun des participants aux deux tâches. Enfin, nous concluons sur l'édition et abordons quelques pistes pour la prochaine édition du défi.

Abstract. In this paper, we present the DEFT 2011 edition. In this edition, we proposed two tasks, the first one dealing with diachronic variation (being the continuation of the diachronic variation task in DEFT2010), the second one being dedicated to the abstracts/scientific articles pairing. We first describe the proposed tasks, how corpora were created, and human evaluation. We then detail the results obtained by each participant. Finally, we made a conclusion of this edition and propose some ideas for the next challenge.

Mots-clés : Campagne d'évaluation, fouille de texte, diachronie, appariements résumés/articles.

Keywords: Evaluation campaign, Text-mining, Diachronic variation, abstracts/articles pairing.

1 Introduction

Depuis 2005, le défi fouille de texte (DEFT) propose un challenge de recherche en fouille de texte autour de thématiques régulièrement renouvellées. Depuis sa création, l'objectif du défi vise à confronter les méthodes élaborées par plusieurs équipes, sur un même jeu de données, à la manière des campagnes d'évaluation internationales qui existent dans le domaine de la recherche d'information (MUC, TREC, CLEF, etc.).

Pour cette nouvelle édition, deux appels ont été diffusés sur les principales listes de discussion dans le domaine du traitement automatique des langues *(Corpora, Humanist, LN, Risc, etc.)*, les 6 janvier et 16 février 2011. Quatorze équipes se sont inscrites, douze ayant travaillé jusqu'à la phase de test. Parmi ces différentes équipes, nous notons avec satisfaction la participation de trois équipes non francophones (FBK en Italie, INAOE au Mexique et UPF en Espagne) et à des équipes nord-américaines (Canada et Mexique). Comme en 2010, deux équipes ayant la même affiliation que les organisateurs (EBSI et LIMSI) se sont inscrites au défi. Nous précisons qu'elles n'ont bénéficié d'aucun traitement de faveur.

- CHArt, *Cognition Humaine et Artificielle*, Paris, France : Yann-Vigile Hoareau, Murat Ahat, Saïd Fouchal, Coralie Peterman et David Medernach.
- EBSI, *Ecole de Bibliothéconomie et des Sciences de l'Information*, Montréal, Canada : Romaric Boley.
- FBK, *Fondazione Bruno Kessler*, Trento, Italie : Sara Tonelli et Emanuele Pianta.
- GREYC, *Groupe de Recherche en Informatique, Image, Automatique et Instrumentalisation de Caen*, Caen, France : Gaël Lejeune, Romain Brixtel et Emmanuel Giguet.
- INAOE, *Instituto Nacional de Astrofísica, Óptica y Electrónica*, Mexico, Mexique : Fernando Sánchez-Vega, Esaú Villatoro-Tello, Antonio Juárez-González, Luis Villaseñor-Pineda, Manuel Montes-y-Gómez et Luis Meneses-Lerín.
- IRISA, *Institut de Recherche en Informatique et Systèmes Aléatoires*, Rennes, France : Christian Raymond et Vincent Claveau.
- LIMSI, *Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur*, Orsay, France : Anne-García-Fernandez, Anne-Laure Ligozat, Marco Dinarelli et Delphine Bernhard.
- LORIA, *Laboratoire Lorrain de Recherche en Informatique et ses Applications*, Nancy, France — LASELDI, *Laboratoire de Sémiotique, Linguistique, Didactique et Informatique*, Besançon, France — Diatopie, Paris, France : Martine Cadot, Sylvain Aubin et Alain Lelu.
- LUTIN, *Laboratoire des Usages en Technologies d'Information Numérique*, Paris, France. Deux membres de ce laboratoire ont participé individuellement à DEFT2011. Nous notons ces deux participations dans la suite de ce papier LUTIN-a (Adil El Ghali) et LUTIN-d (Daniel Devatman Hromada).
- UCL, *Université Catholique de Louvain*, Louvain-la-Neuve, Belgique : Yves Bestgen.
- UPF, *Universitat Pompeu Fabra*, Barcelone, Espagne : Horacio Saggion.

Nous avons proposé aux participants de travailler sur deux tâches distinctes. La première concerne la variation diachronique en corpus de presse et attend des participants qu'ils identifient l'année de publication d'extraits de 500 mots (piste 1) ou de 300 mots (piste 2) d'articles de presse, parus entre 1801 et 1944. La seconde tâche traite du résumé d'articles scientifiques et s'articule autour de l'appariement de résumés et d'articles scientifiques complets (piste 1) et de l'appariement de résumés et du texte des articles scientifiques (piste 2, les articles ont été amputés de leur introduction et conclusion). Ces deux tâches ont porté sur des textes rédigés en français.

2 Tâche 1. Diachronie

Pour faire suite à la tâche diachronique de l'édition 2010 du défi DEFT (Grouin *et al.*, 2010) pour laquelle des méthodes originales ont été élaborées, nécessitant cependant des améliorations, nous avons décidé de proposer de nouveau une tâche sur la variation diachronique. L'édition 2010 concernait l'identification de la décennie de publication d'un extrait de 300 mots. L'un des points problématiques de ce type de tâche concerne les frontières de classe. En effet, un document paru en 1919 sera rattaché à la décennie 1910 (une décennie couvrant les années de 0 à 9 selon notre définition). Ainsi, un système retournant la décennie 1920 sera pénalisé par cette réponse, alors que l'année de parution se situe à proximité immédiate de cette décennie. Pour pallier ce problème, nous avons décidé de nous focaliser non plus sur les décennies mais sur l'année exacte de parution d'un document.

2.1 Corpus

2.1.1 Présentation générale

Le corpus a été créé à partir des archives de journaux numérisées avec reconnaissance optique de caractères mises à la disposition du public par la BNF via son portail Gallica[1]. L'ensemble des journaux ainsi numérisés a été téléchargé. Chaque fichier dans sa version texte a été découpé en portions de 500 mots et seules les portions vides de tout caractère exotique (le tilde, l'esperluette, l'accent circonflexe sans voyelle, etc.) ont été conservées. Contrairement à DEFT2010, nous n'avons travaillé que sur les pages 1 et 2 de ces journaux, partant du principe que ces pages contenaient davantage d'articles de fond que les pages 3 et 4, plutôt dévolues aux programmes du théâtre, à la bourse ou à des réclames. Nous avons conservé les proportions de l'année dernière en terme de volume pour l'apprentissage et le test. Ainsi, pour DEFT2010, 252 documents/décennie (apprentissage) et 169 documents/décennie (test) ; pour DEFT2011 : 25 documents/année (apprentissage) et 17 documents/année (test).

Deux pistes ont été proposées, l'une sur des extraits de 300 mots (comme lors de DEFT2010 afin de mesurer l'évolution des systèmes ayant participé aux deux éditions), l'autre sur des extraits de 500 mots (nouveauté de DEFT2011, de manière à voir si des améliorations sont notables avec un passage à l'échelle). Les mêmes documents ont été utilisés pour les deux pistes, les documents de 300 mots étant une partie de ceux de 500 mots (extraction aléatoire du début, du milieu ou de la fin), ordonnés différemment dans les corpus des deux pistes. Alors que l'édition 2010 du défi intégrait des extraits provenant de cinq journaux, le portail Gallica s'est entre temps enrichi de deux nouveaux titres disponibles au format texte : *La Presse* et *Le Temps*. Nous avons intégré ces deux nouveaux titres aux corpus de cette année, portant le nombre total de journaux traités à sept.

Les caractéristiques principales de ces corpus et des traitements appliqués sont les suivants :
– Les documents de travail sont le résultat d'une reconnaissance optique de caractères ; ils contiennent donc du bruit lié à cette reconnaissance de caractères (« efTorcée », « rcatisô », « cotte », « ?uf ») ;
– Nous avons éliminé les portions de journaux comportant des caractères inexistants à l'état brut en français (le tilde ˜, l'esperluette &, et le circonflexe isolé ˆ) ;
– Les caractères chevrons < et > sont remplacés par l'entité HTML correspondante (&lt ; et &gt ;) ;
– Le résultat de la reconnaissance optique de caractères ne comprenant aucun élément de structuration, les documents proposés intègrent donc des extraits d'articles incomplets (début et/ou fin manquants) ;
– Les années, lorsqu'elles étaient explicitement présentes et sans erreur dans le texte (« 1813 » par exemple, mais pas « 18 !3 »), ont été remplacées par une balise typante <annee/>.

2.1.2 Corpus d'apprentissage

Le corpus d'apprentissage intègre des articles provenant de 6 journaux différents (*Le Journal des Débats, Le Journal de l'Empire, Le Journal des Débats politiques et littéraires, Le Figaro, Le Temps* et *La Croix*), à raison d'un maximum de 25 documents par année (ce qui correspond aux 249 articles par décennie de DEFT2010). Seule l'année 1815 compte moins de 25 articles, cette année étant charnière entre deux journaux, avec trop peu de documents de qualité selon les critères fixés. La répartition des articles en fonction du journal d'origine est ici donnée (voir figure 1), sachant qu'un journal s'est prolongé dans le temps sous trois noms différents : *Le Journal des Débats* devenu *Le Journal de l'Empire* en 1805 et *Le Journal des Débats politiques et littéraires* en 1814.

2.1.3 Corpus de test

Afin d'éprouver la robustesse des systèmes, le corpus de test intègre des extraits d'articles provenant d'un septième journal absent du corpus d'apprentissage, *La Presse*. Nous avons retenu ce journal comme matériau inconnu pour deux raisons : parce qu'il n'a pas servi dans le corpus de DEFT2010 d'une part[2], et parce que sa parution a commencé plus tôt que pour *Le Temps* – autre journal absent du corpus de DEFT2010 – en 1836 au lieu de 1861. Le corpus de test propose 17 documents par année, sauf pour 1815 pour les mêmes raisons que celles évoquées dans la présentation de la constitution du corpus d'apprentissage.

[1] `http://gallica.bnf.fr/`

[2] En 2010, nous avions utilisé des articles provenant des quotidiens *Le Journal des Débats, Le Journal de l'Empire, Le Journal des Débats politiques et littéraires* et de *La Croix* pour le corpus d'apprentissage ; le corpus de test comprenait en plus des articles du quotidien *Le Figaro*.

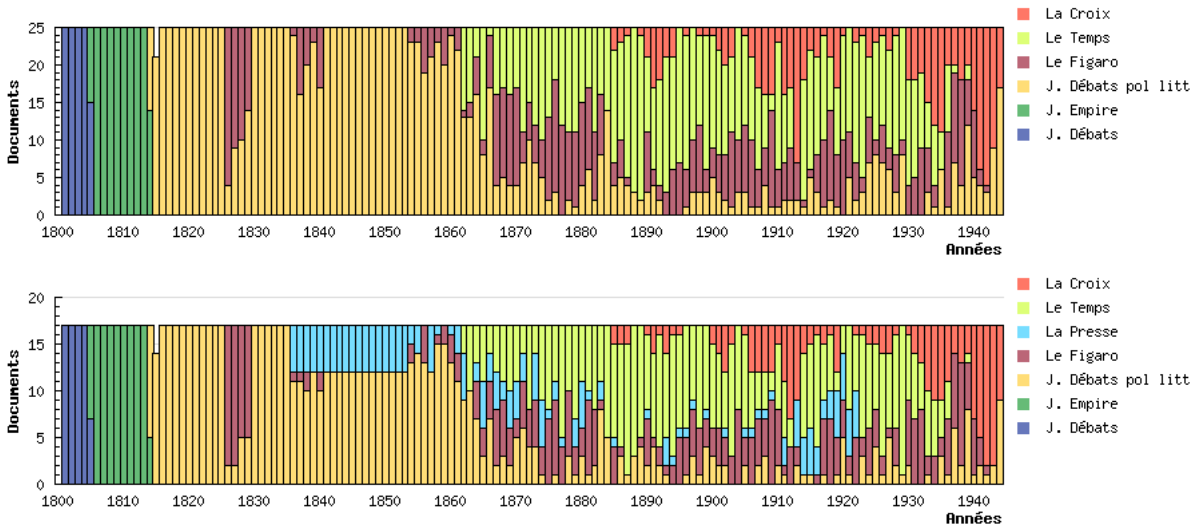


FIG. 1 – Nombre d'articles par journal et par année dans les corpus d'apprentissage et de test

2.2 Méthodes d'évaluation

Puisque l'identification d'une année au lieu d'une décennie multiplie par dix le nombre de classes à traiter d'une part, et pour ne pas faire chuter les résultats d'autre part, nous avons décidé d'évaluer les résultats sur la base d'une fenêtre de 15 ans autour de l'année de référence. Étant donné un fragment d'article a_i dont la date de parution indiquée dans la référence est $d_r(a_i)$, un système prédit une date de parution $d_p(a_i)$. Le système reçoit pour cette tâche un gain qui est d'autant plus grand que l'année prédite est proche de l'année de référence.

– Nous définissons ce gain comme une similarité entre date prédite et date de référence. Il varie entre 0 (pire) et 1 (meilleur). Notation : $s(d_p, d_r)$. Le principe d'une similarité (plutôt que d'une distance) permet de raisonner en termes de masse de notes à donner à un système. Ces notes s'additionnent directement lorsque l'on passe d'un fragment d'article individuel à l'ensemble des fragments à dater.
– La note totale S pour un système est la moyenne des notes s_i obtenues pour chaque fragment d'article a_i des N fragments du corpus de test (1) :

$$S = \frac{1}{N} \sum_{i=1}^{N} s\left(d_p\left(a_i\right), d_r\left(a_i\right)\right) \tag{1}$$

Nous choisisons pour calculer la similarité entre date prédite et date de référence la fonction gaussienne (2) :

$$s_g\left(d_p, d_r\right) = e^{-\frac{\pi}{10^2}\left(d_p - d_r\right)^2} \tag{2}$$

Le maximum de s_g vaut 1 pour $d_p = d_r$. La fonction tend vers 0 lorsque d_p s'éloigne de d_r. Le tableau 1 donne les valeurs de s_g en fonction de la valeur absolue de la différence $d_p - d_r$. L'aire sous la courbe (intégrale) de s_g est égale à 10 : la masse totale de score de tolérance offerte à d_p est la même que celle qui serait produite par un intervalle de tolérance de 10 ans centré sur la date de référence d_r et à l'intérieur duquel le score de d_p vaudrait 1 (configuration de DEFT 2010). La fonction s_g remplace le score de similarité binaire de DEFT 2010 (1 si on est dans ces 10 ans, 0 sinon) par une décroissance plus graduelle.

C'est cette fonction de similarité s_g, moyennée sur l'ensemble des N fragments d'articles du corpus (formule 1, précisée en 3), qui a été utilisée pour calculer le score officiel d'un système p dans cette tâche :

$$S\left(p\right) = \frac{1}{N} \sum_{i=1}^{N} e^{-\frac{\pi}{10^2}\left(d_p(a_i) - d_r(a_i)\right)^2} \tag{3}$$

$\lvert d_p - d_r \rvert$	0	1	2	3	4	5	6	7	8
$s_g(d_p, d_r)$	1,000	0,969	0,882	0,754	0,605	0,456	0,323	0,215	0,134
$\lvert d_p - d_r \rvert$	9	10	11	12	13	14	15	> 15	
$s_g(d_p, d_r)$	0,078	0,043	0,022	0,011	0,005	0,002	0,001	0,000	

TAB. 1 – Valeur du score de similarité s_g selon la distance entre deux années. On peut vérifier que la somme de ces valeurs pour $d_p - d_r$ variant entre -15 et $+15$ est 10.

Extension à des hypothèses multiples avec score de confiance Dans la situation où un système donne plusieurs hypothèses de dates pour un fragment d'article, le gain assigné à cet ensemble d'hypothèses est la combinaison linéaire des gains de chaque hypothèse, pondérée par les scores de confiance donnés par le système. Mis en formules : pour un fragment d'article a_i, le système p prédit n_i dates d_p^j :

$$D_p\left(a_i\right) = \left(d_p^1, d_p^2, \ldots, d_p^{n_i}\right)$$

Le système p attribue la confiance c_p^j à la prédiction d_p^j :

$$C_p\left(a_i\right) = \left(c_p^1, c_p^2, \ldots, c_p^{n_i}\right) \text{ avec } \sum_{j=1}^{n_i} c_p^j = 1$$

Le score pondéré obtenu pour ce fragment d'article est alors :

$$s_c\left(a_i\right) = \frac{1}{n_i} C_p\left(a_i\right) \cdot D_p\left(a_i\right) = \frac{1}{n_i} \sum_{j=1}^{n_i} c_p^j \cdot s\left(d_p^j\left(a_i\right), d_r\left(a_i\right)\right) \tag{4}$$

ce qui donne la formule (5) pour l'évaluation d'un fragment d'article avec n_i hypothèses d_p^j pondérées par les scores de confiance c_p^j :

$$s_c\left(a_i\right) = \frac{1}{n_i} \sum_{j=1}^{n_i} c_p^j \cdot e^{-\frac{\pi}{10^2}\left(d_p^j(a_i) - d_r(a_i)\right)^2} \tag{5}$$

et la formule (6) pour l'évaluation globale des résultats d'un système p produisant des hypothèses multiples pondérées par score de confiance :

$$S_c\left(p\right) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} c_p^j \cdot e^{-\frac{\pi}{10^2}\left(d_p^j(a_i) - d_r(a_i)\right)^2} \tag{6}$$

2.3 Tests

2.3.1 Tests humains

Des tests ont été effectués auprès de cinq évaluateurs humains sur les 15 premiers documents du corpus d'apprentissage distribué aux participants. Chaque évaluateur a mis entre 30 et 45 minutes pour travailler sur ce corpus. Les résultats varient fortement : 0,879 – 0,691 – 0,443 – 0,303 et 0,201, pour un score moyen de 0,503. Notons qu'un tirage aléatoire de dates engendre un score final de 0,071. Tous les évaluateurs humains ont effectué un repérage d'entités nommées qu'ils ont ensuite essayé de dater ; ils ont pour la plupart effectué des recherches dans Google et dans l'encyclopédie Wikipédia. Nous observons que l'évaluatrice qui s'est classée première sur ces tests a suivi des études d'histoire et sciences de l'information. L'évaluatrice classée deuxième a, en plus de la datation des entités nommées, également cherché à définir une thématique générale traitée dans le document (conflit, politique, théâtre, etc.) puis effectué une recherche sur Internet combinant cette thématique avec les entités nommées précédemment identifiées.

Afin de représenter la précision de chaque évaluateur et l'évolution des datations, nous avons établi un graphique à la manière des courbes ROC présentant l'incrémentation du nombre de documents correctement identifiés en

augmentant au fur et à mesure la distance en années vis à vis de la référence. Sur ce graphique (voir figure 2), un système précis est celui qui identifiera le maximum de documents avec le minimum d'écart par rapport à la référence, notamment parmi les quinze premières années qui sont celles qui rapportent des points (cf. l'importance des surfaces jaune et vert foncé par rapport aux surfaces vert clair ou bleu clair) : l'évaluatrice classée première a identifié 11 documents avec l'année exacte, lui garantissant une surface importante dès l'année 0 de distance (surface jaune, score de 0,879). À l'inverse, le 11ème document identifié par le dernier évaluateur humain l'a été avec presque 40 ans d'écart (zone bleu clair, score de 0,201). Un tirage aléatoire (zone vert clair, score de 0,071) témoigne d'une précision encore moindre que celle obtenue par les évaluateurs humains. Ces tests révèlent une tâche difficile mais qui devrait engendrer des disparités selon les méthodes suivies par les participants.

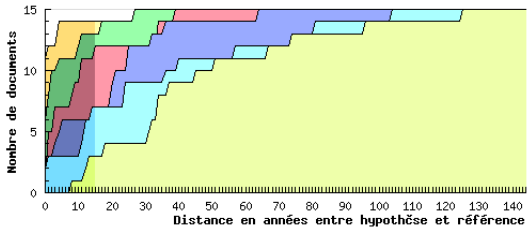


FIG. 2 – Surfaces présentant les résultats des évaluateurs humains sur la tâche diachronie

2.3.2 Test automatique

Nous avons procédé à un deuxième type de test, fondé sur l'adaptation aux données de DEFT2011 d'un système ayant participé à la tâche diachronie pour DEFT2010. Pour ce second test, Rémy Kessler, membre de l'équipe du LIA l'année précédente, a modifié l'un des systèmes entrant dans la composition de la chaîne de traitements utilisée par cette équipe (Oger *et al.*, 2010). Ce système utilise un classifieur à large marge spécialisé dans le traitement de données textuelles, ICSIBOOST[3] développée par le laboratoire ICSI[4] et basé sur un algorithme de boosting (Schapire & Singer, 2000) de classifieurs simples (des arbres de décision à 1 niveau de profondeur sur la présence ou l'absence de n-grammes). L'adaptation de ce système aux données de DEFT2011 s'est faite en l'espace d'une après-midi. L'ensemble des paramètres tels que les prétraitements linguistiques, le remplacement de la ponctuation par des balises lexicales ou T, le nombre de tours de l'algorithme ont été conservés comme lors de l'édition 2010. Les principales modifications ont été l'adaptation des formats d'entrées et de sorties afin de prédire des décennies ou des années en fonction du besoin.

Le système ainsi modifié a été évalué sur le corpus de test de DEFT2011 pour les deux pistes proposées aux participants, avec deux types de sortie (voir tableau 2) : en premier lieu, le système a renvoyé des décennies, à l'image de ce qui était demandé en 2010 ; en second lieu, le système a produit des années, tel que demandé pour la présente édition du défi. L'évaluation sur des décennies est meilleure que celle portant sur des années, ce qui est conforme avec l'idée qu'un nombre plus réduit de classes (15 classes pour les décennies contre 144 classes pour les années) permet d'obtenir de meilleurs résultats. On notera cependant un accroissement important des temps d'apprentissages pour le système avec la version 144 classes. Comparativement aux résultats obtenus par les participants de cette année (voir tableau 3), ce système se classe virtuellement troisième sur les deux pistes, mais avec des scores inférieurs aux moyennes et médianes calculées sur les meilleures soumissions.

	Décennies		**Années**	
Évaluation	Piste 1	Piste 2	Piste 1	Piste 2
Sans confiance	0,236	0,287	0,140	0,167
Avec confiance	—	—	0,109	0,108

TAB. 2 – Scores obtenus par l'adaptation d'un outil DEFT2010 aux données DEFT2011

[3] http://code.google.com/p/icsiboost/
[4] http://www.icsi.berkeley.edu/

2.4 Résultats des participants

Les résultats des participants (tableau 3) sont tout juste inférieurs à la moyenne des tests humains (0,503). Les résultats entre parenthèses correspondent à des soumissions reçues après la fin de la période de test, le participant pensant les avoir soumises en temps et en heure. Nous les indiquons néanmoins car ils représentent un travail conséquent. Comme nous l'envisagions, les résultats sur des extraits de 500 mots sont supérieurs (score moyen de 0,332) à ceux obtenus sur des extraits de 300 mots (score moyen de 0,247). Le corpus provenant d'une reconnaissance optique de caractères sur de journaux anciens, il apparaît légitime d'obtenir de meilleurs résultats sur un ensemble plus vaste de données.

	Piste 1			**Piste 2**		
Équipe et renvoi bibliographique	Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
EBSI (Boley, 2011)	0,062	0,073	0,061	0,069	—	—
IRISA (Raymond & Claveau, 2011)	0,342	0,317	**0,472**	0,266	0,285	**0,430**
LIMSI (García-Fernandez *et al.*, 2011)	0,452	0,428	0,363	0,378	0,374	0,358
LUTIN-a (El Ghali, 2011)	(0,098)	(0,108)	(0,100)	0,113	(0,117)	(0,081)
Moyenne	*0,332*			*0,247*		
Médiane	*0,452*			*0,358*		
Écart-type	*0,225*			*0,183*		
Variance	*0,051*			*0,033*		

TAB. 3 – Scores des participants, moyenne, médiane, écart-type et variance sur les meilleures soumissions

Comme pour les tests humains, nous avons représenté la progression des résultats des participants sous la forme de surfaces (figure 3). Puisque les 15 premières années sont les seules à apporter un gain dans l'évaluation finale, nous avons effectué un zoom sur ces premières années (figure 4). Le code couleur utilisé est le suivant (par ordre d'apparition) : IRISA (orange), LIMSI (jaune), LUTIN-a (bleu), EBSI (vert clair).

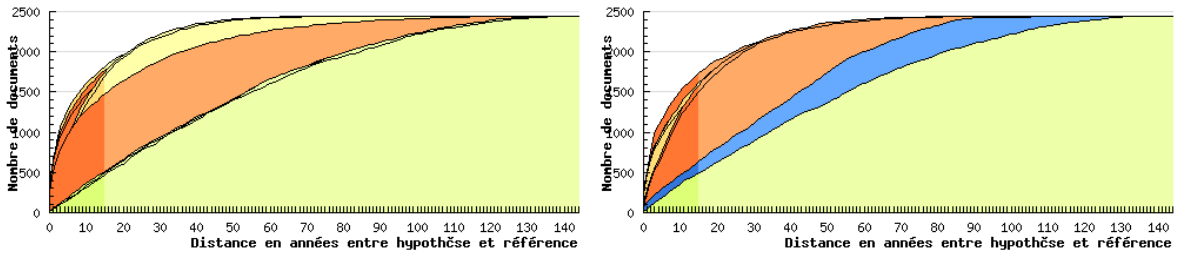


FIG. 3 – Précisions obtenues par les systèmes sur les pistes 1 (graphique de gauche) et 2 (graphique de droite)

2.5 Méthodes des participants

Les participants à cette tâche ont tous eu recours à des méthodes à base d'apprentissage, avec des résultats hétérogènes : un classifieur bayésien naïf pour (Boley, 2011), des SVM par (García-Fernandez *et al.*, 2011), les arbres de décisions et les k-plus-proches voisins par (Raymond & Claveau, 2011). Au niveau linguistique, (Boley, 2011) a comparé les stratégies sémantiques et asémantiques sur ce type de données ; alors que la stratégie sémantique se dégageait nettement sur le corpus d'apprentissage, les résultats sont équivalents sur le corpus de test. (García-Fernandez *et al.*, 2011) ont produit des ressources chronologiques externes aux corpus (une base de données des dates de naissance de personnes célèbres nées entre 1781 et 1944) et effectué une analyse linguistique (étude des archaïsmes et néologismes, études des réformes orthographiques) s'inspirant des travaux de (Albert *et al.*, 2010). Partant du principe que plus un document est ancien, plus la reconnaissance des caractères sera bruitée (les tâches

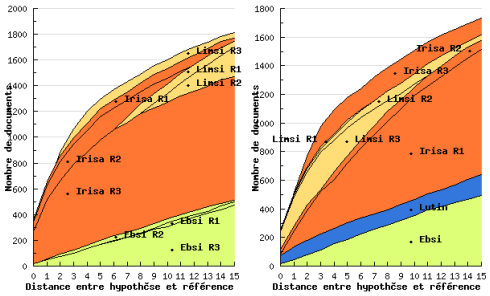


FIG. 4 – Précisions obtenues par les systèmes sur les pistes 1 (graphique de gauche) et 2 (graphique de droite), zoom effectué sur les distances inférieures à 15 ans vis à vis de la référence

d'encre sur un papier fin étant reconnues à tort comme des signes de ponctuation), (Raymond & Claveau, 2011) ont pris en compte la fréquence des ponctuations au fil du temps.

3 Tâche 2. Appariements

Pour cette seconde tâche, nous avons décidé de nous focaliser sur le résumé d'articles scientifiques. Plutôt que de se placer dans une tâche de génération automatique de résumés, contraignante au niveau de l'évaluation – *Qu'est-ce qui constitue un résumé de référence ? Comment évaluer la pertinence d'un résumé ?* – comme c'est le cas dans des campagnes d'évaluation telles TSC sur le résumé automatique (Okumura *et al.*, 2003), nous avons adopté le point de vue inverse qui consiste à utiliser les résumés déjà associés à des articles scientifiques, et à proposer comme tâche d'apparier chaque résumé avec l'article scientifique pour lequel il a été écrit. De récents travaux se sont penchés sur le rapport entre le contenu du résumé et celui de l'article scientifique qui lui correspond, en particulier dans le domaine biomédical (Cohen *et al.*, 2010), permettant de mettre en évidence des caractéristiques propres aux résumés, et propres aux corps des articles scientifiques. Nous émettons l'hypothèse que les méthodes permettant de relier les résumés aux articles devraient permettre de mettre en évidence les éléments saillants d'un résumé, ces éléments et méthodes pouvant par la suite conduire au développement d'outils de génération automatique de résumés.

Dans le cadre de cette tâche, nous avons proposé deux pistes aux participants. En premier lieu, apparier chaque résumé avec l'article scientifique complet qui lui correspond. En second lieu, apparier chaque résumé avec l'article scientifique qui lui correspond, l'article ayant été amputé de son introduction et de sa conclusion. Pour cette seconde piste, nous émettons l'hypothèse que les éléments présents dans l'introduction et la conclusion sont davantage repris dans le résumé, à tout le moins sous une forme quasi identique. Les résultats sur cette piste devraient donc être moins bons que ceux obtenus en étudiant les articles au complet de la première piste.

3.1 Corpus

Les corpus de cette tâche ont été constitués à partir d'articles scientifiques parus dans des revues en Sciences Humaines et Sociales. Ces articles proviennent de la plateforme universitaire Erudit[5]. Chaque article est disponible au format XML avec une structuration permettant la récupération distincte des méta-données (auteurs, résumé, bibliographie, etc.) et du contenu (article scientifique structuré en sections et paragraphes). Nous avons constitué le corpus de la piste 1 en désolidarisant le résumé de l'article de chaque fichier XML, en deux fichiers distincts. Le corpus de la piste 2 a été constitué en supprimant les introduction et conclusion de chaque article scientifique (production d'un fichier de texte scientifique « .txt »). Si les introduction et conclusion n'étaient pas clairement indiquées sous la forme de sections nommées, nous avons considéré les paragraphes précédent le premier titre de section comme étant une introduction, et les paragraphes de la dernière section comme étant la conclusion.

[5] `http://www.erudit.org/` – plateforme de diffusion issue d'un consortium universitaire québécois.

Nous avons utilisé le même jeu d'articles scientifiques pour les deux pistes, les fichiers de résumés, d'articles et de textes ayant été nommés différemment dans les deux pistes.

Le corpus d'apprentissage compte 5 revues (*Anthropologie et Société, Études Internationales, Études littéraires, Philosophiques* et la *Revue des Sciences de l'Éducation*) avec 60 articles par revue. Nous avons ajouté une sixième revue (*Meta*) dans le corpus de test de manière à éprouver la robustesse des systèmes sur une source inconnue ; chaque revue de ce corpus intégrant une trentaine d'articles. Le nombre moyen de mots par article varie selon les revues, avec du nombre moyen le plus long au plus court : *Études Internationales, Revue des Sciences de l'Éducation, Philosophiques, Anthropologie et Société, Études Littéraires,* et enfin *Meta* (voir tableau 4).

Corpus	**Apprentissage**		**Test**	
	Articles	Textes	Articles	Textes
Études Internationales	7044	5557	7032	5236
Revue des Sciences de l'Éducation	6332	5352	6049	5143
Philosophiques	5343	4687	6912	5004
Anthropologie et Société	5549	4358	5889	4540
Études Littéraires	4814	3489	4883	3559
Meta	—	—	4136	3481

TAB. 4 – Nombre moyen de mots par article (pour chaque revue et corpus)

3.2 Méthodes d'évaluation

Nous considérons que l'hypothèse retournée par le système est correcte ou pas (évaluation binaire). On peut compter la proportion de résumés pour lesquels l'hypothèse fournie est correcte. Comme tout résumé doit recevoir une réponse, et que cette réponse est unique, cette proportion peut être vue aussi bien comme une précision (proportion des réponses proposées qui sont correctes) que comme un rappel (proportion des réponses attendues qui sont correctement proposées par le système) ou encore une correction (proportion des décisions qui sont correctes).

Si l'on définit un score élémentaire pour chaque résumé qui vaut 1 ou 0 selon que l'article trouvé est correct ou pas, cela revient à calculer la moyenne de ce score sur l'ensemble des résumés. Mis en formules, pour chacun des n résumés r_i, le système prédit quel article $a_p(r_i)$ parmi les N articles a_j lui correspond. Le score $s(a_p(r_i), a_r(r_i))$ donné à chaque prédiction vaut 0 ou 1 selon que l'article prédit $a_p(r_i)$ est ou pas l'article de référence $a_r(r_i)$:

$$s(a_p(r_i), a_r(r_i)) = \{ 1 \ si \ a_p(r_i) = a_r(r_i) \ 0 \ sinon$$

Le score global est la moyenne des scores obtenus par le système p :

$$S(p) = \frac{1}{N} \sum_{i=1}^{n} s(a_p(r_i), a_r(r_i)) = \frac{1}{N} \sum_{i=1}^{N} |\{r_i \,;\, a_p(r_i) = a_r(r_i)\}| \tag{7}$$

Extension à des hypothèses multiples avec indice de confiance Dans cette variante, le système peut donner plusieurs hypothèses d'articles pour chaque résumé, en associant un indice de confiance à chaque hypothèse. Si l'une de ces hypothèses est correcte, le score attribué au système est l'indice de confiance que le système a associé à cette hypothèse ; si aucune hypothèse n'est correcte, le score est nul. Comme dans le cas à une seule étiquette, le score global pour l'ensemble des résumés est la moyenne des scores par résumé.

Mis en formules : pour un résumé r_i, le système p prédit n_i étiquettes a_p^j :

$$A_p(r_i) = \left(a_p^1, a_p^2, \ldots, a_p^{n_i}\right)$$

Le système attribue la confiance c_p^j à la prédiction a_p^j :

$$C_p(r_i) = \left(c_p^1, c_p^2, \ldots, c_p^{n_i}\right) \text{ avec } \sum_{j=1}^{n_i} c_p^j = 1$$

Le score pondéré obtenu pour ce résumé est alors :

$$s_c\left(A_p\left(r_i\right), C_p\left(r_i\right), a_r\left(r_i\right)\right) = \{\, c_p^{j_i}\left(r_i\right) si\ \exists j_i \in \{1 \ldots n_i\}\ ;\ a_p^{j_i}\left(r_i\right) = a_r\left(r_i\right) 0 sinon \tag{8}$$

ce qui donne la formule (9) pour l'évaluation globale des résultats d'un système p produisant des hypothèses multiples pondérées par score de confiance :

$$S_c\left(p\right) = \frac{1}{N}\sum_{i=1}^{N} s_c\left(A_p\left(r_i\right), C_p\left(r_i\right), a_r\left(r_i\right)\right) = \frac{1}{N}\sum_{\left\{i\,;\,\exists j_i \in \{1 \ldots n_i\}\,;\,a_p^{j_i}(r_i) = a_r(r_i)\right\}} c_p^{j_i}\left(r_i\right) \tag{9}$$

3.3 Tests humains

Les évaluateurs humains ont travaillé à l'identification de 15 couples résumé/article et résumé/texte provenant de deux revues (*Anthropologie et Société* et *Études Internationales*). Ils ont tous correctement identifié les paires résumé/article et résumé/texte, obtenant le score maximal, en au plus d'une demi-heure. Nous en concluons que la tâche est aisée pour un humain et ne devrait pas poser trop de problèmes pour un système.

3.4 Résultats des participants

Pendant la phase de test, un participant nous a signalé l'existence d'un fichier texte vide « 077.txt », ce qui se révèle particulièrement handicapant pour l'aligner avec le fichier de résumé qui lui correspond, « 017.res ». La correspondance a été communiquée à ce participant. Pour les autres participants, nous avons fait le choix de ne pas transmettre cette information mais de modifier le script d'évaluation en conséquence : premièrement, pour ne pas évaluer la sortie sur « 017.res » (appariemment forcément erroné) et deuxièmement, pour accorder gratuitement un point correspondant au point perdu par le participant qui apparie le fichier « 077.txt » avec un résumé quelconque.

3.4.1 Réponse unique : classement officiel

Il s'agit des mesures qui seront utilisées pour le classement final. Si le participant a envoyé des soumissions sans score de confiance, nous prenons la réponse fournie. Si le participant a transmis des soumissions avec score de confiance, nous ne retenons que la réponse ayant le score de confiance le plus élevé.

	Piste 1			**Piste 2**		
Équipe et renvoi bibliographique	Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
CHART (Hoareau *et al.*, 2011)	**1,000**	—	—	0,995	—	—
EBSI (Boley, 2011)	0,980	0,985	0,980	0,954	0,954	—
FBK (Tonelli & Pianta, 2011)	0,975	0,960	0,990	0,964	0,934	0,964
GREYC (Lejeune *et al.*, 2011)	**1,000**	0,975	0,626	0,909	0,482	—
INAOE (Sánchez-Vega *et al.*, 2011)	0,970	0,960	0,949	0,904	0,848	0,858
IRISA (Raymond & Claveau, 2011)	0,995	—	—	0,990	—	—
LORIA (Cadot *et al.*, 2011)	**1,000**	—	—	**1,000**	—	—
LUTIN-a (El Ghali, 2011)	0,965	0,934	0,970	0,919	0,883	0,873
LUTIN-d (Devatman Hromada, 2011)	0,909	—	—	0,873	—	—
UCL (Bestgen, 2011)	**1,000**	—	—	**1,000**	—	—
UPF (Saggion, 2011)	0,975	0,975	—	0,959	0,959	—
Moyenne	*0,981*			*0,956*		
Médiane	*0,990*			*0,959*		
Écart-type	*0,027*			*0,042*		
Variance	*0,001*			*0,002*		

TAB. 5 – Résultats des participants sur les réponses de rang 1, moyenne, médiane, écart-type et variance calculés sur les meilleures soumissions

3.4.2 Réponses avec confiance : évaluation alternative

Cette seconde évaluation prend en compte toutes les réponses fournies par le participant et pondère le score d'association par la confiance renseignée par le système. Alors que dans le classement officiel, une réponse juste vaut 1 point, ici elle vaut la valeur du score de confiance qui lui a été associée, ce qui conduit à une dégradation des résultats. Les participants qui n'ont pas utilisé de score de confiance obtiennent les mêmes scores que dans le classement officiel (cela revient à avoir affecté une confiance de 1 à chaque résultat).

	Piste 1			**Piste 2**		
Équipe	Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
INAOE	0,417	0,412	0,389	0,368	0,345	0,348
UPF	**0,512**	—	—	**0,402**	—	—
Moyenne	*0,465*			*0,385*		
Médiane	*0,465*			*0,385*		
Écart-type	*0,067*			*0,024*		
Variance	*0,005*			*0,001*		

TAB. 6 – Résultats des participants avec prise en compte du score de confiance, moyenne, médiane, écart-type et variance calculés sur les meilleures soumissions

3.5 Méthodes des participants

La majorité des participants a envisagé cette tâche comme étant de la recherche d'information. (Raymond & Claveau, 2011) ont ainsi considéré que le résumé constituait la requête alors que (Lejeune *et al.*, 2011) ont adopté la démarche inverse consistant à apparier un article avec les différents résumés. La majorité des participants a utilisé une représentation vectorielle des documents, parfois avec une pondération issue du tf*idf et une lemmatisation, la similarité étant généralement calculée au moyen du cosinus ou de la distance euclidienne. (Hoareau *et al.*, 2011) ont combiné les espaces sémantiques vectoriels aux modèles de graphe. (Cadot *et al.*, 2011) ont mis au point une méthode d'appariement qui s'apparente à celle des voisins réciproques « résumé-texte et texte-résumé », en univers fermé, robuste et sans nécessité d'information extérieure. (Bestgen, 2011) a développé une approche fondée sur trois composants : l'analyse sémantique latente (LSA), les machines à support vectoriel (SVM) et l'assignation finale selon l'algorithme du meilleur d'abord (MA). (Devatman Hromada, 2011) a utilisé une approche simple consistant à comparer la fréquence d'utilisation des mots dans les résumés avec celle dans les articles.

Conclusion

La première tâche de datation d'archives de journaux issues d'une reconnaissance de caractères a donné lieu à l'utilisation de différentes méthodes d'apprentissages, certaines étant combinées avec des ressources linguistiques. Contrairement à l'édition 2010 où nous attendions des participants qu'ils identifient la décennie de publication, l'édition 2011 s'est focalisée sur l'année exacte, soit 144 années contre 15 décennies. Les résultats obtenus cette année se révèlent néanmoins plus élevés que ceux obtenus lors de la précédente édition. La méthode ayant obtenu les meilleurs résultats repose sur l'utilisation des k-plus-proches voisins sans recourir à des données externes mais en associant des étiquettes morpho-syntaxiques à chaque token. La seconde tâche relative aux appariements résumés/articles et résumés/textes s'est révélée facile à traiter, au regard des résultats obtenus par les participants. La majorité des méthodes utilisées ne nécessite pas de ressources externes et a été envisagée comme étant une tâche de recherche d'information.

Remerciements

Nous remercions la plateforme Erudit pour la mise à disposition des articles scientifiques et le portail Gallica pour la possibilité d'utiliser les archives de presse. Nous remercions les évaluateurs humains (Marcela Baiocchi,

Roxane Cayer-Tardif, Janie Gauthier-Boudreau et Laurie-Anne Gignac) pour le temps passé à tester la faisabilité des différentes tâches et les résultats obtenus, ces derniers nous ayant confortés dans l'opportunité de proposer ces tâches aux participants. Nous remercions également tous les participants de l'édition 2011 pour les méthodes originales qu'ils ont développées dans le cadre de ce défi et les idées nouvelles qu'ils ont apportées au domaine. Enfin, nous remercions Rémy Kessler, membre de l'équipe du LIA à DEFT2010, pour avoir adapté son système de datation aux données de DEFT2011 de manière à tester la tâche sur la diachronie et à évaluer les possibilités d'adaptation d'un tel système sur de nouvelles données.

Ces travaux ont été en partie réalisés dans le cadre du programme Quaero, financé par Oseo, agence française pour l'innovation.

Références

ALBERT P., BADIN F., DELORME M., DEVOS N., PAPAZOGLOU S. & SIMARD J. (2010). Décennie d'un article de journal par analyse statistique et lexicale. In *Actes DEFT 2010*.

BESTGEN Y. (2011). LSVMA : au plus deux composants pour apparier des résumés à des articles. In *Actes DEFT 2011*.

BOLEY R. (2011). Comparaison de méthodes sémantiques et asémantiques pour la catégorisation automatique de documents. In *Actes DEFT 2011*.

CADOT M., AUBIN S. & LELU A. (2011). Indexer, comparer, apparier des textes et leurs résumés : une exploration. In *Actes DEFT 2011*.

COHEN K. B., JOHNSON H. L., VERSPOOR K., ROEDER C. & HUNTER L. E. (2010). The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC Bioinformatics*, **11**(492).

DEVATMAN HROMADA D. (2011). One simple formula for losing DEFT with more than 90% of correct guesses. In *Actes DEFT 2011*.

EL GHALI A. (2011). Expérimentations autour des espaces sémantiques hybrides. In *Actes DEFT 2011*.

GARCÍA-FERNANDEZ A., LIGOZAT A.-L., DINARELLI M. & BERNHARD D. (2011). Méthodes pour l'archéologie linguistique : datation par combinaison d'indices temporels. In *Actes DEFT 2011*.

GROUIN C., FOREST D., SYLVA L. D., PAROUBEK P. & ZWEIGENBAUM P. (2010). Présentation et résultats du défi fouille de texte DEFT2010 : Où et quand un article de presse a-t-il été écrit ? In *Actes DEFT 2010*.

HOAREAU Y. V., AHAT M., PETERMANN C. & BUI M. (2011). Couplage d'espaces sémantiques et de graphes pour le deft 2011 : une approche automatique non supervisée. In *Actes DEFT 2011*.

LEJEUNE G., BRIXTEL R. & GIGUET E. (2011). Deft 2011 : Appariement de résumés et d'articles scientifiques fondé sur des similarités de chaînes de caractères. In *Actes DEFT 2011*.

OGER S., ROUVIER M., CAMELIN N., KESSLER R., LEFÈVRE F. & TORRES-MORENO J.-M. (2010). Système du LIA pour la campagne DEFT'10 : datation et localisation d'articles de presse francophones. In *Actes DEFT 2010*.

OKUMURA M., FUKUSIMA T. & NANBA H. (2003). Text summarization challenge 2 : text summarization evaluation at NTCIR workshop 3. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop*.

RAYMOND C. & CLAVEAU V. (2011). Participation de l'IRISA à DEFT2011 : expériences avec des approches d'apprentissage supervisé et non supervisé. In *Actes DEFT 2011*.

SAGGION H. (2011). Matching Texts with SUMMA. In *Actes DEFT 2011*.

SÁNCHEZ-VEGA F., VILLATORO-TELLO E., JUÁREZ-GOZÁLEZ A., VILLASENOR-PINEDA L., Y GÓMEZ M. M. & MENESES-LERÍN L. (2011). INAOE DEFT2011 : Using a Plagiarism Detection Method for Pairing Abstracts-Scientific Papers. In *Actes DEFT 2011*.

SCHAPIRE R. E. & SINGER Y. (2000). BoosTexter : A Boosting-based System for Text Categorization. *Machine Learning*, **39**(2/3), 135–168.

TONELLI S. & PIANTA E. (2011). Matching documents and summaries using key-concepts. In *Actes DEFT 2011*.

Tâche 1. Diachronie

Participation de l'IRISA à DEFT 2011: expériences avec des approches d'apprentissage supervisé et non-supervisé

Christian Raymond[1,2,3] Vincent Claveau[1,4]
(1) IRISA, Campus de Beaulieu, 35042, Rennes, France
(2) Université Européenne de Bretagne
(3) INSA de Rennes,35708, Rennes, France
(4) CNRS, Rennes, France
christian.raymond@irisa.fr, vincent.claveau@irisa.fr

Résumé. Cet article présente la participation de l'équipe TexMex de l'IRISA à DEFT 2011. Nous avons participé aux deux tâches proposées et à toutes les pistes. Nous avons exploré différentes approches. Nous avons notamment employé des techniques d'apprentissage particulières à base de *boosting* et de *lazy-learning* et des pondérations issues du domaine de la recherche d'information. Ces différentes approches nous ont permis d'obtenir de bons résultats et de nous classer premiers sur la tâche de datation et d'obtenir une précision de 99 % et 99.5 % sur la tâche d'appariement.

Abstract. This article presents the participation of IRISA TexMex team at DEFT in 2011. We participated in the two proposed tasks and all tracks. We explored different approaches. We employed specific learning techniques based on *boosting* over decision trees and *lazy-learning* together with weights from the information retrieval field. These different approaches enabled us to obtain good results since we rank first on the task of dating and we obtained an accuracy of 99% and 99.5% on the pairing task.

Mots-clés : Classification, boosting, arbre de décision, bonzaiboost, Okapi, apprentissage paresseux, k-plus-proches voisins.

Keywords: Classification, boosting, decision tree, bonzaiboost, Okapi, lazy learning, k-nearest neighbours.

1 Introduction

Cet article décrit la participation de l'équipe TexMex de l'IRISA à ce défi. Il s'agit de notre première participation à cette compétition. Les deux tâches proposées dans le cadre du DÉfi Fouille de Texte (DEFT) en 2011 portaient sur la classification de documents, qui est un domaine d'intérêt pour nous.

Les délais très courts (phase d'entraînement de 5 semaines) imposés par les contraintes d'organisation nous ont orientés sur des mise-en-œuvre simples mais efficaces. Pour ce faire, nous nous sommes appuyés notamment sur des techniques classiques de recherche d'information ou des techniques d'apprentissage développées en interne. Malgré ce manque de temps, les bonnes performances obtenues et nos classements montrent le bien fondé de ces approches. Par ailleurs, nos différentes approches ont pour point commun de n'avoir pas recours à des connaissances externes.

L'article est structuré comme suit. Les deux sections suivantes décrivent les deux approches utilisées pour la tâche de datation (les runs 1 et 2 pour les deux pistes de la tâches 1 sont donc décrits en section 2 et le run 3 dans la section 3). La section 4 décrit l'approche utilisée pour la tâche 2 d'appariement résumé/article.

2 Contribution à la tâche 1 : classification

La première piste envisagée dans la résolution de la tâche 1 a été une approche à base de classification. Cela a permis dans un premier temps de cerner la difficulté de la tâche qui *a priori* devait être compliquée pour au moins trois raisons :

1. retrouver une période temporelle semble raisonnable, retrouver une année en particulier beaucoup plus difficile ;
2. il faut discriminer entre 145 années différentes ;
3. l'entrée est très bruitée en raison des erreurs induites par l'OCR, même si c'est le problème qui, de notre point de vue, ne paraissait pas le plus critique.

En raison des deux premiers points soulevés, une approche de classification classique ne semble pas adaptée, mais un test en condition réelle s'impose pour avoir une confirmation. L'idée principale était ensuite d'adapter le classifieur ou la présentation de la tâche elle-même pour la résoudre en utilisant une méthode d'apprentissage supervisé.

2.1 Pré-traitement des données

Le pré-traitement des données semblait être une phase importante dans le traitement de cette tâche, par manque de temps nous n'avons malheureusement pas vraiment approfondi cette partie. Les traitements effectués (ou envisagés) sont (ont été) les suivants :

– la première des choses à laquelle on pense est la correction des erreurs de l'OCR, mais nous avons réalisé que la tâche en elle-même était compliquée : outre des erreurs de graphie, des erreurs de segmentation sont glissées ce qui rend non seulement la correction aussi bruitée que la non-correction, mais il ne semble pas évident qu'une correction parfaite puisse véritablement apporter un avantage indéniable dans la résolution de la tâche principale. Nous avons donc abandonné ce traitement ;
– comme le souligne (Oger *et al.*, 2010) dans l'édition de DEFT précédente, il est légitime de penser que les erreurs de l'OCR peuvent être dues à la qualité du document numérisé. Les OCR sont assez sensibles à la qualité du papier, à la police de caractères, à l'encre utilisée, *etc.* On peut donc penser que le nombre d'erreurs rencontrées est lié à la date d'écriture du document. Au plus un document est ancien, au plus le taux d'erreur est élevé. Nous proposons d'utiliser la ponctuation car les artefacts d'un document (tâches d'encre, *etc.*) sont souvent transformés par l'OCR en signes de ponctuation. Pour chaque document nous conserverons les fréquences d'apparition des signes de ponctuation.

Nous avons par la suite adopté un pré-traitement classique : pour l' approche à base de classification nous avons mis à notre disposition trois attributs :

1. le texte lui-même (sauf la ponctuation) où à chaque mot est associé une étiquette

- premièrement, les étiquettes associées aux mots sont des étiquettes morpho-syntaxiques. En raison de l'entrée bruitée, nous n'avons pas utilisé d'analyseur, nous avons juste associé à un mot son étiquette la plus probable sans tenir compte du contexte.
- deuxièmement, nous avons enrichi ce jeu d'étiquettes par des étiquettes provenant de liste de connaissances *a priori* communes (*i.e.* villes, pays, mois, jours de la semaine, titres de noblesse, grade militaire, et quelques autres).
- troisièmement, nous n'avons conservé que les couples (mot/étiquette) dont l'étiquette appartient soit à la liste des connaissances *a priori* soit à l'ensemble suivant des étiquettes morpho-syntaxique (noms,adjectifs,verbes) supprimant ainsi, nous l'espérons, une partie du bruit et de mots insignifiants.

2. en guise de deuxième attribut, nous avons pour chaque mot ou étiquette du premier attribut associé sa fréquence d'apparition dans le texte. Nous utilisons par exemple le nombre de verbes utilisé, le nombre de titres de noblesses utilisé,*etc.*
3. le troisième attribut est identique au premier excepté que nous n'avons pas appliqué le troisième point de traitement, tout les mots ont été conservés en espérant retrouver d'éventuelles figures de style à l'intérieur.

Pour la suite, si aucune mention contraire n'est posée, les classifieurs utilisés vont s'appuyer sur des descripteurs *N*-grammes dans les attributs 1 et 3 (de taille $<= 2$ pour l'attribut 1 et de taille $[2, 3]$ pour l'attribut 3), étant données les deux informations en entrée (*i.e.* le mot ainsi qu'une étiquette correspondante) tous les *N*-grammes issus de la combinaison de ces deux informations sont générés. L'attribut deux sera interprété comme du « scoredtext », c'est-à-dire que des seuils sur la valeur numérique seront évalués (*i.e.* on peut apprendre des seuils sur les fréquences d'apparition d'un mot ou d'une étiquette). Afin d'éliminer une partie du bruit et surtout faciliter la sélection de descripteurs pertinents des filtres sont appliqués : seuls les descripteurs ayant été observés au moins 2 fois pour l'attribut 1 sont conservés. La coupure est faite à 10 pour l'attribut 3. Le choix de la taille des *N*-gramme ou du paramètre de filtrage pour les attributs 1 et 3 a été guidé par les objectifs suivants :
– attribut 1 : taille *N*-gramme filtrage réduits : capturer les mots ou expressions caractéristiques d'une année
– attribut 3 : taille *N*-gramme > 1 et filtrage important : capturer des informations caractéristique d'une époque (*e.g.* style)
Le logiciel BonzaiBoost (Raymond, 2010) est utilisé pour l'extraction *N*-gramme et pour toutes les approches de classification .

2.2 Premier aperçu avec un arbre de décision

Pour des raisons de diagnostic sur la difficulté de la tâche j'ai décidé de commencer par l'apprentissage d'un arbre de décision car le modèle produit est facilement interprétable. J'ai développé un arbre classique basé sur un critère de segmentation entropique et un critère d'arrêt statistique qui stoppe l'induction lorsqu'il considère que le gain obtenu est trop faible pour que le descripteur choisi soit véritablement pertinent, résultat : l'arbre ne se développe pas. Bien entendu en présence de 145 classes aucun critère ne peut apporter de gain significatif. Lorsque l'on continu l'induction de l'arbre en utilisant des critères d'arrêts plus conventionnel (taille des feuilles) on obtient hélas un arbre dont même les premiers choix ne semblent guère pertinent, le premier descripteur choisi est par exemple un descripteur *N*-gramme de l'attribut 1 : « etoit ». Les performances sur la tâche 1 sont de l'ordre de 0.14 à 0.16 avec un critère d'arrêt sur la taille minimale d'une feuille fixé entre 5 et 30 documents.

Ces résultats étaient plus ou moins prévisibles : il n'existe pas ou peu de descripteurs caractéristiques d'une année. On peut par contre espérer qu'il existe des descripteurs caractéristiques d'une période temporelle. De plus, la nature même du problème implique que sa résolution ne doit pas être envisagée par une classification brutale en année : en effet les classes ne sont visiblement pas indépendantes et se tromper d'un an dans la prédiction ne doit pas avoir la même répercussion que de se tromper de 100 ans. Pour modéliser cela, nous avons appris un arbre de décision où le critère de segmentation est la minimisation de la variance autour de l'année médiane d'une feuille. Une fois de plus, même si le classifieur est beaucoup mieux adapté à la tâche, les descripteurs sélectionnés par l'arbre sont peu convaincants : l'arbre sélectionne des descripteurs réduisant la variance sur les sous-nœuds gauche et droit, c'est-à-dire des descripteurs qui permettent de séparer au mieux des documents antérieurs à une date dans le nœud gauche et des documents postérieurs à cette même date dans le nœud droit : vraisemblablement, il n'en existent pas de véritablement pertinents : sont choisis alors des descripteurs qui répondent à ce critère de manière circonstancielle qui ont une réalité sur le corpus d'apprentissage mais ne sont pas assez généraux. On peut remarquer dans l'arbre de la figure 1 que de nombreux critères sur la fréquence de caractères de ponctuation sont

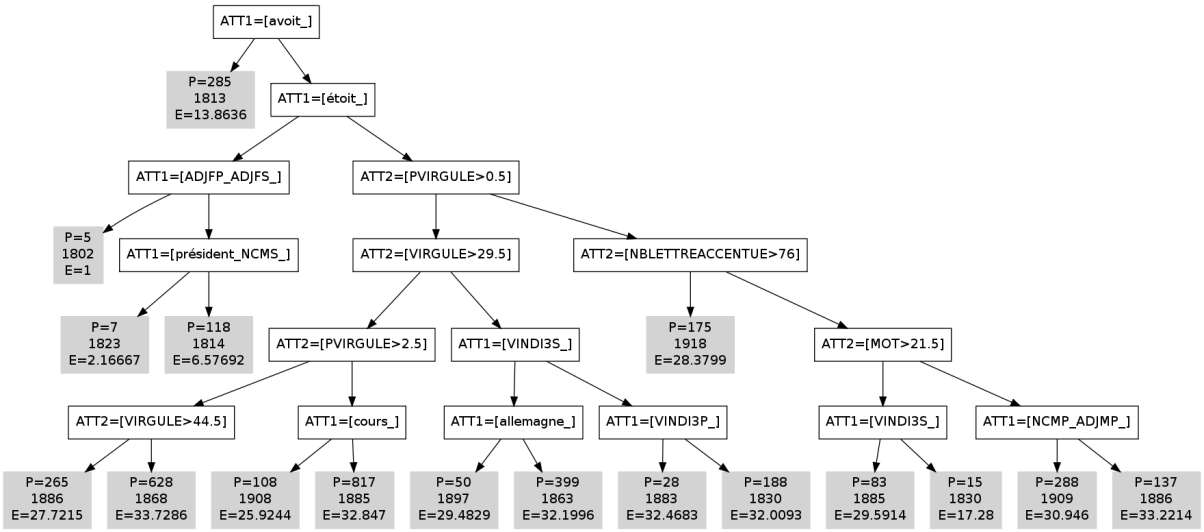


FIGURE 1 – Arbre de décision (élagué) construit en minimisant la variance autour d'une année médiane dans un nœud, chaque feuille indique : P=nombre de documents, l'année médiane, et E=l'écart type

sélectionnés confirmant l'hypothèse que la sur-reconnaissance de ceux-ci par l'OCR est potentiellement attaché à la mauvaise qualité des vieux documents. Les résultats sur la tâche 1 tourne autour de 0,17.

L'approche précédente n'a pas fonctionné, mais il semble pourtant que cette approche n'est pas dénuée de sens, les mauvais résultats précédents viennent du fait qu'il n'existe pas de descripteur pertinent pour décider si un document est supérieur ou inférieur à une année (qui fonctionne avec la plupart des documents), mais on suppose toujours qu'il existe des descripteurs caractéristiques d'une période. Pour modéliser cela en s'affranchissant du problème précédent, nous avons tenté l'induction d'un arbre minimisant la variance dans le nœud gauche exclusivement provoquant l'induction d'un « peigne » où chaque branche gauche pourrait être caractéristique d'une époque particulière. Pour éviter l'établissement de branches du peigne avec un seul document, nous avons imposé une taille minimale. Cette fois-ci, les descripteurs sélectionnés semblent très pertinents, tout du moins dans la partie haute de l'arbre, car malheureusement les descripteurs caractéristiques semblent s'épuiser très rapidement, et cette arbre-peigne qui semblait prometteur est au final inefficace globalement car il ne peut se développer au delà d'un certaines profondeur de manière efficace. La figure 2 montre un arbre-peigne illustrant les descripteurs caractéristiques de certaines périodes historiques.

2.3 Un peu plus de souplesse

Dû au manque évident de caractéristiques fortement discriminantes, il semble nécessaire d'adopter une approche de classification moins rigide. L'idée est de faire voter des participants (dont l'opinion n'est pas parfaite) et de combiner les décisions des participants. Les méthodes de boosting sont des méthodes permettant de combiner des classifieurs faibles pour obtenir au final un classifieur puissant. Les classifieurs faibles (*i.e.* les participants) sont ici des arbres de décision limités à 1 niveau (deux nœud/feuilles). L'algorithme de boosting utilisé est Adaboost.MH (Schapire & Singer, 2000). Une utilisation classique de cet algorithme sur les données présentées comme décrit dans la section 2.1 obtient un score de 0.226 sur la piste 1 (avec 1300 rounds de boosting). Les résultats ne sont pas vraiment excellents mais nettement meilleurs par rapport à ceux obtenus avec des arbres de décision classiques. Là aussi, le principal problème est la rigidité liée aux erreurs entre classes, une prédiction erronée de 1 an est considérée comme fausse au même titre qu'une erreur d'un siècle. Avec plus de temps, nous aurions aimé étudier les arbres dont la construction est guidée par la minimisation de la variance combinés avec un algorithme de boosting. Nous avons opté pragmatiquement pour une transformation du problème : le problème de classification ne sera plus de trouver la bonne année, mais de retrouver un ensemble de période temporelles. Pour faire simple, le problème K-classes est décomposé en un ensemble de K problèmes binaires où est indiqué si une année de référence est supérieure ou non à une des 145 années de la liste (*e.g.* à un document de 1910 sera associé la liste de labels : SUP_{1801}, SUP_{1802},...,SUP_{1909}). Le même algoritme de boosting, AdaBoost.MH étant un algorithme

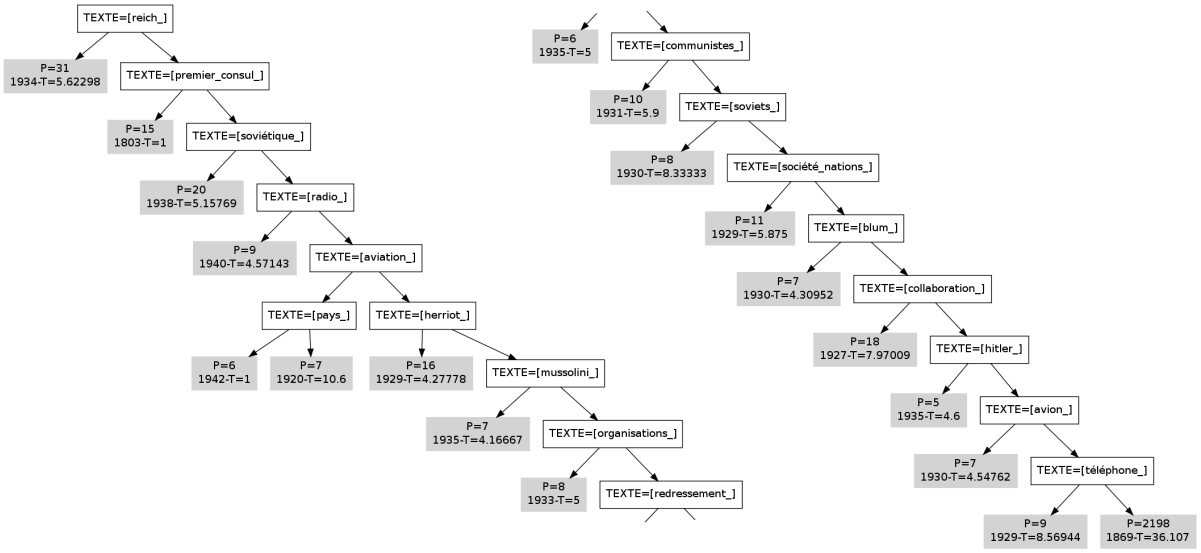


FIGURE 2 – arbre-peigne de profondeur 17 dont l'induction est basée sur la minimisation de la variance dans le nœud gauche. Chaque feuille indique : P=nombre de documents, l'année médiane, et T=l'écart type

multi-classes/multi-labels, est utilisé pour l'apprentissage. Ensuite une règle simple de vote permet de prendre une décision :

– pour chacun des labels possibles, s'il a été prédit, un vote est fait pour chacune des années supérieurs ou égales à l'année indiqué par le label, sinon le vote se porte sur chacune des années inférieures,
– on cumule le vote pour chacun des labels prédits pour un document,
– on sélectionne l'année qui obtient le plus de voix ;

Avec cette stratégie on espère obtenir les avantages suivants :

– chaque problème est plus simple à résoudre que le global
– chaque problème de séparation en deux profite de la sélection de multiples descripteurs au fur et à mesure des tours de boosting,
– on espère obtenir à la fin un ensemble d'informations qui permettent de localiser au moins grossièrement l'année même si le nombre d'erreurs est relativement élevé.

Cette méthode n'est au final pas forcément très convaincante non plus, car la rigidité de la barrière binaire (inférieur ou supérieur à une année) ne favorise pas la précision de l'algorithme (comme on pouvait s'y attendre mais dans le temps imparti nous n'avons pas eu le temps de creuser). Toutefois les résultats sont considérablement meilleurs que ceux obtenus par une approche directe sur l'année à prédire. Le résultat sur la tâche 1 étant aux alentours de 0,33. Cette méthode a été utilisé pour les runs 1 et 2 de la piste 1 et 2 de la tâche 1. La différence entre les deux runs concernent les filtres appliqués sur les descripteurs, donc sur le bruit des données d'entrées. Le tableau 1 illustre ce qui est appris durant les premiers tours de boosting.

2.4 Conclusion

L'approche par classification supervisée est limitée sur cette tâche. Les méthodes font de la sélection de descripteurs alors que peu de descripteurs sont véritablement discriminants. Le modèle obtenu a alors une vision très restreinte (exclusivement centrée sur les descripteurs qu'il a lui même retenus) et échoue dans la plupart des cas où il n'y a pas d'évidence. Une approche plus appropriée semble être de conserver un maximum d'information et de les exploiter de manière non-supervisée, voir section 3.

Tour	descripteur	présence	absence
1	étoit	[1813, 1944]	[1879, 1944]
2	VINDI3S	[1934, 1944]	[1802, 1937] 1942
3	PVIRGULE>0.5	[1802, 1944]	
4	au cours		[1802, 1944]
5	avoit	[1802, 1944]	
6	VIRGULE>29.5	1941 1943	[1802, 1944]
7	MOT>13.5	[1802, 1944]	
8	reich	1944	[1802, 1943]
9	M_Mme PRENOM MOTMAJ.	1804	[1802, 1803] [1805, 1944]
10	monsieur1 DETMS	[1826, 1944]	[1802, 1825]
11	long-temps	1824 [1810, 1820] [1827, 1944]	[1802, 1809] [1821, 1823] [1825, 1826]
12	NBLETTREACCENTUE>65.5	[1802, 1832]	[1833, 1944]
13	enfants		[1802, 1944]
14	la situation	[1937, 1941] [1943, 1944]	[1802, 1936] 1942
15	NBTITRENOBLESSE>0.5	1802 [1809, 1811] [1816, 1826] [1828, 1944]	[1803, 1807] [1812, 1815] 1827
16	écrit de	[1803, 1807] [1825, 1826] [1842, 1944]	1802 [1808, 1824] [1827, 1841]
17	télégraphie	1930 [1932, 1944]	[1802, 1931]
18	allemagne	[1802, 1811] [1932, 1944]	[1813, 1931]
19	cit MOT	[1802, 1944]	1944
20	enfans	1803, 04, 12, 13 [1815, 1819] [1828, 1944]	1802 [1804, 1814] [1816, 1827]
21	lit PREP DETMS	[1835, 1944]	[1802, 1834]
22	région		[1802, 1944]
23	DETMS « NCMS		[1802, 1944]
24	milieux	[1942, 1943]	1944 [1802, 1941]
25	président	[1935, 1944]	[1802, 1934]
26	DETMS NOMBRE PREP	[1802, 1944]	
27	VIRGULE>22.5	1927 [1922, 1925] [1934, 1944]	[1802, 1921] 1924, 26, 36, 43 [1928, 1934]
28	société des nations	[1935, 1944]	[1802, 1934]

TABLE 1 – Détails des 28 descripteurs choisis par les 28 arbres de décision construits au long de 28 tours de boosting. Pour chaque ligne la colonne « présence » montre les années pour lesquelles le classifieur donne son vote si le descripteur est présent dans le document, la colonne « absence » montre les années pour lesquelles le classifieur donne son vote si le descripteur est absent

3 Contribution à la tâche 1 : k-plus proches voisins

Cette section décrit une approche différente que nous avons expérimentée pour cette même tâche de classification diachronique. Elle repose sur un apprentissage paresseux, qui se veut plus souple et plus adapté à la tâche.

3.1 Vision de la tâche : *lazy-learning*

Cette approche repose sur un constat. Ces dernières années l'emploi de techniques d'apprentissage (principalement numériques et supervisées) s'est très largement répandu dans le domaine du TAL. Cependant, ces techniques sont souvent utilisées sans tenir compte des spécificités de la tâche à accomplir, des données et du classifieur. Il est ainsi courant de voir utilisées des approches à base de modèle alors que les instances d'une même classe sont connues pour apparaître sous des formes très diverses. Cela oblige à utiliser des classifieurs très complexes à mêmes de construire un unique modèle permettant de prendre en compte ces instances peu comparables entre elles. Dans beaucoup de cas, une approche par plus proches voisins est alors bien plus adaptée.

C'est ce même cas qui se présente dans cette tâche de datation. En effet, deux articles de la même année n'aborde pas forcément les mêmes sujets. Leurs descriptions, si elles sont basées sur les mots qu'ils contiennent, ont peu de chance d'être comparables. Chercher à apprendre un unique modèle sur ces instances est donc inutilement difficile. Une approche par k-plus proches voisins nous a donc paru plus adaptée.

Dans une approche par k-plus proches voisins, une instance inconnue est classée en trouvant les k instances connues les plus similaires et en lui assignant la classe majoritaire de ces instances. Il n'y a donc pas à proprement parler d'apprentissage, d'où le nom de *lazy-learning*, mais l'induction repose sur la calcul de similarité et la mise en œuvre du vote.

3.2 Mesures de similarité

Dans le cas présent, la similarité entre deux articles de presse est simplement calculée en utilisant les mesures classiques utilisées en recherche d'information. Nous avons exploré l'utilisation de deux de ces mesures pour cette tâche. Nous avons tout d'abord implémenté une similarité inspirée de la mesure OKapi BM-25 (Robertson *et al.*, 1998). Celle-ci repose sur une pondération donnée dans l'équation 1 qui indique le poids du terme t dans le document d.

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

où $k_1 = 2$ and $b = 0.75$ sont des constantes, tf le nombre d'occurrences du terme t dans le document d, dl la longueur du document, dl_{avg} la longueur moyenne des documents, N le nombre total de documents et df le nombre de documents contenant le terme t. Les deux parties de l'équation peuvent être interprétés comme un TF et un IDF.

En RI, une pondération spécifique pour le terme dans la requête est utilisée. Dans notre cas, la requête étant aussi un article, nous avons adopté la même pondération TF. Nous avons également donné un poids plus important à l'IDF pour améliorer la précision de l'appariement. Finalement, la similarité entre un article à dater d et un article connu D_{ex} est :

$$sim(d, d_{ex}) = \sum_{t \in d \cap d_{ex}} TF_{BM25}(t,d) * TF_{BM25}(t,d_{ex}) * (IDF_{BM25}(t))^3$$

Une autre mesure de similarités a été testée mais n'a pas fait l'objet d'une soumission de run. Il s'agit d'une mesure basée sur le modèle de langue proposé pour la RI par Hiemstra et Kraaij (Hiemstra & Kraaij, 1999). Cette mesure offre plus de possibilités grâce aux différentes stratégies de lissage qui peuvent être utilisées. Mais elle implique des temps de calculs plus importants puisque même les mots absents des documents doivent être pris en compte. Ces temps de calcul et le nombre limité de runs par équipe ne nous a pas permis d'évaluer ces variantes des modèles de langue à la Hiemstra.

Pour appliquer ces mesures, nous pré-traitons les textes simplement en les étiquetant avec TreeTagger et en ne retenant que les noms communs et propres, verbes et adjectifs. Ce sont donc sur ces termes que sont calculées les similarités. Bien entendu, du fait des erreurs d'OCR, de l'ancien français ou d'anciennes orthographes dans certains articles, l'étiquetage produit lui-même des résultats très bruités. Comme nous l'avons dit précédemment, des pré-traitements sur les textes permettraient d'améliorer considérablement cette représentation et certainement les résultats, mais le manque de temps ne nous a pas permis de les mettre en œuvre.

3.3 Procédure de vote

Dans les runs fournis, les cinquante plus proches voisins ont été retenus. À partir des dix années ainsi collectées à partir de ces voisins, différentes stratégies peuvent être mise en œuvre pour décider de l'année à attribuer à l'article inconnu.

Il est par exemple possible de faire un vote et de garder l'année majoritaire, ou de calculer une moyenne ou une médiane sur les années. Dans notre cas, nous avons implémenté un vote pondéré par le score de similarité. Pour tirer au mieux parti du caractère continu des classes, il est important de faire en sorte que les années proches des années des articles voisins soient également considérées. Pour savoir quels poids donner à ces années voisines, nous utilisons la même fonction gaussienne que celle utilisée pour l'évaluation : l'année n du voisin reçoit un poids de $1 * sim(d, d_{ex})$, les années $n-1$ et $n+1$ reçoivent $0.969 * sim(d, d_{ex})$... Finalement, chacun des cinquante voisins vote donc pour son année de parution et les années connexes, pondéré par le score de similarité entre ce voisin et l'article, et l'année obtenant le poids le plus important est proposé.

3.4 Résultats

Les résultats de cette approche correspondent au run 3 de l'équipe TexMex pour les deux pistes de la tâche. Ces runs se classent premiers pour les deux pistes et les scores obtenus sur cet échantillon de test correspondent aux

évaluations menées en *leave-one-out* durant la phase d'entraînement. Ces résultats témoignent du bien fondé de notre approche, même si une large part à l'amélioration existe.

Outre ces résultats bruts, il est intéressant de noter de cette approche par k-plus-proches voisins est calculatoirement légère. Il n'y a en effet aucune phase d'apprentissage, et l'ajout de nouveaux documents datés peut bénéficier immédiatement aux résultats.

4 Contribution à la tâche 2 : appariement résumé/article

Cette section décrit notre participation à la tâche d'appariement entre résumés et articles scientifiques. Deux pistes étaient proposées, se différenciant par la présence ou non des introductions et des conclusions dans les bases d'article. Nous avons soumis un seul run sur chacune de ces pistes.

4.1 Vision de la tâche

Cette tâche d'appariement apparaît clairement comme une tâche classique de recherche d'information où le résumé joue le rôle de requête. Nous avons donc là encore utilisé une simple approche de calcul de similarité par pondération Okapi-BM25 (cf. supra). Comme précédemment, les documents ont été étiquetés à l'aide de TreeTagger et seuls les noms, verbes et adjectifs ont été conservés pour représenter les documents.

Lors de la phase d'entraînement, l'évaluation a montré d'excellents niveaux de performances, avec une précision de l'ordre de 99 % à 100 %. Nous n'avons pas cherché à améliorer ces résultats, étant donné le peu d'intérêt pour une tâche si simple et le manque de temps. En particulier, aucune adjudication n'a été faite ; un même article peut donc avoir été assigné à différents résumés.

4.2 Résultats

Les runs soumis à la piste 1 (articles complets) et à la piste 2 (articles sans les introductions et conclusions) obtiennent respectivement des précisions de 99.5 % et 99 %. Ces résultats correspondent bien aux évaluations que nous avons effectuées sur le training set et soulignent encore une fois la facilité de cette tâche.

5 Conclusion

Cette première participation de l'IRISA à DEFT se traduit donc par de bons résultats, malgré un calendrier trop serré pour réellement permettre le développement de méthodes innovantes. Ces bons résultats sont donc le fruit de l'emploi de techniques classiques, mais choisies de manière à être bien adaptées aux tâches et aux données.

Les deux tâches proposées sont très différentes par leur niveaux de difficultés. La tâche de datation mêle en effet plusieurs niveaux de difficultés (données bruitées, classification en classes continues, diversité des exemples,...) qui peut rendre l'analyse des résultats difficile. À l'inverse, la trop grande simplicité de la tâche d'appariement fait que les équipes ont toute obtenu des niveaux de performances comparables proche de la perfection.

En revanche, ces deux tâches relevaient bien du même paradigme de comparaison de documents, paradigme plus proche de la recherche d'information que de la fouille de données. Nos runs les plus efficaces se sont donc inspirés des techniques de RI pour mener à bien ces deux tâches.

Références

HIEMSTRA D. & KRAAIJ W. (1999). Twenty-one at trec-7 : ad-hoc and cross-language track. In *Proceedings of the 7th Text Retrieval Conference TREC-7, NIST Special Publication 500-242*, p. 227–238.

OGER S., ROUVIER M., CAMELIN N., KESSLER R., LEFÈVRE F. & TORRES-MORENO J. M. (2010). Système du lia pour la campagne deft'10 : datation et localisation d'articles de presse francophones. In *DEFT'10*.

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

ROBERTSON S. E., WALKER S. & HANCOCK-BEAULIEU M. (1998). Okapi at TREC-7 : Automatic Ad Hoc, Filtering, VLC and Interactive. In *Proceedings of the 7th Text Retrieval Conference, TREC-7*, p. 199–210.

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

Méthodes pour l'archéologie linguistique : datation par combinaison d'indices temporels

Anne Garcia-Fernandez[1] Anne-Laure Ligozat[1, 2] Marco Dinarelli[1] Delphine Bernhard[1]
(1) LIMSI-CNRS, BP133, 91403 Orsay cedex
(2) ENSIIE, 1 square de la résistance, 91000 Évry
{annegf,annlor,marcod,bernhard}@limsi.fr

Résumé. Dans cet article, nous présentons notre participation au défi fouille de texte (DEFT) 2011 à la tâche de datation d'un document. Notre approche est fondée sur une combinaison de plusieurs sous-systèmes, certains supervisés, d'autres non supervisés, et utilise plusieurs ressources externes comme Wikipédia, les Google Books n-grams ainsi que des connaissances sur les réformes orthographiques du français. Notre meilleur système obtient un score de 0,378 sur les portions de 300 mots et de 0,452 sur les portions de 500 mots, ce qui représente 37% de décennies correctes et 10% d'années correctes au premier rang sur les portions de 300 mots, et 42% de décennies correctes et 14% d'années correctes au premier rang sur les portions de 500 mots.

Abstract. In this article, we present a method for automatically determining the publication dates of documents, which was evaluated on a French newspaper corpus in the context of the DEFT 2011 evaluation campaign. Our approach is based on a combination of different sub-systems, both supervised and unsupervised, and uses several external resources, e.g. Wikipedia, Google Books n-grams, and etymological background knowledge about the French language. Our best system obtains a score of 0.378 on 300 words portions and 0.452 on 500 words portions. This represents 37% of correct decades and 10% of correct years at first rank on 300 words portions, and 42% of correct decades and 14% of correct years at first rank of 500 words portions.

Mots-clés : Analyse diachronique, classification de documents, apprentissage supervisé.

Keywords: Diachronic analysis, document classification supervised learning.

1 Introduction

En 2011, le DÉfi Fouille de Texte, DEFT, a proposé deux tâches. La première, à laquelle nous avons participé, consiste à identifier l'année d'extraits d'articles de journaux. Cette tâche s'inscrit dans la continuité de l'édition 2010 qui proposait notamment d'identifier la décennie d'un extrait de document. Pour déterminer automatiquement l'année d'un texte, nous avons choisi d'utiliser différentes méthodes et ressources, puis de les combiner. Dans cet article, nous présentons les méthodes que nous avons utilisées et les résultats que nous avons obtenus.

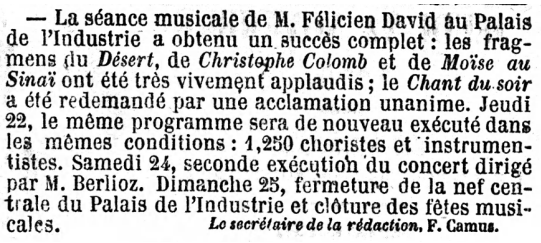
Une première section présente le corpus et les pré-traitements que nous avons effectués. La section suivante décrit de façon générale notre approche puis nous présentons les deux types d'approches que nous avons mis en œuvre. Dans la section 4, nous présentons des approches à base de ressources externes et indépendantes du corpus que nous nommons *méthodes chronologiques*. Dans la section 5, nous exposons les approches par apprentissage dites *méthodes de similarité temporelle*. Nous présentons les résultats en terme de F-mesure (telle que définie par DEFT et présentée dans (Grouin *et al.*, 2011)) et en terme de pourcentage d'années et de décennies identifiées correctement.

2 Corpus et prétraitements

Le corpus, décrit dans (Grouin *et al.*, 2011), est composé d'article de journaux issus de la base de donnée Gallica [1]. Il s'agit d'extraits d'articles publiés entre 1801 et 1944 de 300 ou 500 mots. Ces documents sont issus de la numérisation de journaux papiers (figure 2) et contiennent, comme nous pouvons le voir dans la figure 1, de nombreuses erreurs provenant du processus de reconnaissance optique des caractères (OCR).

La séance musicale de M. Félicien David au Palais de l'Industrie a obtenu un succès complet les fragmens du Désert, de Christophe Colomb et de Moïse au Sinaï ont été très vivemçnt applaudis ; le Chant du soir a été redemandé par une acclamation unanime. Jeudi 22, le même programme sera de nouveau exécuté dans les mêmes conditions : l,2S0 choristes et'instrumentistes. Samedi 24, seconde exécution du concert dirigé par M. Berlioz. Dimanche 2S, fermeture de la nef centrale du Palais de l'Industrie et clôture des fêtes musicales. Lotecfètairedela rédaction, F. Carani.

FIGURE 1: Version électronique d'un extrait de journal datant de 1855



— La séance musicale de M. Félicien David au Palais de l'Industrie a obtenu un succès complet : les fragmens du *Désert*, de *Christophe Colomb* et de *Moïse au Sinaï* ont été très vivement applaudis ; le *Chant du soir* a été redemandé par une acclamation unanime. Jeudi 22, le même programme sera de nouveau exécuté dans les mêmes conditions : 1,250 choristes et instrumentistes. Samedi 24, seconde exécution du concert dirigé par M. Berlioz. Dimanche 25, fermeture de la nef centrale du Palais de l'Industrie et clôture des fêtes musicales. *Le secrétaire de la rédaction*, F. Camus.

FIGURE 2: Image du même extrait de journal datant de 1855

2.1 Découpage du corpus d'entraînement

Le corpus de développement fourni par DEFT contient 3 596 portions de documents. Nous avons divisé ce corpus en deux parties : un ensemble d'entraînement (TRN) constitué de 2 396 portions et un ensemble de validation (DEV) constitué de 1 200 portions.

2.2 Lemmatisation

Afin de réduire la taille du vocabulaire des corpus, nous avons remplacé les mots par leurs lemmes, en appliquant le TreeTagger (Schmid, 1994). Ceci nous a permis de passer, pour le corpus TRN, d'un vocabulaire de 74 000 mots à un vocabulaire de 52 000 mots.

1. http://gallica.bnf.fr/

3 Description générale de l'approche

Notre approche est fondée sur l'utilisation de deux types de méthodes. Les méthodes dites chronologiques s'appuient sur des ressources externes et indépendantes du corpus d'entrainement fourni. Les méthodes de similarité temporelle sont quant à elle des approches par apprentissage fondées sur l'utilisation de la similarité cosinus et de modèles SVM.

4 Méthodes chronologiques

Les méthodes chronologiques ont pour objectif de déterminer des périodes de temps qui sont les plus probables pour chaque portion. Elle ne permettent pas d'estimer l'année précise de publication d'une portion.

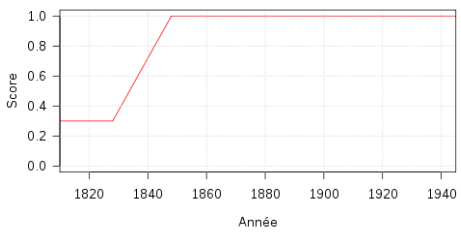
4.1 Dates de naissance de personnes

La présence d'un nom de personne peut être un indice de la date d'un texte (Albert *et al.*, 2010), puisque le document est nécessairement postérieur à l'année de naissance de cette personne. Afin d'utiliser cette information, nous souhaitions reconnaître les noms de personnes dans nos corpus, puis aller chercher leurs dates de naissance dans Wikipédia.

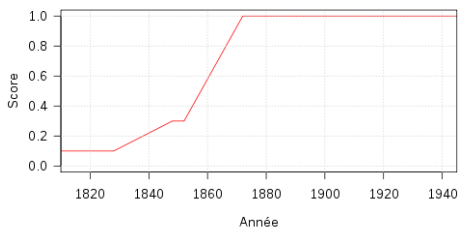
Nous avons dans un premier temps essayé d'appliquer un système de reconnaissance d'entités nommées au corpus d'entraînement (TRN), mais les résultats n'étaient pas suffisamment fiables. Nous avons donc employé une stratégie différente : nous avons collecté de façon automatique les années de naissance de personnes nées entre 1781 et 1944 en utilisant les catégories Wikipédia «Naissance_en_AAAA», qui regroupent des personnes étant nées à une année donnée et présentes dans Wikipédia. Nous avons ainsi recueilli environ 99 000 noms de personnes associés à leurs années de naissance, à partir desquelles nous en avons sélectionné 96 000 non ambiguës (par exemple, deux «Albert Kahn» avaient été trouvés), puisque nous n'avons pas de moyen simple de savoir de quelle personne il s'agit précisément.

Nous avons ensuite appliqué WMatch, un moteur d'expressions régulières [2] permettant notamment une annotation rapide de textes (Rosset *et al.*, 2008; Galibert, 2009), à chaque portion, afin de détecter la présence de ces noms de personnes dans nos corpus. Pour le corpus TRN, 529 noms de personnes ont été trouvés (concernant 375 portions), dont 16 (3%) correspondaient en réalité à des homonymes ou des annotations erronées : par exemple, Wikipédia a une entrée pour la romancière Colette, qui est par ailleurs un prénom, ce qui donnait lieu à des ambiguïtés.

Un score a ensuite été donné à chaque portion de chaque année possible, en fonction de la présence de noms de personnes. La figure 3 montre les scores obtenus dans le cas de la présence de deux noms de personnes, Jules Verne, né en 1828, et Antoni Gaudì, né en 1852.



(a) avec Jules Verne, né en 1828



(b) avec Jules Verne, né en 1828 et Antoni Gaudí, né en 1852

FIGURE 3: Scores en fonction de la présence de noms de personnes

Nous partons du principe qu'une portion citant une personne n'a pas pu être publiée avant la naissance de cette

2. Disponible sur demande.

personne. Le score de cette portion pour toutes les années précédant l'année de naissance de la personne a été fixé à 0,3. Le score augmente au fur et à mesure jusqu'à atteindre 1 vingt ans après l'année de naissance de la personne. Si plusieurs personnes sont citées dans un même extrait, les scores pour chaque personne et pour chaque année sont multipliés entre eux.

4.2 Néologismes et archaïsmes

Les néologismes correspondent à des mots nouvellement créés, tandis que les archaïsmes sont des mots qui ont cessé d'être utilisés à un certain moment et qui ne sont donc plus d'usage. Les deux phénomènes constituent des indices utiles pour déterminer la date de publication d'un document : s'il contient un néologisme, une probabilité très faible peut être attribuée aux années précédant la date d'apparition du néologisme, l'inverse étant vrai pour les archaïsmes. Cependant, les dates d'apparition et de disparition des mots ne sont pas des informations aisément accessibles. Nous avons donc développé une méthode pour extraire automatiquement des néologismes et des archaïsmes à partir des données de Google Books pour le français [3].

4.2.1 Acquisition automatique de néologismes et d'archaïsmes

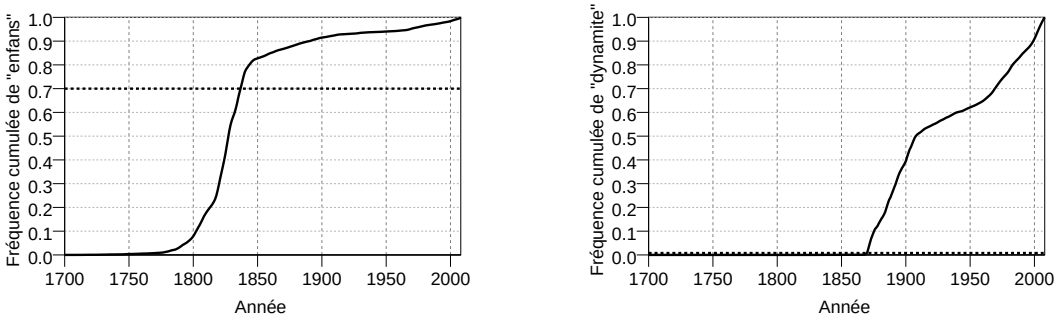
L'acquisition des dates d'apparition ou de disparition des mots n'est pas une tâche triviale. En effet, les métadonnées associées à Google Books ne sont pas toujours précises (Nunberg, 2009). Il n'est donc pas possible d'utiliser un critère simple comme l'extraction de la première année d'occurrence du mot pour identifier les néologismes.

Nous avons donc appliqué la méthode suivante, qui se fonde sur la distribution des fréquences cumulées :

1. recueil de la distribution des occurrences du mot entre les années 1700 et 2008 [4] ;
2. lissage de la distribution avec une fenêtre de 3 années ;
3. calcul de la distribution des fréquences cumulées et extraction de la date d'apparition/disparition, correspondant à l'année à laquelle la fréquence cumulée dépasse un certain seuil.

Nous avons défini les meilleurs seuils en utilisant 32 néologismes (par exemple «photographie» ou «télévision») et 21 archaïsmes (anciennes orthographes, voir section 4.3). Les seuils ainsi obtenus sont 0,008 pour les néologismes et 0,7 pour les archaïsmes. De plus, nous n'avons conservé que les néologismes ayant un nombre d'occurrences moyen par année supérieur à 10 et les archaïsmes ayant un nombre d'occurrences moyen supérieur à 5 pour les années considérées. Nous avons ainsi extrait 34 396 néologismes et 53 392 archaïsmes avec leur date d'apparition/disparition.

La figure 4 présente deux courbes de fréquences cumulées : l'une pour un archaïsme (l'orthographe ancienne «enfans»), et l'autre pour le néologisme «dynamite» (inventée en 1867).



(a) Archaïsme

(b) Néologisme

FIGURE 4: Fréquences cumulées

3. Disponibles à l'adresse suivante : `http://ngrams.googlelabs.com/datasets`

4. La première année disponible dans les Google Books n-grams est en réalité 1536, mais étant données les années qui nous intéressent ici, nous avons considéré que 1700 était un seuil adéquat. En outre, les données pour les 16e et 17e siècle sont trop peu nombreuses pour constituer des données fiables pour ce type de traitement.

Le seuil correspond aux lignes horizontales en pointillés. Ces courbes ont des profils très différents : les archaïsmes sont caractérisés par une fonction logistique, qui atteint un plateau bien avant la fin de la période considérée ; les néologismes correspondent quant à eux à des courbes exponentielles.

Nous avons calculé le taux d'erreur sur le corpus d'entraînement : pour 90% des archaïsmes trouvés dans le corpus, la date de la portion est bien antérieure à la date de disparition, et, si l'on suppose que le mot peut encore être utilisé jusqu'à 20 ans après la date de disparition calculée, la date de la portion est antérieure à la date de disparition plus 20 ans pour 97% des archaïsmes. Pour les néologismes, la date de la portion est postérieure à la date d'apparition dans 97% des cas, et à la date d'apparition moins 20 ans dans 99,8% des cas.

4.2.2 Score attribué par les néologismes et archaïsmes

Les listes de néologismes et archaïsmes établies ont été utilisées pour attribuer un score à chaque année pour chaque portion. Pour les néologismes, un score élevé a été attribué aux années postérieures à la date d'apparition, et un score faible pour les années la précédant. La formule utilisée pour les néologismes est la suivante, avec p la portion de texte, w un mot, y une année dans la période considérée 1801-1944 et $année(w)$ la date d'apparition extraite pour un néologisme :

$$score_{néo}(p,y) = \frac{\sum_{w \in t} score_{néo}(w,y)}{|p|} \text{ avec :}$$

$$score_{néo}(w,y) = \begin{cases} 1 & \text{si } w \notin \text{néologismes} \\ 1 & \text{si } w \in \text{néologismes et } y \geq année(w) \\ 0,2 & \text{si } w \in \text{néologismes et } (année(w) - y) > 20 \\ 0,2 + 0,04 \cdot (20 + y - année(w)) & \text{sinon} \end{cases}$$

Une formule équivalente est utilisée pour les archaïsmes : dans ce cas les années suivant la date de disparition d'un mot ont un score faible.

4.3 Réformes orthographiques

Pendant la période 1801-1944, le français a connu deux réformes orthographiques majeures : la première en 1835 et la seconde en 1878. Le principal changement introduit par la première concerne certaines conjugaisons finissant par «oi», qui deviennent «ai» (par exemple pour le verbe «avoir», «avois» devient «avais»). La seconde réforme concerne principalement les noms finissant par «ant» ou «ent» dont le pluriel devient «ants»/«ents» au lieu de «ans»/«ens» (par exemple, «enfans» devient «enfants»).

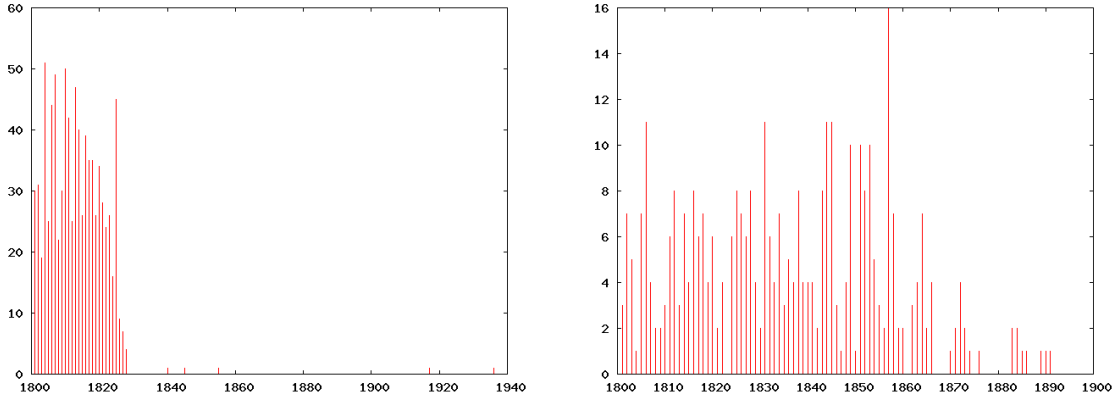
À la suite de (Albert *et al.*, 2010), nous avons utilisé ces informations. La figure 5 montre la distribution de chaque type de mots dans le corpus d'entraînement TRN pour chaque année. Les mots ayant été modifiés par la première réforme sont bien présents principalement avant 1828, et ceux modifiés par la deuxième réforme sont présents uniquement avant 1891.

4.3.1 Scores attribués grâce aux réformes orthographiques

Nous avons utilisé les informations fournies par les réformes de la même façon que (Albert *et al.*, 2010) : nous avons attribué un score à chaque année pour chaque portion de texte. Afin de détecter les anciennes orthographes dans nos textes, nous avons utilisé la méthode suivante :

- recueil des mots inconnus avec hunspell [5]. Puis pour chaque mot inconnu :
- si le mot finit par «ois/oit/oient», remplacer «o» par «a» ;
  - si le mot ainsi formé est dans le dictionnaire, incrémenter le compteur n_{28}, qui correspond au nombre de mots dont l'orthographe a été modifiée par la première réforme ;
- si le mot finit par «ans/ens», insérer «t» avant «s» ;
  - si le nouveau mot est dans le dictionnaire, incrémenter le compteur n_{91}, qui correspond au nombre de mots dont l'orthographe a été modifiée par la deuxième réforme ;

5. Hunspell a été utilisé avec le dictionnaire DELA pour le français (Blandine & Silberzstein, 1990)



(a) Mots terminant par «oi»

(b) Mots terminant par «a/ens»

FIGURE 5: Distribution des mots modifiés par les réformes orthographiques dans le corpus d'entraînement (TRN)

Une fonction permet de déterminer un score pour chaque année y, à partir des compteurs n_{28} et n_{91} selon les formules suivantes :

$score_{ortho}(p, y) = score_{28}(p, y) \cdot score_{91}(p, y)$ avec :

$$score_r(p,y) = \begin{cases} f_r(p,y) & si\ y > r \\ 1 & si\ y \leq r \end{cases} \quad , \quad f_{28}(p,y) = \begin{cases} 1 & si\ n_{28}(p) = 0 \\ 0,15 & si\ n_{28}(p) = 1 \\ 0 & si\ n_{28}(p) > 1 \end{cases} \quad et \quad f_{91}(y) = \begin{cases} 1 & if\ n_{91}(p) = 0 \\ 0 & if\ n_{91}(p) > 0 \end{cases}$$

Ainsi, si $n_{28} = 1$ et $n_{91} = 1$ pour une portion de texte, le score pour les années antérieurs à 1828 est de 1, puis de 0,15 pour les années comprises entre 1828 et 1892, ce qui correspond au taux d'erreur pour ce critère dans notre corpus d'entraînement, et de 0 pour les années postérieures à 1891, puisque la présence d'une orthographe modifiée par la deuxième réforme est un indicateur très fiable d'une date de publication antérieure à 1891.

5 Méthodes de similarité temporelle

Les méthodes de similarité temporelle calculent des similarités entre chaque portion de texte et un corpus de référence. Elles permettent de dater précisément une portion, mais sont sujettes aux erreurs, qui devraient être en partie corrigées par les méthodes chronologiques. Notre intuition est que les informations apportées par les deux types de méthodes sont complémentaires.

5.1 Similarité cosinus

5.1.1 Corpus de documents comme référence

Le corpus d'entraînement fournit des exemples de textes pour chaque année dans la période 1801-1944. Ces textes peuvent être utilisés comme référence pour obtenir des statistiques temporelles. Nous avons donc regroupé toutes les portions d'une même année du corpus d'entraînement et utilisé ces groupes de portions comme référence pour les années correspondantes. Chaque groupe et chaque portion testée ont été indexés par le tf·idf, selon la formule suivante pour le mot i et le document j :

$$tf \cdot idf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \cdot \log \frac{|Y|}{|\{y_j : w_i \in y_j + smoothing\}|}$$

avec $|Y|$ le nombre d'années dans le corpus d'entraînement, y_j le groupe de portions de texte pour l'année j, $n_{i,j}$ le nombre d'occurrences d'un mot w_i dans le document j. $smoothing$ est un lissage appliqué pour tenir compte

des mots du corpus d'évaluation qui n'étaient pas dans le corpus d'entraînement.

Pour une portion de texte du corpus d'évaluation, nous avons calculé les similarités entre la portion et chaque groupe représentant une année grâce à une mesure cosinus. Nous avons essayé de calculer cette similarité pour des n-grams de mots, n allant de 1 à 5 ; cependant, pour n>2, la taille du corpus d'entraînement ne permet pas d'obtenir de bons résultats. Nous avons utilisé pour ces n-grams la version lemmatisée des corpus.

Puisque le corpus est composé de documents numérisés automatiquement, de nombreuses erreurs sont présentes dans les textes, ce qui pose problème pour le tf.idf : les tf et df sont plus petits que ce qu'ils devraient être pour des mots réels, puisque des erreurs de numérisation empêchent l'identification de certaines occurrences, tandis que des mots erronés ont des idf élevés, puisque nécessairement peu présents dans le corpus. Afin de prendre en compte cette particularité, nous avons également calculé la similarité en utilisant des n-grams de caractères (comme par exemple (Naji *et al.*, 2011) pour la recherche d'information dans des corpus numérisés). Ainsi, par exemple, pour le texte «sympathie1» qui contient une erreur de numérisation, les n-grams de caractères (pour n<9) correspondront en grande partie à ceux du mot «sympathie» malgré l'erreur de numérisation.

5.1.2 Google n-grams comme référence

Le corpus d'entraînement étant relativement petit, nous avons également tenté d'utiliser les Google n-grams comme données d'entraînement. Pour des raisons de temps de calcul, nous avons utilisé uniquement les n-grams ayant un contenu alphanumérique et ayant plus de 10 occurrences pour une année donnée. Les données ainsi créées ont été utilisées à la place du corpus d'entraînement. La formule du tf.idf est alors légèrement modifiée pour le corpus d'entraînement, puisque $n_{i,j}$ devient le nombre d'occurrences du n-gram w_i pour l'année j et y_j correspond aux données des Google n-gram pour l'année j.

5.2 Système SVM

Les SVM sont des algorithmes d'apprentissage automatique largement utilisés en TAL, appartenant à la classe des classifieurs à marge maximale (Vapnik, 1998). Nous avons utilisé l'implémentation `svm-light` [6] (Joachims, 1999). Nous avons utilisé deux fonctions noyaux dans les SVM : la `fonction à base radiale` et le `noyau polynomial` toutes deux disponibles dans le logiciel `svm-light`. Étant donnée la faible quantité de données disponibles pour chaque année (25 portions par année, sauf pour 1815, pour laquelle nous disposons de 21 portions), l'approche d'entraînement `un-contre-tous` a été utilisée, et non pas `un-contre-un`. Le système SVM est ainsi composé de 144 modèles binaires, un pour chaque année entre 1801 et 1944. Dans chaque modèle, les instances positives sont celles extraites des portions appartenant à l'année à détecter, et les instances négatives sont les autres. Chaque modèle reconnaît ainsi les portions appartenant à l'année qu'il décrit. Au moment de la classification, chaque portion est évaluée par les 144 modèles et celui donnant le meilleur score est sélectionné.

Les paramètres et jeux d'attributs ont été réglés sur les corpus d'entraînement (TRN) et de validation (DEV) décrits plus hauts. Nous n'avons pas paramétré tous les paramètres ni tous les types d'attributs, ce qui aurait requis un trop grand nombre d'expériences, mais utilisé notre expérience pour en régler certains. Pour les autres nous avons utilisé les valeurs par défaut. Le paramètre de compromis C a été fixé à 1. Dans la plupart des tâches, sa valeur optimale est entre 1 et 10, 1 donnant toujours de bons résultats. Le paramètre de coût, qui affecte le poids des erreurs faites sur les instances positives et négatives, a été fixé au ratio entre le nombre d'instances négatives et positives, comme suggéré dans (Morik *et al.*, 1999). En ce qui concerne les fonctions noyau, nous avons testé les `fonctions à base radiale` et le `noyau polynomial`. Cette dernière était plus efficace sur l'ensemble DEV et a donc été conservée. Les valeurs par défaut ont été utilisées pour les paramètres du noyau polynomial (1 pour c et 3 pour le degré polynomial d).

Pour les attributs, nous avons effectué plusieurs expériences et conservé le jeu donnant le meilleur résultat sur DEV. Nous avons tout d'abord essayé les configurations courantes dans des tâches de classification de textes. Par exemple, nous avons supprimé les mots vides et remplacé les mots par leurs lemmes. Cette configuration dégradait les performances. En revanche, en gardant les mots vides et en utilisant à la fois les mots et leurs lemmes, nous obtenions de meilleurs résultats qu'avec les mots seuls. Cette configuration a donc été choisie comme système

6. Disponible à l'addresse `http://svmlight.joachims.org/`

SVM de base. Nous avons également testé l'utilisation de n-grams pour n allant de 1 à 4 et les 2-grams ont donné les meilleurs résultats.

À partir de cette configuration, nous avons intégré les informations founies par les systèmes décrits dans la section 4, c'est-à-dire les années de naissance des personnes, les néologismes et les archaïsmes, et les réformes orthographiques. Chacun de ces systèmes a fourni des informations pour le SVM sous la forme de vecteurs $attribut : année$, où $attribut$ est un nom de personne dans le cas des dates de naissance, un néologisme ou un archaïsme ou un mot qui a été affecté par l'une des réformes orthographiques. Cette forme a posé problème pour le SVM, même en représentant les années dans l'intervalle 1..144 au lieu de 1801..1944 : les performances étaient moins bonnes en utilisant cette représentation. Nous avons donc changé le mode de représentation, et représenté l'information fournie par les méthodes chronologiques par une information binaire (0 si l'attribut est absent, 1 si il est présent) : un attribut code l'information elle-même, c'est-à-dire par exemple la présence d'un néologisme particulier, et un autre attribut code l'année associée à cette information. Cette représentation a permis d'améliorer nos résultats.

Puisque dans nos expériences préliminaires le comportement des systèmes était identique sur les portions de 500 mots et sur les portions de 300 mots, nous avons mené nos expériences uniquement sur celles de 300, et avons appliqué la meilleure configuration observée sur celles de 300 pour les portions de 500.

6 Combinaison des scores

Nous avons ensuite effectué une combinaison des scores définis par les méthodes décrites précédemment. Les différentes méthodes fournissent en effet des informations complémentaires : par exemple, les archaïsmes indiquent une limite haute à la date de publication, alors que la similarité cosinus va donner des années probables de publication. Pour la combinaison des scores, nous avons utilisé deux stratégies : la multiplication de tous les scores et la régression linéaire.

6.1 Multiplication des scores

La méthode la plus simple de combinaison était la multiplication des scores, qui sont tous entre 0 et 1, 0 indiquant une probabilité nulle pour une année donnée, et 1 indiquant une probabilité maximale. Le score final est donc la multiplication des scores précédents :

$$score_{multiplication}(p, y) = \prod_k score_k(p, y)$$

avec $score_k(p, y)$ le score du système k pour la portion p et l'année y.

6.2 Régression linéaire

Dans ce cas, les scores des différents systèmes ne sont pas multipliés mais additionnés avec des coefficients de pondération, selon la formule suivante :

$$score_{régression}(p, y) = \sum_k \alpha_k \cdot score_k(p, y) + \varepsilon$$

avec α_k coefficient pour le système k, $score_k(p, y)$ le score donné par le système k à la portion p pour l'année y et ε le terme d'erreur.

Les coefficients ont été calculés sur le corpus d'entraînement en utilisant la fonction R `lm()`. Le processus de régression linéaire trouve le meilleur modèle (déterminé par les coefficients α pour prédire une valeur numérique à partir de plusieurs indices ici, les scores des systèmes). Dans notre cas, la valeur numérique à prédire dépend de la distance $dist$ entre une année et l'année réelle de publication de la portion : la valeur est $1 - dist/143$.

Dans la phase de développement, nous avons réglé les valeurs de α et ε sur le corpus d'entraînement TRN et testé la combinaison sur le corpus DEV. Comme le cosinus et les SVM avaient besoin d'une phase d'entraînement,

nous n'avons pas inclus les scores de ces systèmes dans notre modèle de régression. Nous avons donc calculé un score de régression à partir des scores donnés par les néologismes, archaïsmes, années de naissance et réformes orthographiques. Les scores du cosinus et des SVM ont ensuite été multipliés par ce score de régression. Pour la phrase d'évaluation, nous avons réglé les paramètres sur le corpus de développement complet.

6.3 Pondération

La régression linéaire permet de pondérer les scores obtenus par les méthodes chronologiques (néologismes, archaïsmes, dates de naissance et réformes orthographiques) mais ne permet pas de pondérer les résultats fournis par les méthodes de similarité temporelle (similarité cosinus et SVM). Nous avons donc utilisé une pondération basée sur la formule suivante :

$$score_{ponderé}(p, y) = \beta \cdot score_{régression}(p, y) + (1 - \beta) \cdot score_{cosinus} \cdot score_{SVM}$$

avec β un coefficient, déterminé à partir du corpus d'apprentissage et donnant alors les meilleurs résultats, de 0,015.

7 Résultats

7.1 Score

Nous avons évalué nos systèmes en utilisant le score proposé par DEFT, qui prend en compte la distance entre l'année prédite et l'année réelle de publication.

Étant donnée une portion de texte a_i dont l'année de publication de référence est $d_r\left(a_i\right)$, un système donne une estimation de l'année $d_p\left(a_i\right)$. Le système reçoit alors un score qui dépend de la distance entre l'année prédite et l'année de référence. Le score est basé sur une fonction gaussienne et est moyenné sur les N portions de test. La formule précise est la suivante :

$$S = \frac{1}{N} \sum_{i=1}^{N} e^{-\frac{\pi}{10^2}\left(d_p(a_i) - d_r(a_i)\right)^2} \tag{1}$$

Nous présenterons d'abord les résultats du cosinus et des SVM, puis deux des combinaisons. Les systèmes utilisés pour les données d'évaluation ont été entraînés sur le corpus de développement complet (TRN + DEV).

7.2 Résultats pour les méthodes de similarité temporelle

7.2.1 Similarité cosinus

Les résultats de la similarité cosinus sont présentés dans les tableaux 1 et 2 (seuls les meilleurs systèmes sont présentés). Nous pouvons voir que les meilleurs résultats sont obtenus en utilisant les n-grams de 5 caractères, ce qui était attendu du fait du bruit dans nos données. Les 1-grams de mots sont meilleurs sur les portions de 300 mots que les 2-grams, mais c'est le contraire sur les portions de 500 mots, ce qui peut s'expliquer par le fait que les 2-grams sont meilleurs avec plus de données d'entraînement.

Pour le cosinus utilisant les Google n-grams, le corpus a été utilisé dans sa version non lemmatisée, puisque les Google n-grams contiennent des mots fléchis. Les meilleurs résultats sont obtenus avec les 2-grams, mais sont inférieurs à ceux obtenus avec le corpus d'entraînement. Ceci est étonnant car les Google n-grams représentent un corpus bien plus grand, mais peut peut-être s'expliquer par la différence de nature des documents, puisque notre corpus est composé uniquement d'extraits de journaux. En outre, la datation des Google Books n'est pas toujours fiable (Nunberg, 2009).

	Corpus d'entraînement (DEV)		Corpus d'évaluation	
	300 mots	**500 mots**	**300 mots**	**500 mots**
1-grams de mots	0,260	0,299	0,267	0,321
2-grams de mots	0,209	0,319	0,263	0,327
5-grams de caractères	0,287	0,327	0,311	0,363

TABLE 1: Résultats obtenus par la méthode fondée sur le cosinus

	Corpus d'entraînement (DEV)		Corpus d'évaluation	
	300 mots	**500 mots**	**300 mots**	**500 mots**
1-grams de mots	0,210	0,221	0,200	0,216
2-grams de mots	0,238	0,295	0,241	0,264

TABLE 2: Résultats obtenus par la méthode fondée sur le cosinus avec les Google n-grams

7.2.2 Système SVM

Les résultats obtenus avec le système fondé sur les SVM sont donnés dans les tableaux 3 et 4. Comme nous pouvons le voir dans le tableau 3, l'ajout incrémental d'attributs encodant l'information donnée par les méthodes chronologiques améliore les résultats.

	Corpus d'entraînement (DEV)
	300 mots
Baseline (2-grams mots + lemmes)	0,228
+néologismes	0,234
+réformes orthographiques	0,242
+années de naissance	0,243

TABLE 3: Résultats additifs du système SVM sur les portions de 300 mots avec différents types d'attributs

7.2.3 Combinaison des scores

Le tableau 5 présente les résultats obtenus sur les corpus d'entraînement et de test pour les meilleurs systèmes.

La combinaison des systèmes améliore nettement les scores des systèmes individuels. Les résultats sur les portions de 500 mots sont meilleurs que ceux sur les portions de 300 mots, ce qui était attendu puisque les portions de 500 mots présentent plus d'indices de leur date de publication. Globalement, la combinaison par multiplication obtient de meilleurs scores que la régression linéaire. La figure 6 montre les résultats en termes d'année et de décennie correcte au premier rang. Nos systèmes obtiennent environ 35% de décennie correcte au premier rang pour les portions de 300 mots, et 40% pour celles de 500 mots. Pour les années, la régression linéaire détecte la bonne année pour 10% des portions de 300 mots et 14% des portions de 500 mots. Ces résultats sont bien au-dessus du hasard (7% pour les décennies et 0,7% pour les années), et sont meilleurs pour les décennies que ceux du challenge DEFT 2010 (Grouin *et al.*, 2010).

7.2.4 Soumissions à DEFT 2011

Nous avons soumis trois résultats lors de la campagne DEFT 2011. Concernant les portions de 300 mots, nous avons soumis les résultats provenant de la combinaison par multiplication, par régression linéaire et par pondération, systèmes ayant donné les meilleurs résultats sur le corpus d'entraînement. Concernant les portions de 500 mots, nous avons, lors des 3 jours de tests, eu une difficulté d'adaptation du système de combinaison par pondération et n'avons donc pas soumis les résultats de ce système. Nous avons ainsi soumis, pour les portions de

Corpus d'entraînement (DEV)		Corpus d'évaluation	
300 mots	**500 mots**	**300 mots**	**500 mots**
0,243	0,293	0,272	0,330

TABLE 4: Résultats du système SVM

	Entraînement (DEV)		Évaluation	
	300 mots	**500 mots**	**300 mots**	**500 mots**
multiplication	0,343	0,401	0,378	0,452
régression	0,356	0,390	0,374	0,428
pondération	0,319	0,425	0,358	0,384

TABLE 5: Résultats obtenus avec la combinaison des scores

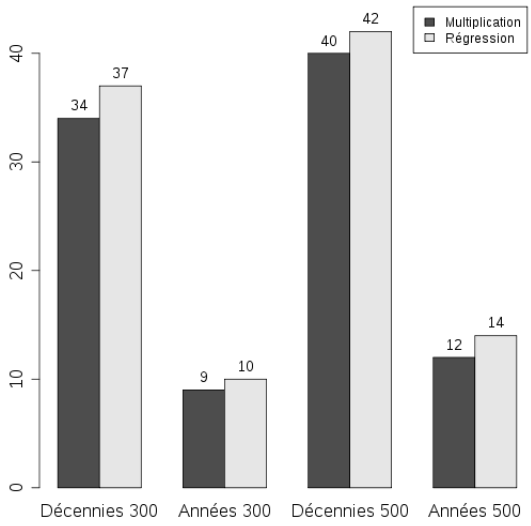


FIGURE 6: Pourcentage de décennies et d'années correctes au premier rang

500 mots, les résultats provenant de la combinaison par multiplication, par régression linéaire ainsi que le résultat fournis par la similarité cosinus calculée sur des 5-grams de caractères.

8 Conclusion

Nous avons présenté une approche pour la datation automatique de documents historiques. Celle-ci est fondée sur différentes méthodes et cherche à tirer avantage de chacune d'elles afin d'estimer au mieux l'année de publication d'extraits de journaux. Notre meilleur système a obtenu une F-mesure de 0,452 sur des portions de 500 mots et 0,378 sur des portions de 300 mots. Les résultats montrent l'importance d'utiliser des données externes qui prennent en compte l'évolution diachronique d'une langue associées à des techniques de fouille de texte par apprentissage.

Cette tâche reste un défi puisque notre meilleur système permet d'estimer correctement pour des portions de 500 mots *seulement* 40% de décennies et 14% d'années. La difficulté d'une telle tâche vient d'une part de la qualité des documents, puisqu'ils sont numérisés (mais ceci fait aussi de la tâche une application proche d'un besoin réel), d'autre part de la faible quantité de données de référence à notre disposition.

Cette tâche nous a permis de confronter des techniques *classiques* utilisées en traitement automatique des langues à des données bruitées par une numérisation (qui rendent très difficile la tâche d'un détecteur d'entités nommées habituellement performant, par exemple) et ainsi de développer des techniques adaptées comme l'utilisation de n-grams de caractères. Il serait bien sûr intéressant de préalablement corriger orthographiquement les corpus.

Remerciements

Ce travail a été partiellement financé par le projet Quæro (financement Oseo, agence française pour l'innovation et la recherche) et le projet DOXA du pôle de compétitivité CAP-DIGITAL.

Références

ALBERT P., BADIN F., DELORME M., DEVOS N., PAPAZOGLOU S. & SIMARD J. (2010). Décennie d'un article de journal par analyse statistique et lexicale. In *Atelier DEFT, Actes TALN 2010*.

BLANDINE C. & SILBERZSTEIN M. (1990). Dictionnaires électroniques du français. *Langue française*, **87**.

GALIBERT O. (2009). *Approches et méthodologies pour la réponse automatique à des questions adaptées à un cadre interactif en domaine ouvert*. PhD thesis, Thèse de doctorat en informatique, Université Paris-Sud 11, Orsay, France.

GROUIN C., FOREST D., PAROUBEK P. & ZWEIGENBAUM P. (2011). Présentation et résultats du défi fouille de texte DEFT2011. In *Atelier DEFT, Actes TALN 2011*.

GROUIN C., FOREST D., SYLVA L. D., PAROUBEK P. & ZWEIGENBAUM P. (2010). Présentation et résultats du défi fouille de texte DEFT2010 : Où et quand un article de presse a-t-il été écrit ? In *Atelier DEFT, Actes TALN 2010*.

JOACHIMS T. (1999). Making large-scale SVM learning practical. In B. SCHÖLKOPF, C. BURGES & A. SMOLA, Eds., *Advances in Kernel Methods - Support Vector Learning* : MIT Press, Cambridge, MA, USA.

MORIK K., BROCKHAUSEN P. & JOACHIMS T. (1999). Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML'99, p. 268–277.

NAJI N., SAVOY J. & DOLAMIC L. (2011). Recherche d'information dans un corpus bruité (OCR). In *CORIA 2011*.

NUNBERG G. (2009). Google's Book Search : A Disaster for Scholars.

ROSSET S., GALIBERT O., BERNARD G., BILINSKI E. & ADDA G. (2008). The LIMSI participation to the QAst track. In *Working Notes of CLEF 2008 Workshop*.

SCHMID H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, p. 44–49.

VAPNIK V. N. (1998). *Statistical Learning Theory*. John Wiley and Sons.

Comparaison de méthodes sémantique et asémantique pour la catégorisation automatique de documents

Romaric BOLEY
Université de Montréal – École de Bibliothéconomie et des Sciences de l'Information
C.P. 6128, Succursale Centre-Ville, Montréal, Québec, Canada, H3C 3J7
romaric.boley@umontreal.ca

Résumé

Cet article présente les résultats obtenus pour les deux tâches proposées par le DEFT 2011. La première, variation diachronique, consistait à déterminer la date de parution d'un article de presse française. La deuxième tâche consistait à apparier un résumé avec l'article scientifique dont il a été extrait. L'objectif à travers ces études était de comparer, pour chacune des tâches, les démarches sémantique et asémantiques (extraction des longueurs des phrases et sélection de mots non porteurs de sens) pour la catégorisation automatique de documents.

Abstract

This paper presents the results for both tasks proposed by the DEFT2011. The first one, the diachronic variation, consisted of determining the publication date of French press articles. The second task consisted of matching a summary to the scientific paper from which it was extracted. The objective through these studies was to compare, for each of the tasks, the unsemantic (i.e. sentences length or stop words extraction) and semantic processes.

Mots-clés : catégorisation ; approche sémantique ; approche asémantique, fouille de texte
Keywords: classification ; semantic ; unsemantic ; text mining

1 Introduction

L'information de nos jours occupe une place centrale dans nos sociétés. L'ère industrielle a laissé place à l'ère de l'information. Du fait du volume croissant d'informations numériques non structurées, le domaine de la fouille de textes est en pleine expansion. Avec le développement d'Internet, on retrouve plusieurs applications issues de la recherche en fouille de textes, par exemple pour la gestion des courriels indésirables ou encore pour la détection de la cybercriminalité (Kontostathis A. 2009). Cette année, le DEFT 2011 propose deux thématiques. D'une part l'étude de la variation diachronique en corpus de presse, et d'autre part, l'appariement d'articles scientifiques avec leur résumé.

L'heure est à la numérisation du patrimoine culturel comme en témoigne l'appel du Ministère de la Culture de la Communication (France) pour rendre ce patrimoine largement accessible. Face à un projet de numérisation de grande ampleur, on peut craindre que certaines informations nécessaires au repérage du document risquent d'être absentes de l'original. À ce titre, les tâches proposées cette année par le comité d'organisation du DEFT 2011 peuvent rapidement trouver une application. En effet, l'année de publication d'un document est une donnée primordiale pour le repérage. La datation d'un document peut se faire manuellement par un expert, mais toutes les structures ne disposent pas des ressources et du temps nécessaires à une telle expertise. Les études menées par les différentes équipes du DEFT 2011 permettront certainement de développer des modèles et des classifieurs afin d'automatiser cette tâche. Toutefois ce n'est pas une tâche aisée car la numérisation à l'aide de la reconnaissance optique de caractères n'est pas exempte d'erreurs, rendant la reconnaissance de motifs difficile.

Le résumé de texte est une autre application de la fouille de texte. Cette année le défi ne consiste pas à résumer automatiquement un texte, mais à l'associer à l'article pour lequel il a été écrit. La réalisation de cette tâche offre plusieurs pistes d'application telles que la recherche d'articles à partir de résumés, ou l'analyse de résumé pour en isoler les principaux éléments afin d'améliorer la génération automatique de résumés. Mais cette tâche peut aussi permettre d'accélérer le processus de recherche d'information, grâce à une indexation automatique de l'article à partir de son résumé et non de l'article ou d'un livre. Sur des volumes importants, le gain de temps pourrait ne pas être négligeable.

2 Objectif général

À travers les deux tâches proposées pour le DEFT 2011, nous souhaitions observer si l'utilisation d'une stratégie asémantique donne des résultats proches ou meilleurs que ceux issus d'une stratégie sémantique. L'approche sémantique consistera à ne garder que les mots les plus significatifs pour chaque texte alors que l'approche asémantique se basera sur l'utilisation des mots non porteurs de sens, aussi appelés mots vides, tels que les particules, les mots de liaisons, les verbes auxiliaires etc. Une autre approche asémantique consistera à ne garder que les longueurs des phrases.

Un des avantages de l'approche asémantique est que l'on peut facilement faire abstraction de la langue (en utilisant la longueur des phrases par exemple), et qu'elle est moins coûteuse en terme de calcul (en utilisant uniquement les mots non porteurs de sens).

3 Tâche 1 : Variation diachronique

3.1 Objectif

Le but de cette tâche est de pouvoir attribuer la bonne année de publication à un extrait d'article de presse. Cette tâche est complexe car s'il est possible d'identifier des mots propres à une époque, notamment grâce aux réformes de l'orthographe (Albert *et al.* 2010), il est plus difficile d'associer un mot à une année précise ou du moins d'identifier un mot caractéristique d'une année précise.

3.2 Méthodologie

La démarche utilisée pour le traitement des documents repose sur un modèle vectoriel (Salton 1988, Memmi 2000, Forest 2009). L'analyse manuelle du corpus a fait ressortir beaucoup de problèmes liés au processus de numérisation avec reconnaissance optique de caractères. Certaines terminaisons de mots se voyaient tronquées, ou encore de nombreuses lettres « a » remplacées par des « o ». Malgré ces erreurs, nous avons fait le choix de ne pas intervenir sur le corpus original, de peur d'ajouter des erreurs en appliquant par exemple une orthographe actuelle à un mot orthographié différemment à son époque.

Les corpus de documents ayant été fournis par l'équipe organisatrice du DEFT 2011, nous avons pu commencer immédiatement par l'extraction et le filtrage du lexique. Trois types d'extraction ont été mis en place suivant la stratégie utilisée, sémantique ou asémantique. Pour la stratégie consistant à ne garder que les mots significatifs, un premier filtrage est appliqué, à partir d'une liste, pour enlever les mots non porteurs de sens. Dans le cas des stratégies asémantiques, un filtrage est effectué pour ne garder que les mots non porteurs de sens. Aucun filtrage n'est effectué pour la stratégie portant sur l'extraction des longueurs des phrases. Toujours dans un souci de garder les spécificité des mots par rapport à leur époque, aucun algorithme de lemmatisation n'a été appliqué.

Une fois le lexique extrait, il reste à représenter chaque document sous forme vectorielle. Ainsi chaque document sera représenté par la présence d'un terme pondérée par le *tf*idf*.

$$tf * idf = \mathrm{tf} \cdot \log_2\left(\frac{n}{df}\right)$$

où n est le nombre de documents

tf est la fréquence du terme (*term frequency*) dans le document

df est la fréquence de document ou apparaît le terme (*document frequency*)

Afin de pouvoir appliquer l'attribution automatique de l'année, nous avons recouru à un classifieur utilisant l'algorithme des bayesiens naïfs (Manning et Schütze, 1999), classifieur entraîné sur le corpus d'apprentissage à l'aide d'une validation de type *leave one out*.

3.3 Corpus

Le corpus est composé de textes numérisés avec reconnaissance optique de caractères (OCR). Ceci implique de nombreuses erreurs. On peut ainsi lire dans l'un des documents : « ne songe à tirer aucun parti des folles qui se icltCBt à>3 tète il le* trompe sans awtre ». Cette exemple montre la difficulté d'identifier certains mots originaux, même placés dans leur contexte. C'est pourquoi aucun traitement ne sera fait pour corriger le corpus des erreurs liées à l'OCRisation pour cette tâche.

Le corpus d'apprentissage est constitué de 3596 extraits de six journaux français (300 ou 500 mots) parus entre 1801 et 1944. Le corpus a été traité différemment selon les trois orientations choisies.

Les mots non porteurs de sens ont été supprimés pour la stratégie consistant à ne garder que les mots qui ont du sens. Seuls les mots non porteurs de sens ont été gardés pour la stratégie asémantique basée sur les mots non signifiants. En ce qui concerne la stratégie asémantique portant sur la longueur des phrases, chaque phrase a été remplacée par sa longueur[1] en nombre de tokens. Ainsi, seule la longueur des phrases à été retenue.

[1] La phrase « trahir cette mission, ni de la déserter » est remplacée par « Lg7 »

3.4 Résultats : phase d'apprentissage

3.4.1 Phase d'apprentissage

L'ensemble des tests a été effectué avec les paramètres suivants :

- Pondération des attributs : Chi2 maximum
- Méthode de validation : Retrait d'un cas
- Algorithme : Bayésien naifs
- Méthode de représentation des traits : Pourcentage de mots clés

Stratégie	Nombre d'attributs	Score
Longueur des phrases	154	7,23%
Mots non porteurs de sens	505	9,75%
Mots porteurs de sens	103897	39,44%

Tableau 1 : Apprentissage pour le corpus contenant des extraits de 500 mots

Stratégie	Nombre d'attributs	Score
Longueur des phrases	137	7,09%
Mots non porteurs de sens	495	8,40%
Mots porteurs de sens	75116	32,78%

Tableau 2 : Apprentissage pour le corpus contenant des extraits de 300 mots

Le score présenté dans les deux tableaux est calculé suivant la formule suivante :

$$S(p)=\frac{1}{N}\sum_{i=1}^{N} e^{-\frac{\pi}{10^2}\left(d_p(a_i)-d_r(a_i)\right)^2}$$

où $d_p(a_i)$ est la date prédit pour le fragment de texte a_i

$d_r(a_i)$ est la date référence pour le fragment de texte a_i

N est le nombre total de fragments de texte

Les résultats des deux tableaux montrent que la stratégie sémantique, basée sur les mots porteurs de sens, est nettement meilleure. Ceci s'explique notamment par le nombre conséquent d'attributs, ce qui permet un apprentissage plus important et donc d'être plus précis dans l'attribution des catégories. Toutefois, il serait intéressant, dans de futurs travaux, de comparer l'approche sémantique avec une combinaison d'approches asémantiques.

3.4.2 Phase de test

Les tests ont été réalisés grâces aux classifieurs modélisés lors de la phase d'apprentissage.

Test	Score
1-Longueur des phrases	6,2%
2-Mots non porteurs de sens	7,3%
3-Mots porteurs de sens	6,1%

Tableau 3 : Résultats pour la variation diachronique sur les extraits de 500 mots

Bien que les résultats soient faibles, on peut constater que la stratégie qui était la plus prometteuse lors de la phase d'apprentissage s'est avérée être la plus faible lors de la phase de test. En revanche la stratégie asémantique portant sur les mots vides (non porteurs de sens) a donné les meilleurs résultats. Ceci pourrait s'expliquer par le fait que ces mots ont moins été touchés par les problèmes de l'OCRisation.

4 Tâche 2 : Appariement résumé / article

4.1 Objectif

Pour résoudre cette tâche il faut associer un résumé à l'article scientifique dont il a été extrait. Pour réaliser l'appariement deux piste étaient proposées. La piste 1 consistait à associer le résumé à un article complet alors que la piste 2 consistait à l'associer à un article privé de son introduction et de sa conclusion.

4.2 Méthodologie

Pour trouver l'article associé à un résumé, nous nous sommes basés sur les travaux de Salton (présentés dans Ibekwe-SanJuan 2007) sur la recherche automatique d'information (Salton, 1988), notamment la formule de similarité entre une requête (Q) et un document (D).

$$similarité\,(Q,D) = \frac{\sum_{k=1}^{t} w_{qk} \cdot w_{dk}}{\sqrt{\sum_{k=1}^{t} (w_{qk})^2} \cdot \sqrt{\sum_{k=1}^{t} (w_{dk})^2}}$$

où w_{qk}, w_{dk} sont respectivement les poids des mots dans la requête et dans le document

La première étape consiste à représenter chaque résumé et chaque document par un vecteur de mots. Par la suite, une pondération *tf*idf* (cf. 3.2) est appliquée sur chaque composante du vecteur, étape suivie d'une normalisation pour avoir des poids compris entre 0 et 1. Ceci afin de minimiser l'importance des termes trop

fréquents dans le corpus qui deviennent alors peu discriminants. Plus le résultat du calcul de similarité, suivant la formule ci-dessus, est proche de 1, plus les vecteurs représentant le résumé et l'article sont proches.

Ainsi, pour chaque résumé, on calcule la similarité entre celui-ci et chacun des articles, ce qui correspond au calcul du cosinus entre les deux vecteurs. L'article qui lui sera associé sera celui qui aura permis d'obtenir le score le plus élevé lors du calcul de similarité.

4.3 Corpus

L'ensemble du corpus d'apprentissage est constitué de 198 articles et résumés. La tâche se subdivise en deux pistes. L'une, contient les articles dans leur intégralité. Une deuxième piste contient les articles desquels ont été supprimées introduction et conclusion.

Un filtrage du lexique a ensuite été effectué pour les différentes stratégies : ne garder que les mots vides , supprimer les mots vides, ou extraire la longueur des phrases.

4.4 Résultats

4.4.1 Phase d'apprentissage

Les paramètres suivants s'appliquent pour chaque test :

- Variation du nombre de traits discriminants : entre 100 et 1000
- Sélection des termes discriminants : fréquence pondérée par IDF

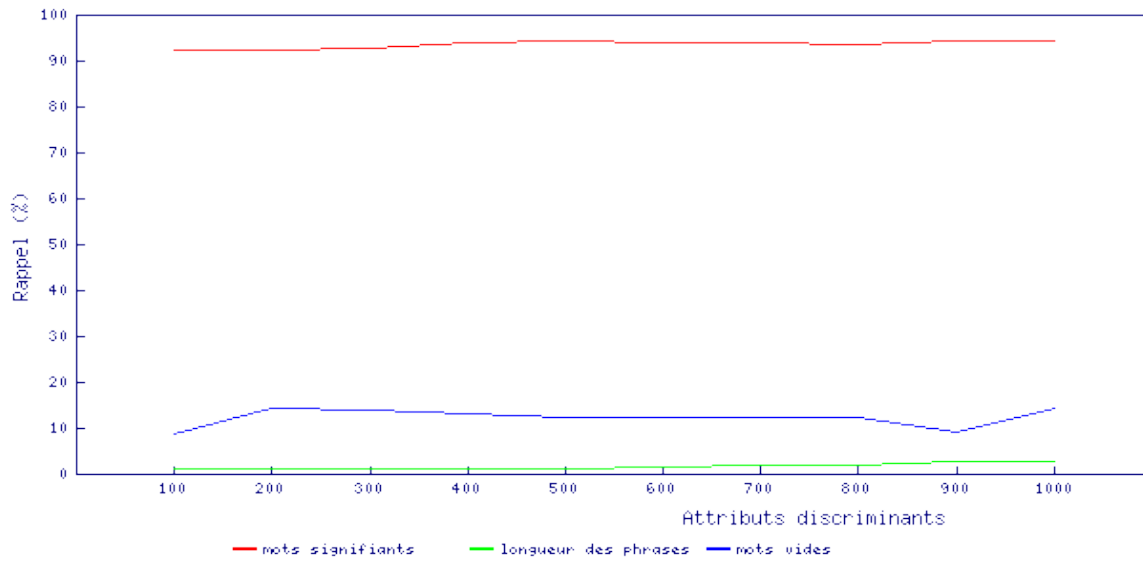


Figure 1: Performances de l'appariement, résumé/article sans introduction ni conclusion, suivant les stratégies adoptées

Les résultats montrent clairement que la stratégie utilisant les mots porteurs de sens est de loin la meilleure. Toutefois, cette dernière n'est pas parfaite, son score maximal est de 94% pour les articles privés de leur

introduction et conclusion et de 96.7% pour les articles complets. Toutefois il est intéressant de noter qu'elle varie seulement entre 92% et 94% pour les articles privés de l'introduction et de la conclusion et entre 95% et 96,7% pour les articles complets.

À l'issu de la phase d'apprentissage, on peut conclure que les approches asémantiques ne sont pas efficaces quand on les compare à l'approche sémantique. Donc, seule la stratégie basée sur les mots porteurs de sens sera retenue pour la phase de test.

4.4.2 Phase de test

Lors de la phase de test, seule l'approche sémantique a été retenue. Pour chaque test, les attributs sélectionnés sont pondérés par le *tf*idf*.

Test	Nombre d'attributs	Score global
1	600	98%
2	800	98,5%
3	1000	98%

Tableau 4 : Résultats pour l'appariement résumé / article complet

Test	Nombre d'attributs	Score global
1	800	95,4%
2	1000	95,5%

Tableau 5 : Résultats pour l'appariement résumé / article sans introduction ni conclusion

Le score global est la moyenne du nombre de résumés correctement appariés sur le nombre total de résumés. Ainsi le score global est obtenu suivant la formule :

$$S(p) = \frac{1}{N} \sum_{i=1}^{n} s\left(a_p(r_i), a_r(r_i)\right)$$

tel que $$s\left(a_p(r_i), a_r(r_i)\right) = \begin{cases} 1 & \text{si } a_p(r_i) = a_r(r_i) \\ 0 & \text{sinon} \end{cases}$$

où $a_p(r_i)$ est l'article prédit pour le résumé r_i

$a_r(r_i)$ est l'article de référence pour le résumé r_i

Les résultats présentés par les deux derniers tableaux montrent l'importance de l'introduction et de la conclusion qui permettent d'améliorer globalement l'appariement entre un résumé et son article. De plus nous pouvons noter qu'augmenter le nombre d'attributs à partir de 600 n'a pas vraiment d'influence sur les résultats.

5 Discussion

Les résultats obtenus pour la tâche d'appariement résumé / article sont relativement bons : 98,5% des résumés ont été correctement associés à l'article complet dont ils ont été extraits, et 95,4% pour l'appariement avec des articles privés d'introduction et de conclusion. L'avantage de la méthode utilisée, le calcul de similarité, est qu'elle n'est pas soumise au risque de surapprentissage. Par contre cette méthode est sensible à la longueur du résumé. Il est difficile de faire l'association d'un résumé très court : sur une ou deux phrase, il y a très peu de termes discriminants pouvant servir à la représentation vectorielle du résumé. Ainsi l'appariement devient incertain. Afin d'atteindre un appariement parfait, une solution pourrait être de mettre en place une lemmatisation des termes retenus afin de regrouper les mots ayant la même racine ou encore de ramener tous les verbes conjugués à leur infinitif. Une autre piste, en s'inspirant de des travaux de Lin, C. (Lin, 1995) serait de travailler avec des concepts plutôt que des mots. En effet, lorsque l'on écrit un texte nous avons tendance à vouloir éviter la répétition pour rendre la lecture plus fluide et agréable. Ainsi des mots comme « table », « armoire », « buffet » seraient regroupés sous le même concept de « meuble ». Toutefois, il y a toujours le risque de perdre la spécificité de certains termes discriminants. De plus, il pourrait être intéressant de donner un poids plus important aux termes extraits des introduction et conclusion afin de vérifier si l'appariement est meilleur.

Il est important de noter que les résultats obtenus pendant la période d'apprentissage n'ont pas été concluants pour les stratégies asémantiques. Il paraît évident que ce qui est associé au style d'écriture (longueur des phrases, utilisation des mots vides), ne permet pas d'identifier précisément le résumé d'un article. Ces résultats sont encourageants pour la génération automatique de résumés. En effet, cela veut dire que le style employé pour rédiger le résumé ne rentre pas en compte dans l'appariement. Quand bien même le résumé serait rédigé par une tierce personne ou automatiquement, nous serions donc encore capable de déterminer quel résumé appartient à quel article. Le générateur automatique de résumé doit par conséquent porter toute son attention sur les mots ou concepts caractérisant l'article.

Les résultats obtenus pour l'étude de la variation diachronique sont particulièrement décevants. Ceci s'explique notamment par un surapprentissage du classifieur. Les résultats obtenus pendant la phase d'apprentissage était de 39,44%, pour le corpus dont les extraits étaient de 500 mots, alors que lors de la phase de test, le meilleur résultat obtenu fut de 7,3%. Du fait de problèmes de ressources matérielles, nous avons été contraint de choisir une validation de type *leave one out*, nécessitant moins de ressources que la validation croisée. Choisir ce type de validation a été une erreur étant donné que cette méthode est caractérisée par un biais faible accompagné d'une grande variance (Cios, 2007). En d'autres termes, cette méthode garantit de meilleurs résultats que celle de la validation croisée, mais sa forte variance risque, lorsque les résultats seront faibles, d'être très éloignée de la meilleure solution.

Ces résultats sont aussi dus aux erreurs générées par le processus d'OCRisation. De nombreuses erreurs sur les mots ont mis à mal la stratégie asémantique, et la méthode asémantique basée sur la longueur des phrases n'a pu être optimale car de nombreux caractères de ponctuation apparaissaient aléatoirement au beau milieu d'une phrase. De plus la longueur des textes (300 ou 500 mots) était probablement insuffisante pour de telles stratégies. Toutefois il est intéressant de noter que la stratégie basée sur les mots non porteurs de sens à donné des résultats légèrement meilleurs. Cette stratégie était bien moins coûteuse en temps que celle basée sur le sens des mots. Si une stratégie asémantique devrait être testée de nouveau, il faudrait s'assurer d'avoir des textes plus longs et de corriger les erreurs du corpus. Par ailleurs, il faudrait définir de manière plus exhaustive ce qui caractérise le style d'écriture et combiner plusieurs facteurs.

Remerciements

Je tiens à remercier particulièrement le comité d'organisation du DEFT 2011. Je tiens aussi à remercier tous les participants des défis antérieurs. Grâce à la qualité des travaux effectués les années précédentes, j'ai pu me familiariser, en français, avec la fouille de textes sur des thématiques bien spécifiques. Mes remerciements vont aussi à Dominic Forest qui m'a fait découvrir le domaine de la fouille de texte et qui m'a encouragé à participer au DEFT2011.

Références

ALBERT P., BADIN F., DELORME M., DEVOS N., PAPAZOGLOU S., SIMARD J. (2010). Décennie d'un article de journal par analyse statistique et lexicale. Acte de *TALN2010*.

CIOS K. J. (2007). D*ata Mining : a Knowledge Discover approach*. New York: Springer.

BATHIA V. (1993). *Analysing Genre. Language Use in Professionnal Setting*. Boston: Addison Wesley.

FOREST D., HOEYDONCK VAN A., LÉTOURNEAU D., BÉLANGER M. (2009). Impacts sur la variation du nombre de trait discriminants sur la catégorisation des documents. Acte de *TALN2009*.

IBEKWE-SANJUAN F. (2007). *Fouilles de textes : méthodes, outils et applications*. Paris: Lavoisier.

KONTOSTATHIS A., EDWARDS L., LEATHERMAN A. (2009). Text mining and Cybercrime In *Text Mining : Application and Theory*. Chichester, U.K. : Wiley.

LIN C. (1995). Knowledge-Based Automatic Topic Identification. Actes de 33RD *Annual Meeting of the Association for Computational Linguistics*, 308-3010.

MANNING, C. D., SCHÜTZE, H.(1999). Foundations of statistical natural language processing. Cambridge (Mass.) : MIT Press.

MEMMI D. (2000). *Le modèle vectoriel pour le traitement de documents*. Grenoble: Cahiers Leibniz, no 2000-14.

ROWLEY J. (1982). *Abstracting and Indexing*. London: Clive Bingley.

SALTON G., BUCKLEY C. (1988). Term-weighting approches. *Automatique texte retrieval in information processing and management* 24(5), 456-465.

Tâche 2. Appariements

Deft 2011: Appariement de résumés et d'articles scientifiques fondé sur des distributions de chaînes de caractères

Gaël Lejeune, Romain Brixtel et Emmanuel Giguet

Université de Caen, GREYC UMR 6072, Boulevard du Maréchal Juin 14032 Caen Cedex, France
prenom.nom@unicaen.fr

Résumé

Nous présentons ici une expérimentation dans le cadre de la seconde tâche du défi fouille de textes (DEFT) 2011: appariement de résumés et d'articles scientifiques en français. Nous avons fondé nos travaux sur une approche à base de distribution de chaînes de caractères de manière à construire un système simple et correspondant à une conception endogène et multilingue des systèmes. Notre méthode a obtenu de très bons résultats pour la piste 1 "articles complets" (100%) mais a été moins efficace sur la piste 2 "articles sans introduction ni conclusion" (96%).

Abstract

We present here our work on the second task of 2011's Deft: pairing scientific articles and their abstract. Our approach is based on distribution of character strings. Our aim was not only to be efficient on that particular task on French but to build a system that can easily be used for other languages. Our method achieved very good results on track 1 "full articles" (100%) but had more problems with track 2 where introduction and conclusion were removed (96%).

Mots-clés : Chaînes de caractères répétées maximales, méthode endogène, approche multilingue, linguistique différentielle, algorithmique du texte

Keywords: Maximal repeated character strings, endogenous method, multilingual approach, differential linguistics, stringology

1 Introduction

Nous présentons ici une expérimentation dans le cadre de la seconde tâche du Défi Fouille de Textes 2011: appariement de résumés et d'articles scientifiques en français. Nous avons fondé nos travaux sur une approche à base de distributions de chaînes de caractères de manière à construire un système sans ressources externes d'une part et potentiellement multilingue d'autre part. Nous allons expliquer les raisons de ces choix et leurs implications.

1.1 Cadre de travail

L'axe de recherche "Multilinguisme, traduction, algorithmique du texte et méthodes différentielles" du laboratoire GREYC promeut depuis fort longtemps un traitement automatique des langues avec des ressources légères, dans la lignée des travaux de Jacques Vergne. Cette approche permet des traitements résolument multilingues (Lucas, 1993 ; Vergne, 2001).

Il défend la position selon laquelle l'interprétabilité du résultat ne passe pas nécessairement par l'interprétabilité des opérandes du calcul ayant produit ce résultat. Ainsi les traitements à base de ressources, si légères soit-elles dans les travaux de Vergne, ont-ils été progressivement délaissés pour laisser place à des traitements dits endogènes suite aux travaux de (Déjean, 1998). Les ressources lexicales ne sont alors pas une entrée nécessaire mais une production du calcul (Giguet & Lucas, 2004 ; Giguet & Luquet, 2006).

Le concept de mot est encore prégnant dans certains travaux du groupe qui nécessitent une segmentation classique en mots. Cette approche tend aujourd'hui à disparaitre de nos approches au profit d'un traitement basé sur les caractères (Lardilleux, 2010 ; Brixtel *et al.*, 2010 ; Lécluze, 2011). Ces opérandes souvent non interprétables par le lecteur humain, puisque pouvant débuter à n'importe quel caractère du texte pour se terminer à n'importe quel autre, ont l'intérêt de laisser envisager des traitements automatiques multilingues, incluant des langues où le mot n'est pas graphiquement délimité (Lejeune *et al*., 2010b). En effet, pour l'ordinateur, un mot est une chaîne de caractères comme une autre et effectuer une opération sur un mot n'a pas plus de sens que de l'effectuer sur une chaîne de caractère quelconque : ses capacités de calcul ne s'en trouvent pas dégradées.

Si le traitement au grain caractère a pu se développer et trouver sa pertinence, c'est certainement par le fait que parallèlement à ces réflexions sur la définition d'un grain d'analyse adéquat pour tel ou tel traitement se développait une approche guidée par le modèle, sous l'influence de Nadine Lucas (Lucas, 2004 ; Lucas, 2009a). Le traitement des langues au GREYC rencontre alors la fouille de données pour donner naissance à des travaux inter-équipes : approches inductives de la fouille de données textuelles (Turmel *et al*., 2003 ; Lucas & Crémilleux, 2004). L'approche guidée par le modèle défendue par Nadine Lucas s'oppose à la vision selon laquelle le texte serait non-structuré, une simple suite de phrases ou pire encore un sac de mots. L'introduction du modèle permet de mettre en scène le critère de position et de grain d'analyse, et ainsi d'ancrer la recherche de cooccurrences de chaînes de caractères (Lejeune *et al*., 2010a), pour produire un résultat qui fait sens pour l'utilisateur. Ainsi pourrait-on résumer le traitement des langues "à la mode de Caen" et illustrer ce positionnement dans la participation à ce Défi Fouille de Textes.

1.2 Stratégie de résolution

C'est par la mise en œuvre la plus simple et la plus immédiate de ces deux caractéristiques de notre méthode, à savoir traitement au caractère et approche guidée par le modèle, que nous avons choisi d'aborder le sujet et tenté de montrer la pertinence de la méthode. D'un point de vue général, nous avons souhaité une solution qui soit simple d'un point de vue calculatoire : nous n'avons donc pas cherché à maximiser une fonction de qualité globale des appariements sur la collection. Nous avons choisi au contraire un appariement séquentiel, et sans remise en cause des appariements effectués. L'hypothèse sous-jacente est qu'un résumé et un article sont en quelque sorte indissociables, de sorte qu'il n'est pas nécessaire d'envisager une quelconque ambiguïté d'appariement entre un article et plusieurs résumés ou entre plusieurs articles et un résumé.

Dans cette même recherche de solution simple d'un point de vue calculatoire, nous avons considéré qu'il était plus efficace de rechercher pour chaque document son résumé, plutôt que de rechercher pour chaque résumé son document. L'espace de recherche de la collection de résumés est en effet plus petit que l'espace de recherche de la collection d'articles. Par ailleurs on suppose qu'un article contient toutes les informations importantes disponibles dans le résumé, alors que l'inverse n'est pas vrai. Chercher à quel article correspond tel résumé serait alors potentiellement générateur d'"ambiguïtés" artificiellement engendrées par la démarche.

Notre approche prend donc à ce titre le contrepied des applications de recherche d'information classique, ou le résumé serait envisagé comme la requête posée à un moteur travaillant sur la collection d'articles indexés. Cette approche aurait été pertinente en terme de réutilisabilité de technologies disponibles, mais peut être plus discutable en terme d'adéquation au problème.

Nous n'avons pas non plus cherché a pondérer les fréquences des éléments recherchés en fonction de la collection (approche de type *tf/idf*). Nous avons opté pour une approche moins coûteuse à calculer, qui consiste a considérer que la simple cooccurrence de séquences communes au résumé et à l'article constitue un indice de corrélation suffisamment fiable pour un appariement de qualité, d'autant plus fiable qu'il est cohérent avec les positions définies dans le modèle d'article attendu.

Du point de vue linguistique, nous avons adopté une tripartition des articles : introduction, développement et conclusion. La mise en œuvre informatique consiste à calculer cette tripartition. La segmentation est déduite de la structure physique dont la trace se manifeste par la présence d'éléments XML "titre", qu'il s'agisse de titre ou de sous-titre. La segmentation ne repose donc pas sur la recherche de mots-clés comme "introduction" , "conclusion" qui induirait une dépendance à la langue ou aux variations de libellé, comme le titre "discussion" qui peut faire fonction de conclusion.

Le premier segment, du début du texte à la première balise titre est associé à l'introduction, le dernier segment, de la dernière balise titre à la fin du texte, est associé à la conclusion, et par différence, le reste est associé au développement. Cette mise en œuvre simple part de l'hypothèse que l'introduction et la conclusion sont souvent non découpées en sous-sections, contrairement au développement de l'article.

D'un point de vue pragmatique, nous supposons que le résumé contient des reprises a l'identique de l'introduction, et que le contexte, la thématique et les perspectives sont des points communs que partage le résumé avec le couple introduction-conclusion. De fait, l'implémentation traduit ces hypothèses : (1) la recherche de la plus longue séquence de caractères présente dans l'article, attendue dans l'introduction, et unique dans la collection de résumés, (2) la plus forte corrélation en terme de séquences de caractères partagées entre l'article et le résumé, attendu principalement dans l'introduction et la conclusion.

Dans la partie qui suit, nous détaillerons nos différentes expérimentations, avec différentes relaxations des contraintes.

2 Cadre théorique et définitions

Le fait que deux documents puissent constituer un couple résumé-article provient a priori de certaines connections que l'on peut trouver entre eux. Dans notre travail nous avons nommé ces connections, ces points communs, des affinités. Ces affinités sont des chaînes de caractères, mots ou non-mots, communes aux deux documents. Dans la terminologie que nous allons utiliser par la suite, chaque article est un célibataire qui possède un certain nombre de prétendants: les résumés. Pour former des couples nous faisons une hypothèse contrastive, parmi une collection de résumés nous recherchons celui qui possède les meilleures affinités avec un article. La proximité entre un article et un résumé ne se juge donc pas localement mais par rapport à la collection.

2.1 Cadre théorique

On cherchera donc à partir d'un corpus de célibataires d'une part et d'un corpus de prétendants de l'autre à obtenir le plus de couples corrects résumé-article. Le bon prétendant pour un célibataire donné sera le résumé qui partagera le plus grand nombre d'affinités avec un article.

Ces affinités seront des chaînes de caractères, mots ou non mots. Nous aurions pu utiliser simplement des mots mais dans la lignée des principes décrits plus haut nous avons souhaité:

- Ne pas nous baser sur des pré-traitements (lemmatisation par exemple) pour pouvoir effectuer certaines comparaisons (retrouver « traduction » dans « traductions » par exemple)

- Favoriser, bien que le corpus soit finalement monolingue, une méthode qui soit facilement réutilisable pour des corpus multilingues.

Plus généralement, nous n'avons stocké aucune information à l'issue de la phase d'apprentissage ni utilisé aucune ressource externe. Cette phase initiale nous a servi simplement à éprouver le système. Dans la même idée de généricité et pour faciliter le passage à l'échelle nous n'avons pas souhaité utiliser les informations concernant la revue. Le système que nous présentons va donc chercher le résumé correspondant à un article donné parmi toute la collection de prétendants sans pré-filtrage. Toutefois, et nous le verrons plus loin, une revue a posé plus de problèmes que les autres.

Nous cherchons ici à mettre en avant la généricité et la parcimonie, dans la lignée des travaux décrits plus haut. Si le but d'un concours est bien entendu de faire le meilleur score, nous nous sommes attachés dans notre démarche à ne pas créer un système trop complexe ou trop paramétrable. Au contraire c'est le même système que nous avons fait fonctionner pour le "run" de référence de chaque piste.

2.2 Définition des affinités

Des segments présents dans l'article sont repris par l'auteur dans l'écriture de l'*abstract*. Selon les stratégies mises en place par l'auteur pour construire son résumé, la recopie pourra être plus ou moins prononcée. Cette recopie pourra être un mot, un groupe de mots voire une phrase ou une proposition. L'utilisation des chaînes de caractères répétées maximales (rstr-max) permet de repérer des unités qui se rapprochent dans une certaine mesure des unités multi-mots (Doucet, 2006).

Pour rechercher quel résumé correspond à quel article nous recherchons donc des points d'ancrage. Ces points d'ancrage sont des rstr-max entre un résumé R et un article A. Notre hypothèse est que nous pouvons les rapprocher s'ils ont des segments en commun longs et nombreux. Nous appelons ces segments des affinités, plus un couple R-A possède d'affinités, si possible de grande taille, plus il y a de chances qu'il constitue un appariement correct.

Les chaînes de caractères constituant nos affinités sont repérées à l'aide d'une implémentation python disponible en ligne[1]. Elles sont présentes plus d'une fois (répétées) et ne sont pas strictement contenues dans une chaîne répétée plus grande de même fréquence (maximales):

tototo a pour rstr-max toto (fréquence 2) et *to* (fréquence 3) mais pas *t* qui est de même fréquence que to et se trouve aux mêmes positions

Grâce à notre implémentation, en comparant un célibataire à tous ses prétendants nous obtenons une structure de données donnant pour chaque rstr-max, les documents du corpus dans lesquels elle apparaît. La fréquence de ces affinités à l'intérieur du document ne nous intéresse pas dans cette étude. D'autre part nous ne conservons que les affinités entre le célibataire et ses prétendants, les affinités entre prétendants ne sont pas prises en compte.

[1]http://code.google.com/p/py-rstr-max/

3 Filtrage des affinités

Les deux mesures utilisées seront la taille en caractères de la plus grande affinité (affinité-max) et le nombre total d'affinités hapax (card-affinités) pour chaque couple potentiel. Le critère affinité-max n'est pas suffisamment fiable pris isolément mais est complémentaire avec le second. Le nombre total d'affinités peut quand à lui souffrir de la sur-représentation d'affinités a priori peu significatives. En effet le nombre de rstr-max pour un document donné est potentiellement très élevé. Sur des documents en langue naturelle, ce nombre est quadratique en la taille des documents.

On filtre en ne gardant que les affinités qui sont "hapax" dans la collection de prétendants. Nous supposons que ce qui est rare peut avoir une grande valeur. En l'occurrence on ne tient compte d'une affinité entre un célibataire et un de ses prétendants que si cette affinité n'est pas partagée par d'autres prétendants, donc n'est pas banale. Ce critère d'exclusivité évite la surgénération de motifs qui pourrait intervenir avec des rstr-max.

Donc si un article a une affinité commune avec plusieurs résumés, cette affinité n'est pas considérée comme significative. De cette façon nous cherchons à ne pas tenir compte de celles qui pourraient être trop peu discriminantes. Notre hypothèse est qu'un couple réussi doit partager des affinités "originales". En pratique en plus d'éviter de prendre en compte des termes trop génériques et trop largement distribués, nous filtrons ainsi de nombreuses affinités "vides" (Figure 1).

«resse» «ymbol» «ssib» «est p» «ns et» «la mise en» «s donné» «ntifi» «à m» «qu'elle» d'une co» «e ap» «les, » «s qua» «ur l'a» «amin» «lum» «ns f»

Figure 1: Exemples d'affinités vides

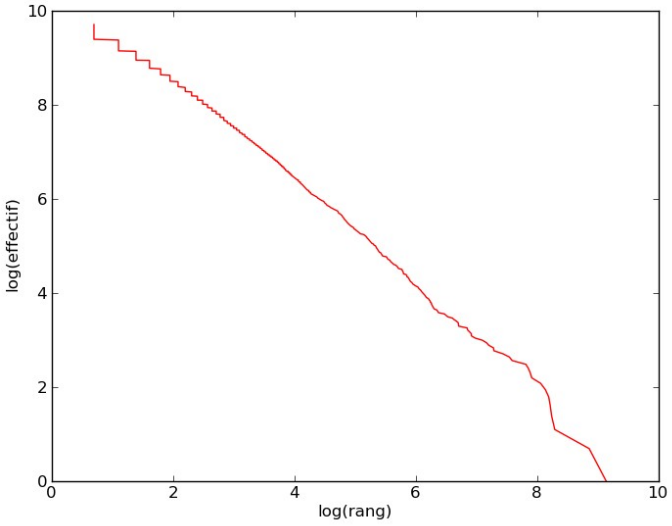


Figure 2: Loi de Zipf sur les rstr-max

On peut remarquer sur la figure 2 que la loi de Zipf s'applique très bien aux chaînes de caractères répétées maximales. Dès lors, on peut d'une certaine manière parvenir à caractériser des chaînes de caractères "vides". Nous utilisons ici le terme vide dans la même acception que celle qui est la sienne dans

l'opposition mot-plein/mot-vide ou terme-plein/terme-vide. Nous trouvons effectivement dans les rstr-max figurant en haut à gauche de la courbe (très courtes et très fréquentes) des affixes, des enchaînements de caractères très fréquents et des mots courts. Au contraire, les affinités rares et tout spécialement les hapax sont des chaînes auxquelles on pourrait plus facilement rattacher un sens, pour analyser des erreurs par exemple. Cette observation nous a semblé un pas intéressant vers la validation de notre hypothèse: les couples semblaient formés pour de "bonnes" raisons (Figure 3).

«a philosophie politique d» «s les organisations» «r la reconnaissance des»
«des organisations internationales» «s les années 1970» «établissements»

Figure 3: Exemples d'affinités pleines

La figure 4 illustre l'importance de l'utilisation du critère de fréquence. La fréquence 2 en abscisse signifie que l'affinité est présente dans l'article et dans un seul résumé, on a donc une affinité qui est hapax dans la collection de résumés. On peut voir sur cette courbe que dès que l'on relâche cette contrainte, les résultats s'en ressentent. Par exemple, tenir compte des affinités présentes dans 2 résumés (fréquence 3) fait passer les résultats sur le corpus d'entraînement de 0.97 à 0.78.

Les chaînes de caractères ci-dessus ne signifient rien en elles-mêmes. C'est au niveau de l'improbabilité relative de la présence d'une chaîne répétée que se situe le critère de décision (Church 2000). Nous pouvons voir enfin sur la figure 5 qu'un certain seuil dans la significativité de la taille des affinités peut être observé. Fixer leur taille minimale entre 7 et 12 caractères peut optimiser les résultats. En dessous de ce seuil le score reste supérieur à 0.9 donc le filtrage par les hapax est efficace. Par contre au delà, et spécialement à partir de 20 caractères, les résultats sont en chute libre : il n'y a plus assez d'affinités à observer.

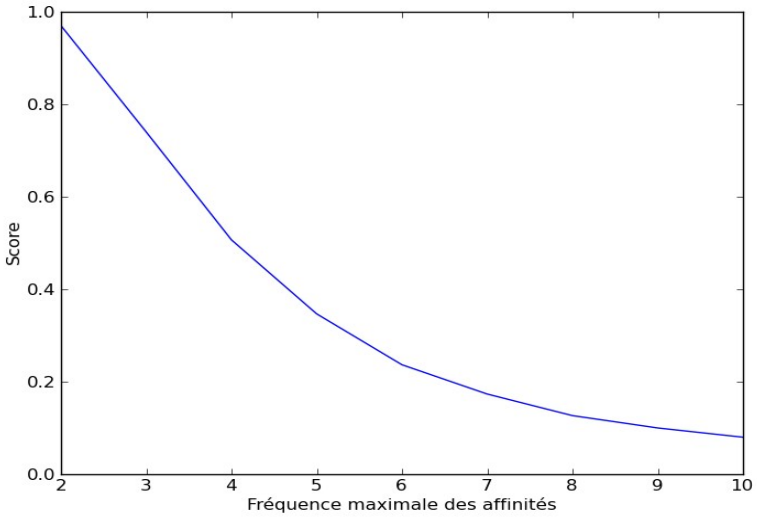


Figure 4: Corpus d'apprentissage, évolution du score selon la fréquence maximale des affinités dans le sous corpus de prétendants

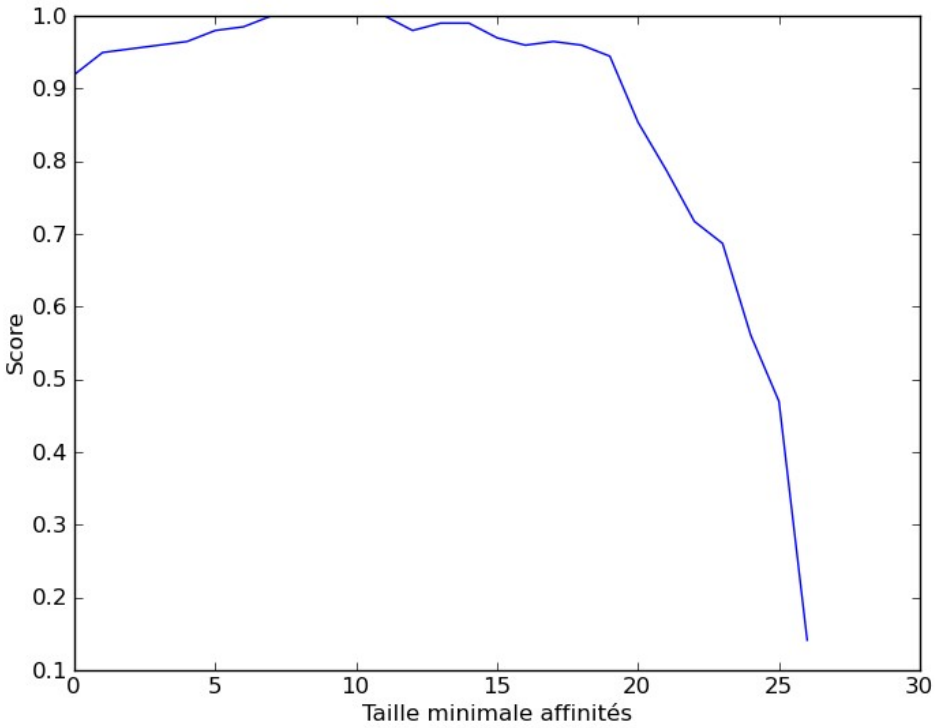


Figure 5: Corpus de test, influence d'un seuil de taille des affinités sur le score

4 Fonctionnement et résultats

Nous prenons en entrée la liste des articles et des résumés à apparier. Chaque célibataire (article à apparier) est comparé à tous ses prétendants (résumés à apparier). L'implémentation py-rstr-max calcule les chaînes de caractères répétées maximales (rstr-max). On ne garde que les affinités qui sont hapax dans le corpus de prétendants et de taille supérieure à 8. Ce seuil a été fixé pour le français de manière empirique et a été validé par le calcul mais pourrait sans doute être recalculé pour chaque collection quelle que soit la langue.

4.1 Description locale

On compare un article à la collection de résumés et on forme un couple chaque fois qu'il semble significativement relié par des affinités. Pour ce faire il faut qu'ils partagent l'affinité-max trouvée dans la collection de prétendants et un nombre significatif d'affinités "uniques". Si un prétendant se détache (figure 4), alors un couple est formé et le célibataire et le prétendant concernés ne seront plus confrontés aux autres.

Nous avons donc cherché comment modéliser la significativité de cette répartition. Comment juger de la significativité du nombre d'affinités d'un prétendant par rapport à un autre? Nous avons remarqué en comparant l'ensemble des affinités d'un couple correct Article 1 - Résumé 1 avec celles de tous les autres couples possibles Article 1 - Résumé "x" (Figure 6) que les affinités hapax étaient réparties en trois tiers globalement équivalents en termes de fréquence:

- Des "affinités vides" mais non filtrées par le critère d'exclusivité
- Des affinités peu significatives et non discriminantes, très proches d'un document à l'autre
- Des affinités pouvant décrire les centres d'intérêt du célibataire, les thématiques de l'article.

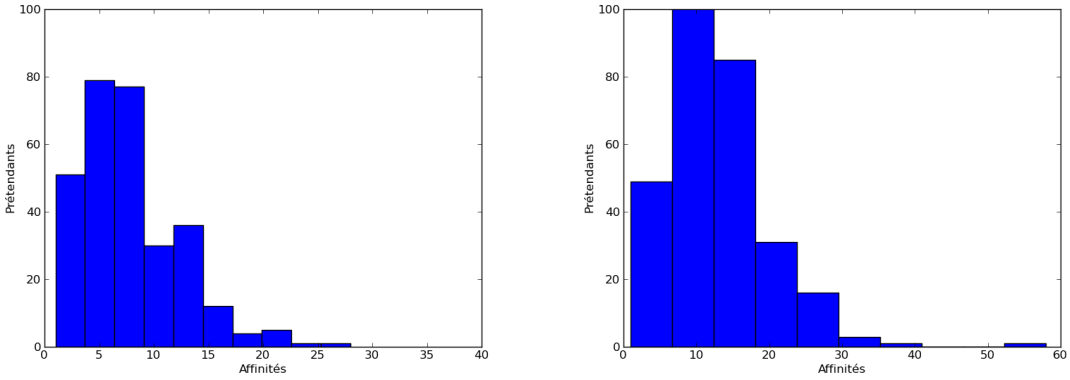


Figure 6: Répartition des prétendants par nombre d'affinités : à gauche pas de bon couple, à droite un prétendant se détache.

Nous avons observé que le nombre d'affinités existant entre un article et son résumé était le plus souvent au moins 1,5 fois supérieur à celui des autres prétendants. Quand ce critère n'est pas respecté on considère qu'il y a jalousie potentielle : aucun prétendant ne se détache il y a donc danger d'erreur dans la constitution du couple. Le célibataire sera alors laissé de côté et attendra une phase ultérieure pour être apparié.

On pourrait dès lors penser que l'ordre dans lequel nous traitons les célibataires introduit un biais. Le tableau suivant montre différents résultats selon l'ordre de tirage des célibataires. Nous avons fait des ordres de tirage aléatoires d'articles sur le corpus d'entraînement avec une différence peu significative. L'ordre n'introduit en fait de différences significatives que dans les dernières phases d'appariement, lorsqu'il ne reste que peu de documents à coupler (cf. Tableau 1).

Run	1	2	3	4	5	6	7	8	9	10
Score	0.97	0.967	0.97	0.97	0.973	0.97	0.967	0.967	0.97	0.963

Tableau 1: Corpus d'entrainement, tirage aléatoire de l'ordre d'apparition des célibataires dans la boucle

Chaque fois qu'un couple est formé, on considère que le prétendant ne doit plus être présenté aux autres célibataires. De cette façon pour les célibataires ayant du mal à trouver leur résumé, la tâche est facilitée: le nombre d'affinités hapax est plus grand, le critère devient plus discriminant tout en restant très efficace. Tant qu'il reste des célibataires, on les compare aux prétendants disponibles. La constitution de tous les couples nécessite en général 5 à 6 phases. Les dernières phases sont celles où l'on voit le plus d'erreurs, soit qu'elles soient dues à des appariements erronés dans les phases précédentes (cas rare), soit que le faible nombre d'affinités en jeu rende les derniers appariements moins convaincants (cas le plus fréquent).

Nos différents tests ont montré que quels que soient les jeux de données, la première phase apparie 80% des célibataires avec une précision supérieure à 99%. Au fur et à mesure des phases la contrainte de significativité est abaissée pour faciliter l'appariement des documents restants.

4.2 Résultats

Nous montrerons ici les résultats obtenus sur le corpus d'apprentissage, sur le corpus de test et sur la concaténation des deux corpus. Le système est rigoureusement le même pour chacune des pistes et pour chacun des corpus. Cela bien que nous aurions pu obtenir de meilleurs résultats avec quelques heuristiques locales. On peut remarquer que nous obtenons de moins bons résultats sur les articles tronqués. C'était assez attendu puisque le phénomène de recopie que nous recherchons est moins visible dans le développement. Comme nous n'avons pas souhaité concevoir un système différent selon les jeux de données, le modèle de document attendu détériore quelque peu les résultats.

Nous avions défini une *baseline* naïve qui consistait à former des couples uniquement selon le critère affinité-max, ses résultats sont faibles. Toutefois il est intéressant de noter la complémentarité des critères affinité-max et card-affinités , affinité-max évite certaines rares mauvaises décisions basées uniquement sur card-affinités (Tableau 3).

Critère	Affinité-max	Card-affinités	Combinaison
Piste 1: Article-résumé	0.626	0.975	1
Piste 2: Texte-résumé	0.48	0.934	0.959

Tableau 2: Corpus de test, score selon les critères utilisés

Le tableau 4 montre les résultats par corpus. Nous tenons à préciser que nos résultats sur le corpus de test sont supérieurs à ce que nous escomptions sur la piste 1. Nous avons donc fait un test en combinant les deux corpus. Remarquons que le résultat de ce test (corpus combinés) n'est pas la simple moyenne des résultats des corpus pris isolément. Nous n'avons pu faute de place intégrer nos tests sur des corpus non symétriques (nombre différents de résumés et d'articles mais ils montraient la même robustesse.

Le système n'utilise pas l'information sur la revue mais n'en souffre pas: il apparie toujours un article de la revue X à un résumé de la même revue dans le cadre du concours. Bien que l'indépendance vis à vis de la revue soit réelle il est intéressant de noter qu'une des revues, *Meta*, a concentré la très grande majorité des erreurs d'appariements et cela quels que soient les jeux de données. Le faible nombre d'affinités hapax rencontrés dans les articles de cette revue a été un facteur déterminant. La distribution des séquences utilisées a semblé plus homogène dans les différents articles issus de cette revue et a fortement nui aux appariements.

	Corpus d'apprentissage	Corpus de test	Corpus concaténés
Piste 1: Article-résumé	0.97	1	0.978
Piste 2: Texte-résumé	0.96	0.959	0.96

Tableau 3: Résultats selon les corpus

Les erreurs les plus fréquentes sur la seconde tâche provenaient là aussi du plus faible nombre d'affinités détectées par le système: sans l'introduction et la conclusion, un grand nombre d'affinités hapax disparaissent (Tableau 4). La différence entre un bon couple et un couple erroné tend à s'estomper et la

qualité des résultats s'en ressent. Comme nous l'avons évoqué, quelques paramètres bien choisis auraient sans doute permis d'atteindre un meilleur score dans la piste 2 mais nous avons voulu garder la simplicité et la reproductibilité comme objectifs primordiaux. Cela corrobore l'hypothèse des linguistes que les débuts et fins de segments sont en soi intéressants à exploiter, à différents niveaux de granularité (Lucas, 2009).

ID Résumés corpus de test	013.res	066.res	073.res	154.res	155.res
Card-affinités du bon couple article-résumé	58	54	76	71	49
Card-affinités du bon couple texte-résumé	42	42	49	33	37

Tableau 4: Nombre d'affinités du bon couple résumé-célibataire selon la piste.

5 Discussion

Nous avons présenté une méthode d'appariements d'articles et de résumés scientifiques basée sur des distributions de chaînes de caractères. Cette méthode a eu de très bons résultats sur la piste 1 qui concernait les articles complets. Le phénomène de recopie que nous cherchions à utiliser était par contre moins prégnant sur les documents de la seconde tâche, ce qui corrobore notre hypothèse fondée sur le distributionnalisme linguistique. Il nous semble que ces résultats apportent une pierre à l'édification de modèles alternatifs au "tout interprétable".

L'utilisation des chaînes de caractères mots ou non-mots revient quelque part à considérer l'espace typographique comme un caractère comme les autres et pas simplement comme une frontière immuable. Bien entendu il reste beaucoup à faire pour améliorer les résultats de ce genre d'approche mais nous pensons que la généricité multilingue en elle même, peut être exploitable dès le prochain défi fouille de textes, et qu'elle justifie les investigations dans cette voie.

Notre participation a porté sur la tâche 2, consistant à rapprocher un article de son résumé, et non sur la tâche 1, consistant à dater un extrait d'article. Ce choix n'est pas anodin puisque dans notre approche orientée modèle, « le texte est pour une linguistique évoluée l'unité *minimale*, et le corpus l'ensemble dans lequel cette unité prend son sens" (Rastier, 2002 ; Rastier, 2009). Alors que nous disposions d'un modèle de structuration des articles académiques suite aux travaux de (Lucas, 2004), et d'un modèle de structuration des articles de presse (Giguet & Lucas, 2004) dans la lignée des travaux de (Van Dijk, 88), nous ne disposions pas de modèles adaptés à la tâche 1, faute d'applications en lien avec des textes non suivis et éventuellement tronqués.

L'on pourrait bien entendu discuter du concept d'"appariement résumé/article" : en effet, un article nettoyé de son résumé reste-t-il vraiment un article académique ? L'on pourrait également s'interroger sur le fait qu'un article restructuré, voire"déstructuré", en XML, nettoyé de sa mise en page, de sa mise en forme, de ses figures ou de ses références, soit encore véritablement un article représentatif du genre académique.

On peut s'interroger plus globalement sur le fait que la déstructuration des documents soit un service rendu à la recherche en traitement des langues. Certes, cette option facilite l'entrée dans la compétition des participants mais n'a-t-elle pas une contrepartie insidieuse, un tribut peut-être un peu trop lourd à payer ? En restreignant le traitement des langues à un traitement littéral, n'est-ce pas ignorer l'importance de la sémiotique des formes non littérales dans la construction du sens ? N'est-ce pas laisser place à une vision de la langue fondamentalement ambiguë et à des traitements parfois inutilement combinatoires pour gérer ces ambiguïtés ? La plupart de ces ambiguïtés ne sont-elles pas que la résultante artificielle d'une vision peut être encore trop lexicale du traitement des documents ?

Au-delà de ces considérations épistémologiques, nous avons apprécié que des méta-informations comme le nom de la revue soient fournies. Non pas pour qu'elles soient systématiquement utilisées, mais pour que ce choix soit laissé aux participants, comme aurait pu être laissé au participant le choix de travailler soit à partir de la version XML du document, soit à partir de la version pdf texte+image. Cette option est notamment retenue par les organisateurs de l'ICDAR Booksearch Track (Doucet *et al*., 2009), a

laquelle nous participons en travaillant précisément a partir du PDF (Giguet, Baudrillart & Lucas, 2009). Dans la phase de mise au point, le nom de la revue a d'ailleurs permis de révéler la plus grande difficulté de notre approche à traiter la revue *Meta*. Nous n'avons cependant pas cherché à améliorer notre approche pour cette revue. Pour expliquer cette différence, on peut s'interroger sur la méthode de rédaction de ces résumés : sont-ils produits par l'auteur ou par le comité éditorial ? est-ce que des consignes particulières sont données, en terme de longueur ou de contenu ?

Nous avons également apprécié que la tâche choisie soit relativement simple. Les résultats des participants, homogènes en qualité, laissent finalement place à une discussion de fond sur les méthodes, sur leur légèreté, sur leur capacité à être appliquée à d'autres revues académiques, à d'autres genres où le résumé figure, à de quelconques autres langues, ou encore à des collections plus volumineuses. On constate en effet que lorsque la disparité des résultats est grande, l'attention sur la méthode est moins marquée pour les méthodes produisant des résultats dégradés, ce qui n'est bien sûr en aucun cas en lien avec la qualité des concepts sous-jacents. Typiquement, une résolution systémique nécessite un effort de conception et de développement qui n'est pas forcement valorisable par une évaluation en cours de mise au point.

Références

Brixtel R., Fontaine M., Lesner B., Bazin C., Robbes R. (2010), Language-Independent Clone Detection Applied to Plagiarism Detection in Tenth IEEE International Working Conference on Source Code Analysis and Manipulation. IEEE Computer Society, Timişoara, Romania. Pp 77-86

Church K., (2000) Empirical estimates of adaptation : The chance of two Noriegas is closer to *p/2* than p^2 . in *Coling 2000 Saarbrücken*, pp.173-179

Dejean H., (1998) Concepts et algorithmes pour la découverte des structures formelles des langues *Thèse de Doctorat*, Université de Caen.

Doucet A., Ahonen-Myka H. (2006) Fast extraction of discontiguous sequences in text : a new approach based on maximal frequent sequences in *Proceedings of IS-LTC 2006, Information Society - Language Technologies Conference*, Ljubljana, Slovenia, October 9-14, 2006, pp. 186-191.

Doucet A., Kazai G., Dresevic B., Uzelac A., Radakovic B. and Todic N. (2009) ICDAR 2009 Book Structure Extraction Competition in *Proceedings of the Tenth International Conference on Document Analysis and Recognition* (ICDAR'2009), Barcelona, Spain, July 26-29, pp.1408-1412.

Giguet E., Lucas N. (2004). La détection automatique des citations et des locuteurs dans les textes informatifs. *Le discours rapporté dans tous ses états : Question de frontières,* J. M. López-Muñoz, S. Marnette, L. Rosier (eds.). Paris, l'Harmattan, 2004, pp. 410-418.

Giguet E.,Luquet P.S. (2006). Multilingual lexical database generation from parallel texts in 20 languages with endogenous resources. Actes de *Coling 2006* Sydney. pp. 271-278.

Giguet E., Baudrillart A., Lucas N. (2009) Resurgence for the Book Structure Extraction Competition. in *INEX 2009 workshop preproceedings*. Brisbane, Australia. pp. 136-142.

Lardilleux A. (2011) Contribution des basses-fréquences à l'alignement sous-phrastique multilingue. *Thèse de Doctorat* Université de Caen.

Lecluze C. (2011) Recherche d'une granularité optimale pour l'alignement multilingue: N-grammes de caractères ou N-grammes de mots ? in *Actes des Journées Toulousaines, JeTou 2011*, Toulouse. pp. 147-151

Lejeune G., Doucet A., Lucas N. (2010a) Tentative d'approche multilingue en Extraction d'Information in *Analyse Statistiques des Donnés textuelles, JADT 2010* Rome pp. 1259-1268

LEJEUNE G., LUCAS N., DOUCET A. YANGARBER R. (2010b). Filtering news for epidemic surveillance: towards processing more languages with fewer ressources. In *Proceedings CLIA/COLING* Beijing. pp. 3-10

LUCAS N. *ET AL.*, (1993) Discourse analysis of scientific textbooks in Japanese : a tool for producing automatic summaries, in *Department of Computer Science Tokyo Institute of Technology, Technical report* 92TR-0004.

LUCAS N. ET CREMILLEUX B. (2004). Fouille de textes hiérarchisée, appliquée à la détection de fautes. *Document numérique vol. 8 n° 3* pp.107-133.

LUCAS N. (2004). The Enunciative Structure of News Dispatches: A Contrastive Rhetorical Approach. *Language, Culture, Rhetoric : Cultural and Rhetorical Perspectives on Communication*. Cornelia Ilie, 154-64. Stockholm: ASLA, 2004.

LUCAS, N. (2009a) Discourse Processing for Text Mining. in : *Information Retrieval in Biomedicine: Natural Language Processing for Knowledge Integration*, ed. by V. Prince & M. Roche, 229 - 62 Hershey, PA, USA: Medical Information Science Reference (imprint of IGI Global).

LUCAS, N. (2009b) Etude des textes en corpus et problèmes d'échelle. in *Corpus 8*. pp.197-220.

RASTIER F. (2002) "Enjeux épistémologiques de la linguistique de corpus," in *Journées de Linguistique de Corpus*, Lorient. http://www.revue-texto.net/Inedits/Rastier/Rastier_Enjeux.html

RASTIER F. (2009). *Sémantique interprétative*, Paris PUF.

TURMEL L., LUCAS N., CREMILLEUX B. (2003). Signalling well-written academic articles in an English corpus by text-mining techniques. *UCREL technical papers Vol. 16 special issue Proceedings Corpus Linguistics*, Lancaster University. pp. 465-474

VAN DIJK T.A, (1988). *News as discourse*, Lawrence Erlbaum Associates, Hillsdale N.J,

VERGNE J. (2001) Analyse Syntaxique Automatique De Langue: Du Combinatoire Au Calculatoire. in *TALN 2001, 8e conférence sur le traitement automatique des langues naturelles*, Tours.

INAOE at DEFT 2011: Using a Plagiarism Detection Method for Pairing Abstracts-Scientific Papers

Fernando Sánchez-Vega[1] Esaú Villatoro-Tello[1] Antonio Juárez-González[1]
Luis Villaseñor-Pineda[1] Manuel Montes-y-Gómez[1,2] Luis Meneses-Lerín[3]
(1) Laboratory of Language Technologies, Department of Computational Sciences,
National Institute of Astrophysics, Optics and Electronics (INAOE), Mexico.
(2) Department of Computer and Information Sciences,
University of Alabama at Birmingham.
(3) LDI (CNRS-UMR) Université Paris 13
{fer.callotl, villatoroe, antjug, villasen, mmontesg}@inaoep.mx

Résumé. Cet article décrit la méthode développée par le Laboratoire de Technologies Langagières de l'INAOE pour la tâche d'appariement résumés/articles dans le cadre de DEFT 2011. Pour aborder cette tâche, on a présupposé qu'un auteur emploi les mêmes expressions contenues dans le corps d'un article pour construire le résumé respectif. En conséquence, notre méthode cherche à retrouver les parties de texte réutilisé dans les résumés et les articles afin de déterminer le degré de dérivation entre eux. Notre méthode suit une stratégie non-supervisée qui ne dépend d'aucune ressources linguistiques, ce qui permet à notre méthode d'être générale et indépendante de la langue. Les résultats obtenus indiquent que le calcul de degré de dérivation entre deux documents peut être utilisé pour ce type de tâches.

Abstract. This paper describes the method developed by the Laboratory of Language Technologies of INAOE for the task of pairing abstracts with their respective scientific articles at the DEFT 2011 edition. Our main hypothesis is that authors commonly employ the same expressions contained in the body of the paper for constructing its respective abstract. Accordingly, our method focuses on the problem of finding portions of reused text between abstracts and papers in order to determine the degree of derivation between them. The proposed method is a non-supervised strategy that does not depend on any external linguistic resource, which allows our method to be general and language independent. Obtained results indicate that using the proposed method for determining the level of derivation between two documents is appropriate for the task of pairing abstracts-papers.

Mots-clés : DEFT 2011, Réutilisation de texte, Document Plagiarism, Indice de réécriture.

Keywords: DEFT 2011, Text reuse, Document Plagiarism, Rewriting Index.

1 Introduction

In order to solve the problem of pairing papers with their respective abstracts, our method assumes that an abstract is in fact a *summary* of the content of some scientific paper. Within such *summary*, normally the main ideas are mentioned reusing the same expressions (or even shorter versions) of those originally exposed in the body of the paper. Therefore, to solve this task we want to find the abstract that better represents the *summary* of some scientific paper. Hence, if we consider that authors will employ very similar (or even the same) expressions to those contained in the original paper for the construction of the abstract, we can model the problem of abstracts-papers pairing as particular case of plagiarism detection.

From a general point of view, document plagiarism detection involves finding similarities between any two documents which are more than just a coincidence and more likely to be result of copying (Clough, 2003). This is a very complex task since reused text is commonly modified with the aim of hide or camouflage the reused text.

The proposed method, which we call the *Rewriting Index*, is able to discover portions of text that have suffered some modifications such as word elimination, different word's order, word insertion and word substitution, allo-

wing to perform a partial matching between any two documents. Our goal was to show that using the *Rewriting Index* algorithm for computing documents' similarity, it is possible to achieve a better performance in the task of abstracts-papers pairing than only considering single words as the general degree of overlap. Is worth mentioning that the proposed method represents a non-supervised strategy (hence, it does not requires of a training phase) and it does not depend on any external linguistic resource, which allows our method to be language independent.

The rest of the paper is organized as follows. Section 2 presents some recent work on the task of plagiarism detection. Section 3 describes the proposed algorithm for finding portions of reused text. Section 4 presents the experimental configuration and results obtained. Finally, Section 5 depicts our conclusions and formulates some directions for future work.

2 Related Work

There are two major approaches for plagiarism detection (Ceska, 2007), which are : *simple* and *structural* approaches. The main variation among these two techniques consists in the strategy used to compute similarities between documents.

2.1 Simple approach

These techniques are called *simple* since they do not consider the structure of the text, and documents are commonly represented by means of their *bag of words* (BOW). Under this type of representation, documents' similarity is computed by some measure that only considers the words contained in both documents, hence, documents with high similarity degree are considered as plagiarized (Barron-Cedeno & Rosso, 2009; Hoad & Zobel, 2003; Zechner *et al.*, 2009). In (Metzler *et al.*, 2005) a similar approach is employed, where the BOW representation is constructed considering only the most frequent words.

Although the BOW strategies are very effective finding relevant documents to some particular text (*e.g.*, finding documents containing some text extracted from the suspicious document), they are not very effective finding documents where the reused text have suffered some modifications (*e.g.* when words are changed by others with similar meaning) which is a common practice of plagiarists. In addition, these strategies are affected by the thematic correspondence of the documents, which implies the existence of common domain-specific words, causing an overestimation of their overlap.

2.2 Structured approach

These type of approaches consider as key element for measuring similarities between documents the natural structure of language, such as the lexical similarity, the word's order and/or the word's *part-of-speech*. Structured techniques for detecting plagiarized documents can be divided into those that are dependent on some external resource and those that are totally independent.

2.2.1 Depending on external resources.

In (Si *et al.*, 1997) the entire structure of the document is considered for evaluation. One of the major drawbacks of this approach is that documents must be in a specific input format (*e.g.*, LaTeX) which includes some labels that facilitate the identification of certain sections of the document. Some other approaches that fit into this category are those that apply automatic translation tools to determine when two documents have high probabilities of been plagiarized (Chien-Ying *et al.*, 2010). Finally, in (Rehurek, 2008) syntactic trees are generated for each documents' sentence, and these trees are employed to determine how many sentences between two documents have the same structure or to identify those words that correspond to the same *part-of-speech*.

The main drawback of these approaches is the high dependency on external resources such as WordNet, or the existence of large and well balanced data sets for training the syntactic analysis and/or the automatic translation process, which drives to a language dependent approach.

2.2.2 Independent from external resources.

Methods employed within this category represent the language structure by means of the order and the adjacency of the words contained in the considered text. For this type of approaches there is no particular interest in knowing the words' *part-of-speech*, instead of that, these techniques try to capture the context of the words of interest. By doing this, it is possible to measure the quantity of reused text portions and also it is possible to know if it occurs in a similar context.

A common factor within these techniques is the text fragmentation, which consists on the generation of *chunks, shingles* or *n-grams* (Barron-Cedeno & Rosso, 2009; Basile *et al.*, 2009; Clough *et al.*, 2002). These parts of the text had some granularity which can be of different sizes (Hoad & Zobel, 2003), from a couple of characters, a few words, or even the entire document. The intuitive idea is to represent documents by using these text *chunks*. The main problem with these strategies is in deciding the size of the *chunk*, since small *chunks* can lead to a high number of repetitions even when there is no actual plagiarism, or if *chunks* are too large it will not be possible to identify those reused text that have suffered minor modifications, such as a different word order.

Our work differs from previous efforts in that our proposed approach, called the *Rewriting Index* (ReI), is able to discover text that have suffered some modifications such as word elimination, different word order, word insertion and word substitution, allowing to compute a partial matching between documents.

3 Plagiarism Detection Method

Generally, as stated above, common word sequences between the suspicious and source documents are considered the primary evidence of plagiarism. Nevertheless, using their presence as unique indicator of plagiarism is too risky, since thematic coincidences also tend to produce sequences of common text (*i.e.*, false positives). In addition, even a minor modification to hide the plagiarism will avoid the identification of the corresponding sequences, generating false negatives.

In order to handle these problems we propose a novel strategy for detecting plagiarised text, called the *Rewriting Index* method, which is able to identify portions of reused text even if the reused text have suffered some modifications.

In the following section we describe our algorithm for identifying and extracting the common text between the suspicious (D^S) and the source document (D^R). From here, the suspicious documents will be the *abstracts* and the source documents will be the *scientific papers*.

3.1 Identifying the reused text

Our proposed approach extracts strings (possible portions of reused text) considering a wide flexibility threshold and it is called the *Rewriting Index* method. This method allows to assign a weight value to each word contained in the suspicious document considering its degree of membership to a possible portion of plagiarized text. Hence, we are able to identify portions of text, that even if they do not represent an exact match, are in fact plagiarized strings. In other words, we are able to obtain non-consecutive portions of reused text ; therefore we are able to capture the common actions of plagiarist such as : word elimination, different word's order, word insertion and word interchange (*e.g.,* by synonyms).

In particular, the *Rewriting Index* method is an *ad-hoc* search algorithm that uses a vicinity (*i.e.,* context window) V that contains v words from the source document D^R (*i.e.*, V moves through the text of document D^R). The position of this vicinity window is defined by its central element which is called the *focus* element and we will refer to it as V^f. Accordingly, V^f will contain the word w^R_j, *i.e.,* the word w at the position j contained in the document D^R.

Additionally, the elements (words) at the right side from V^f will be defined as the V^+ elements, and in similar form, the elements at the left side from V^f will called as the V^- elements. Notice that the V^+ and V^- elements are within the context window. Finally, since our method considers the entire document D^R when searching for possible reused strings, we define as D^R_- to all the elements that appear at the left side of V, and as D^R_+ to all the

elements that appear at the right side of V.

According to these definitions, the *Rewriting Index* algorithm will assign a ReI value to each word w_i^S (*i.e.,* the word at position i within document D^S) depending on its position within D^R. The pseudocode for computing the ReI values for each word w_i^S is described in the Algorithm 1.

As it is possible to observe in the Algorithm 1, the ReI takes different values (c_i) depending on the position of the searched word w_i^S.That is, if the searched word appears at the V^f position the ReI is equal to 1 indicating a literal copying case ; if the word appears at the right of the *focus* it takes values c_2 or c_4 suggesting a moderate or a great number of deletion/insertion operations respectively ;if the word appears at le left of the *focus* it takes values c_3 or c_5 signifying a moderate or severe change on the word's order respectively ; finally, if the searched word does not appears in D^R, its ReI value is equal to 0. In general, the constants c_i fulfil the following condition :

$$1 > c_2 > c_3 > c_4 > c_5 > 0 \tag{1}$$

The *Rewriting Index* algorithm is able to provide the ReI value of each w_i^S (*i.e.,* to evaluate the complete document D^S) in a time proportional to $O(n)$ in the best case, considering n is the number of words contained in D^S, which means that the suspicious document represents an exact copy of D^R. In the other hand, the worst case will be when every w_i^S does not exists in D^R, which means there is no plagiarism, which leads to a time proportional to $O(nm)$ considering that m represents the number of words contained in D^R.

3.2 Computing the Rewriting Index measure

Previously we explained how to compute the ReI value for each word contained in D^S. However, the *Rewriting Index* measure f^{ReI} refers to a single value that represents how much the words from D^S are taken from D^R (*i.e.,* how much the words from the abstract are taken from the paper in evaluation). The definition of this similarity measure f^{ReI} is as follows :

$$f^{ReI} = \sum_{w_i^S \in D^S} \frac{ReI(w_i^S)}{\mid D^S \mid} \tag{2}$$

Notice that for computing the f^{ReI} measure we considered all the ReI values of every w_i^S.

4 Experiments and Results

4.1 DEFT 2011 Task 2 description

For the DEFT 2011 edition two main tasks were proposed, however we only participate in the Task 2, which main goal is to pair a scientific paper with an abstract. Two different modalities were proposed for this particular task :

1. **TRACK 1** - Pairing abstracts with full papers : For this track, participants were provided with abstracts and a complete version of several papers, *i.e.*, papers that contain all their original sections, such as introduction, related work, evaluation, discussion, conclusions, etc.).
2. **TRACK 2** - Pairing abstracts with incomplete papers : Contrary to the previous track, for this exercise papers do not contain the introduction and conclusion sections.

4.2 Corpus

The provided corpus is composed of scientific papers mainly published in journals from the humanities field (Anthropologie et Sociétés, Études internationales, Études littéraires, Meta, Revue des sciences de l'éducation), all of them in French.

The training corpus is composed of 300 abstract and 300 papers from 5 different reviews in the humanities (one abstract per file and one article per file). Notice that for TRACK 2 papers were reduced by eliminating the introduction and conclusions sections. And for the test corpus, 200 abstract and 200 papers were given, adding papers from a 6th review that was not part of the training set.

input : word w_i^S, source document D^R and the vicinity window V
output: the rewriting index score of w_i

//Enters if the searched word w_i^S is equal to the one in V^f, and as a result moves V^f one position and the ReI gets the higher value
if $(w_i^S = V^f)$ **then**
    $V^f \leftarrow V^f + 1$;
    $ReI(w_i^S) \leftarrow 1$;
end
//Enters if searched word w_i^S appears at the right of V^f, and as a results moves V^f to the next position where w_i^S exists in D^R and the ReI value is assigned to constant c_2
else if $(w_i^S$ *appears in* $V^+)$ **then**
    $V^f \leftarrow position_{D^R}(w_i^S) + 1$;
    $ReI(w_i^S) \leftarrow c_2$;
end
//Enters if searched word w_i^S appears at the left of V^f, and as a result V^f remains at the same position and ReI is assigned to constant c_3
else if $(w_i^S$ *appears in* $V^-)$ **then**
    $V^f \leftarrow V^f$;
    $ReI(w_i^S) \leftarrow c_3$;
end
//Enters if searched word w_i^S appears at the right side of V, as a result ReI is assigned to the constant c_3
else if $(w_i^S$ *appears in* $D_+^R)$ **then**
    //Enters if there is more than one single word coincidence in the D_+^R region, and as a result the V^f element is moved to position where the coincidence was found, and the V^- elements are updated for those that previously were in V^+
    if $(w_{i+1}^S = w^R_{position_{D^R}(w_i^S)+1})$ **then**
        $temp \leftarrow V^+$;
        $V^f \leftarrow position_{D^R}(w_i^S) + 1$;
        $V^- \leftarrow temp$;
    end
    $ReI(w_i^S) \leftarrow c_4$;
end
//Enters if searched word w_i^S appears at the left side of V, as a result V^f remains the same and ReI is assigned to the constant c_4
else if $(w_i^S$ *appears in* $D_-^R)$ **then**
    $V^f \leftarrow V^f$;
    $ReI(w_i^S) \leftarrow c_5$;
end
//Enters if the searched word w_i^S does not exists in D^R, as a result ReI is assigned to 0
else
    $ReI(w_i^S) \leftarrow 0$;
end
Exit;

Algorithm 1: The *Rewriting Index* evaluation algorithm

4.3 Proposed method configuration

The strategy followed for finding abstracts-papers pairs was : for each pair abstract-paper we computed the f^{ReI} measure (See expression 2). Afterwards, we performed a ranking process considering the f^{ReI} measure, resulting in an ordered list. Hence, the three best evaluated papers for each abstract are presented as the owners of the respective abstract.

Notice that in the proposed algorithm (Algorithm 1) there are a couple of parameters that have not been defined : the size of the window V and the values of the constants c_i. For the later one, these were defined as : $c_i = 1/i$. Notice that such definition do not violate the constraint defined in expression 1, resulting in the following values :

$$1 > \frac{1}{2} > \frac{1}{3} > \frac{1}{4} > \frac{1}{5} > 0 \tag{3}$$

And with respect to the size of V, we probed with 2 different sizes (5, and 11). Therefore, our experiments labelled as **RUN-1** refer to a context window V of size 5, meanwhile experiments labelled as **RUN-2** refer to a context window V of size 11.

4.4 Baseline method configuration

As our baseline method we employed the well known tool ROUGE (Chin-Yew, 2004) which is mainly oriented to the automatic evaluation of summaries. ROUGE is a tool that is able to measure word's co-occurrences between two documents[1] indicating somehow the degree of overlap between evaluated documents.

ROUGE is able to measure co-occurrences of single words (*i.e.,*$1 - grams$) up to $4 - grams$ (ROUGE-N), the intuition is that the greater the length of n, the better the estimation of the fluency between evaluated documents. Additionally, the ROUGE tool also proposes measuring the co-occurrences of *Longest Common Subsequences* (ROUGE-L), where the intuition is that the longer the LCS of two documents is, the more similar the two documents are.

For our experiments we assign a ROUGE score (R) to each pair of abstract-paper which is computed as follows :

$$R(a,p) = \frac{ROUGE-1(a,p) + ROUGE-2(a,p) + ROUGE-L(a-p)}{3} \tag{4}$$

where a refers to an abstract and p to some paper. Hence, once we have computed the R value for every pair of abstract-paper we kept the three papers that obtain a higher value of R for generating the proposed solution, *i.e.,* the three best evaluated papers for each abstract are presented as those with higher probabilities of being the owners of the respective abstract. Experiments performed using the baseline method are labelled as **RUN-3**.

4.5 Results

Since our proposed approach does not requires any training information, we only present obtained results over the test corpus. Table 1 shows the results from the proposed approach for both tracks. As it is possible to observe, the proposed method using a context window V of size 5 (RUN-1) allows to obtain a better accuracy performance for both tracks.

Notice that RUN-1 also outperforms the baseline method (RUN-3) which is a configuration based on word and common sequences co-occurrences. As expected, results from TRACK-2 are lower than those obtained in TRACK-1, which is a clear indicator of how common is the use of expressions contained in both the *Introduction* and *Conclusions* sections for the construction of the abstract.

[1] Ideally between summary pairs.

Track	Run ID	Accuracy
TRACK 1	RUN-1	0.970
	RUN-2	0.960
	RUN-3	0.949
TRACK 2	RUN-1	0.904
	RUN-2	0.848
	RUN-3	0.858

TAB. 1 – Results of the proposed approach

5 Conclusions

In this paper we propose to solve the problem of abstracts-papers pairing by means of a method design for detecting document plagiarism. Our method focuses on the detection of common (possible reused) strings between the source and suspicious documents.

The proposed algorithm, called the *Rewriting Index* method allows to assign a value to each word contained in the suspicious document considering its degree of membership to a possible portion of plagiarized text. By employing the *Rewriting Index* method we are able to identify portions of text, that even if they do not represent an exact match are in fact reused strings, therefore we are able to capture the common actions of plagiarist such as : word elimination, different word's order, word insertion and word interchange.

Experimental results on the DEFT corpus are encouraging ; they indicate that the proposed method for identifying portions of reused text, and measuring similarities between documents are appropriate for the abstracts-papers pairing task. Results also demonstrate that using word or even common sequences co-occurrences is insufficient for this task, conducting to several false positives cases.

Acknowledgments.
This work was done under partial support of CONACyT (Project grants 106013, 134186 and scholarship 224483 - 258345).

Références

BARRON-CEDENO A. & ROSSO P. (2009). On automatic plagiarism detection based on *n*-grams comparison. In *Proceedings of the SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 2009)*, Donostia-San Sebastian, Spain.

BASILE C., BENEDETTO D., CAGLIOTI E., CRISTADORO G. & ESPOSTI M. D. (2009). A plagiarism detection procedure in three steps : Selection, matches and "squares". In *Proceedings of the SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 2009)*, Donostia-San Sebastian, Spain.

CESKA Z. (2007). The feature of copy detection techniques. In *Proceedings of the 1st Young Researches Conference on Applied Sciences (YRCAS 2007)*, p. 5–10, Pilsen, Czech Republic.

CHIEN-YING C., JEN-YUAN Y. & HAO-REN K. (2010). Plagiarism detection using rouge and wordnet. *Journal of Computing.*, **2**(3), 34–44.

CHIN-YEW L. (2004). Rouge : a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, Barcelona, Spain.

CLOUGH P. (2003). Old a new challenges in automatic plagiarism detection. *National Plagiarism Advisory Service*, p. 391–407.

CLOUGH P., GAIZAUSKAS R., PIAO S. & WILKS Y. (2002). Meter : Measuring text reuse. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, USA.

HOAD T. C. & ZOBEL J. (2003). Methods for identifying versioned and plagiarized documents. *Journal of the American Society for Information Science and Technology*, **54**(3), 203–215.

METZLER D., BERNSTEIN Y., CROFT W., MOFFAT A. & ZOBEL J. (2005). Similarity measures for tracking information flow. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM'05)*, p. 517–524.

REHUREK R. (2008). Plagiarism detection through vector space models applied to a digital library. In *Proceedings of Recent Advances in Slavonic Natural Language Processing.*, p. 75–83.

SI A., LEONG H. V. & LAU R. W. H. (1997). Check : A document plagiarism detection system. In *Proceedings of the 1997 ACM Symposium an Applied Computing.*, p. 70–77, San Jose CA, USA.

ZECHNER M., MUHR M., KERN R. & GRANITZER M. (2009). External and intrinsic plagiarism detection using vector space models. In *Proceedings of the SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 2009)*, Donostia-San Sebastian, Spain.

Matching documents and summaries using key-concepts

Sara Tonelli, Emanuele Pianta
Fondazione Bruno Kessler, Via Sommarive 18, Povo (Trento) Italy
satonelli@fbk.eu, pianta@fbk.eu

Résumé. Nous présentons une méthodologie pour trouver le meilleur appariement d'articles scientifiques et de résumés en trois étapes : la première étape consiste à extraire d'un document une série de concepts-clé, pour donner une représentation concise de son contenu. Ensuite, la liste de concepts-clé est associée à chaque résumé en assignant un score de similarité inspiré par les critères de mesure standard : Précision, Rappel et F1. Enfin, étant donné un graphe biparti pondéré qui représente tous les possibles paires document-résumé, on exécute un algorithme qui trouve le meilleur appariement. Nous évaluons notre approche sur le corpus de test de DEFT, et montrons qu'il obtient de bons résultats (la configuration la plus performante obtient 0.99 F1). Nous exposons en détail plusieurs avantages de cette approche. Par exemple, il ne requiert pas de corpus d'entrainement, mais uniquement un corpus de développement pour le réglage des paramètres. En plus, l'utilisateur peut configurer facilement le type de concepts-clé à extraire, selon des paramètre allant de leur longueur maximale à leur degré de spécificité.

Abstract. We present a methodology to find the best document - summary match based on three steps : first, a set of key-concepts is extracted from the document in order to give a concise representation of its content. Then, the key-concept list is compared with each abstract by assigning a similarity score inspired by the standard metrics of Precision, Recall and F1. Finally, an algorithm is run that, given a weighted bipartite graph representing all possible document - summary pairs, finds the best matches. We evaluate our approach on the DEFT test data and we show that it achieves good results, with 0.99 F1 in the best performing configuration. We also detail some advantages of this approach. For example, it does not require training data except for a development set for parameter tuning. Besides, the user can easily configure the type of key-concepts to be extracted, from their maximum length to the degree of specificity.

Mots-clés : Extraction de concepts-clé, mesures de similarité, appariement de graphes.

Keywords: Key-concept extraction, similarity metrics, graph-matching techniques.

1 Introduction

Key-concepts are simple words or phrases that provide an approximate but useful characterization of the content of a document and offer a good basis for applying content-based similarity functions. In general, key-concepts can be used in a number of interesting ways both for human and automatic processing. For example, a quick topic search can be carried out over a number of documents indexed according to their key-concepts, which is more precise and efficient than full-text search. Also, key-concepts can also be used to calculate semantic similarity between documents and to cluster the texts according to such similarity (Ricca *et al.*, 2004). Furthermore, key-concepts provide a sort of quick summary of a document, thus they can be used as an intermediate step in *extractive* summarisation in order to identify the text segments that reflect the content of a document. (Jones *et al.*, 2002), for example, exploit key-concepts to rank the sentences in a document by relevance in that they count the number of key-concept stems occurring in each sentence. In the light of the increasing importance of key-concepts in different applications, from search engines to digital libraries, a recent task for the evaluation of key-concept quality was also proposed at the last SemEval-2010 campaign (Kim *et al.*, 2010)

In this work, we assume that the list of n-topmost relevant key-concepts of a text can be used as a sort of summary of the document. We will call this list a *key-concept summary*. We make the hypothesis that, in order to select the abstract that corresponds to an article, we need to find the abstract that is most similar to the key-concept summary extracted from such article. This is achieved by applying a *similarity function* that tries to measure the degree of coherence and the completeness between the key-concept summary and the abstract.

The paper is structured as follows : in Section 2 we provide an overview of the system and briefly describe the four main steps it includes. In Section 3 we detail the architecture of KX, the key-concept extraction tool, providing an insight into its parameter configuration. In Section 4 we present the similarity function applied to weight each possible article/abstract combination, while in Section 5 we describe the algorithm devised for finding the best article/abstract pairs. In Section 6 we detail the experimental setup and the workflow optimization based on training data. Besides, we describe the final setup employed in the test phase and we comment on the evaluation results. Finally, we draw some conclusions and discuss some future improvements of our approach in Section 7.

2 General system description

The algorithm for abstract - article matching that we present includes four steps, as illustrated in Figure 1.

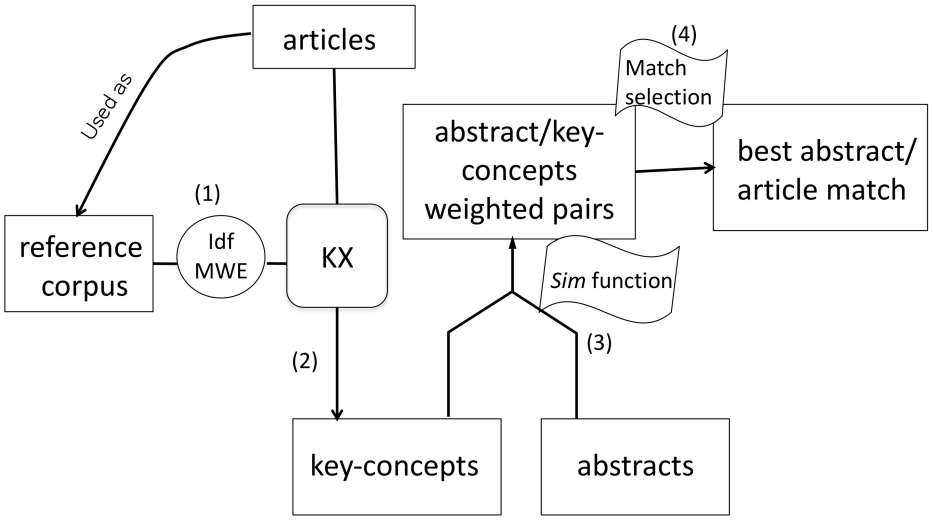


FIGURE 1 – Abstract/article matching workflow

KX (Pianta & Tonelli, 2010), the key-concept extraction component, is involved in the two initial steps. The *first*

one is optional and concerns the extraction of relevant statistical information from a reference corpus. In particular, the tool extracts information about plausible multiword expressions (MWEs) from such corpus and calculates the inverse document frequency of the key-concepts found in each document of the corpus. In the DEFT competition, the reference corpus adopted includes 996 documents, i.e. 300 articles and 300 abstracts from the training set, and 198 articles and 198 abstracts from the test set.

In the *second step*, KX is run over each article to be paired with an abstract in order to extract from each article the corresponding key-concept summary. Details about KX parameter configuration are given in Section 3. Note that KX is an unsupervided system, so the training set is only used to find the best parameter combination for the task and to tune the system for handling French data, but no information about the corresponding abstracts is taken into account.

In a *third step*, for each key-concept summary representing an article and each abstract in the data set a similarity score is computed. In this way, we obtain a complete weighted bipartite graph where the two vertex sets correspond to the articles and to the abstracts, and each article is connected to each abstract by an edge with a weight ≥ 0, assigned by the similarity function.

In the *last* step, we run an algorithm that, given the complete bipartite graph, finds the subgraph corresponding to the best possible match between articles and abstract, in which each vertex can be connected only to another vertex by a single edge.

3 KX : a flexible Key-concepts eXtractor

In this section, we thoroughly describe the basic KX architecture for unsupervised key-concept extraction. With KX, the identification of key-concepts can be accomplished with or without the help of a reference corpus, from which some statistical measures are computed in an unsupervised way. KX is distributed with the TextPro NLP Suite (Pianta *et al.*, 2008) and the current version can handle English and Italian texts. Nevertheless, since the only language-dependent part of the system is the morphological analyzer, we developed for the DEFT competition a new version for French documents, where the morphological analysis is disabled and is replaced by an extensive use of black lists. For details, see the following description.

3.1 Key-concept extraction at document level

Figure 2 displays the key-concept extraction process that, starting from a document, outputs a list of key-concepts ranked by relevance. The same workflow applies both to the extraction of key-concepts from a single document and to the extraction of relevant statistical information about multiwords and key-concepts from a *reference corpus*, which can be optionally used as additional informaton when processing a single document. For more information, see below and in Section 3.2.

The system takes a document in input and first performs tokenization. Then, all possible n-grams composed by any token sequence are extracted, for instance ‘éclipse de soleil’, ‘tous les’, ‘où chacun’. The max length of the selected n-grams can be set by the user and for DEFT it was set to four.

Then, from the n-gram list a sublist of *multiword expressions (MWE)* is derived, i.e. combinations of words expressing a unitary concept, for example ‘compréhension de concepts’ or ‘expression métaphorique’.

In the selection step, the user can choose to rely only on local (document) evidence or to make use also of global (corpus) evidence. As for the first case, a frequency threshold called $MinDoc$ can be set, which corresponds to the minimum number of occurrences of n-grams in the current document. If a reference corpus is also available, another threshold can be added, $MinCorpus$, which corresponds to the minimum number of occurrences of an n-gram in the corpus. KX marks an n-gram in a document as a multiword term if it occurs at least $MinCorpus$ times in the corpus or at least $MinDoc$ times in the document. The two parameters depend on the size of the reference corpus and the document respectively. In our case, the corpus was the set of documents and abstracts made available in the DEFT competition. More details about the thresholds applied can be found in Section 6.

A similar, frequency-based, strategy is used to solve ambiguities in how sequences of contiguous multiwords should be segmented. For instance, given the sequence ‘retour des bonnes manières’ we need to decide whether we

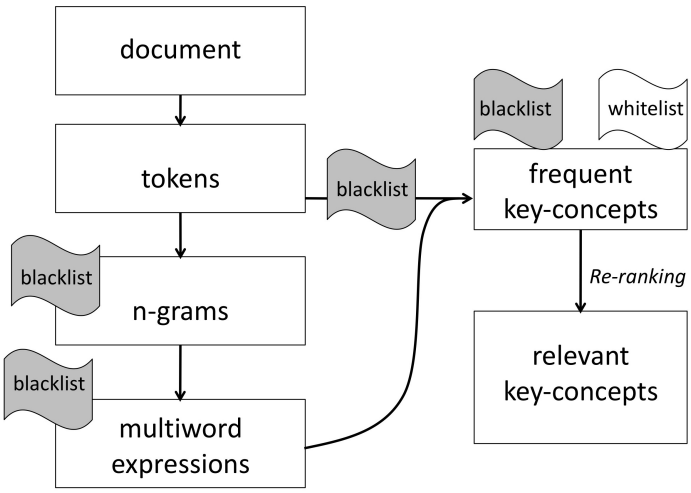


FIGURE 2 – key-concept extraction process

recognize ‘retour des bonnes’ or ‘bonnes manières’. To this purpose, we calculate the strength of each alternative MWE as

$$Strength_{colloc} = docFrequency * corpusFrequency$$

and then choose the stronger one.

In the next step, the single words and the MWEs are ranked by frequency in order to obtain a first list of key-concepts. Thus, frequency is the baseline ranking parameter, based on the assumption that important concepts are mentioned more frequently than less important ones. Frequency is normalized by dividing the number of key-concept occurrencies by the total number of tokens in the current document.

Note that the first key-concept list is obtained by applying *black and white lists* almost at every step of the process, as shown in Fig. 2. This is particularly relevant to the DEFT task, in which we have used a system configuration that does not include the morphological filter usually employed to select the linguistic patterns of MWEs. A black list is applied for discarding n-grams containing one of the language-specific stopwords defined by the user, for example ‘avons’, ‘peut’, ‘puis’, ‘parce’. Also single words corresponding to stopwords are discarded when the most frequent tokens are included into the first key-concept list. For example, in French we may want to exclude all key-concepts containing the word ‘toi’, ‘très’, ‘finalement’, etc.

A black list is applied also when selecting MWEs from n-grams. The list can be enriched after running the first system tests, in order to eliminate multiwords that have been selected by mistake. For example, it may be useful to discard n-grams such as ‘de plus en plus’, ‘par exemple’, ‘en effet’, etc.

Finally, black and white lists can be manually compiled also for key-concepts, in order to define expressions that should never be selected as relevant key-concepts as well as terms that should always be included in the key-concept rank. For example, the preposition ‘de’ is not included in the stopword list because key-concepts like ‘problème de reformulation’ should be admitted. However, ‘de’ is very frequent in documents, so it can happen that it is selected as single-word key-concept. In order to avoid this, ‘de’ can be included in the key-concept black list.

3.2 Parameters for first key-concept ranking

After the creation of a frequency-based list of key-concepts, various techniques are used to re-rank it according to relevance. If a reference corpus is available, as in our case, additional information can be used to understand which key-concepts are more specific to a document, and therefore are more likely to be relevant for such document.

In order to find the best ranking mechanism, and to tailor it to the type of key-concepts we want to extract, the following parameters can be set :

Key-concept IDF : This parameter takes into account the fact that, given a data collection, a concept that is mentioned in many documents is less relevant to our task than a concept occurring in few documents. In order to activate it, a reference corpus must undergo a pre-processing step in which the key-concepts are extracted from each document in the corpus and the corresponding inverse document frequency (IDF) is computed following the standard formula :

$$IDF_k = log\frac{N}{DF_k}$$

where N is the total number of documents and DF_k is the number of documents in the corpus that contain the key-concept k. The IDF of a rare term tends to be high, while the IDF of a frequent one is likely to be low. Therefore, IDF may be a good indicator for distinguishing between common, generic terms and specific ones, which are good candidates for being a key-concept.

When this parameter is activated, for each key-concept found in the current document, its IDF computed over the reference corpus is retrieved and multiplied by the key-concept frequency at document level.

Key-concept length : Number of tokens in a key-concept. Concepts expressed by longer phrases are expected to be more specific, and thus more informative. When this parameter is activated, frequency is multiplied by the key-concept length. For example, if 'expression verbale' has frequency 6 and 'expression verbale des émotions' has frequency 5, the activation of the key-concept length parameter gives 'expression verbale' = 6 * 2 = 12 and 'expression verbale des émotions' = 5 * 4 = 20. In this way, the 4-gram is assigned a higher ranking than the 2-gram.

Position of first occurrence : Important concepts are expected to be mentioned before less relevant ones. If the parameter is activated, the frequency score will be multiplied by the $PosFact$ factor computed as :

$$PosFact = \left(\frac{DistFromEnd}{MaxIndex}\right)^2$$

where $MaxIndex$ is the length of the current document and $DistFromEnd$ is $MaxIndex$ minus the position of the first key-concept occurrence in the text.

The parameters can be independently activated by the user in a configuration file. The key-concept relevance is then calculated by multiplying the normalized frequency of a key-concept by the score obtained by each active parameter. We eventually obtain a ranking of key-concept ordered by relevance. The user can also set the number of top ranked key-concepts to consider as best candidates.

3.3 Parameters for final ranking

After the first ranking described in Section 3.2, a further set of operations can be optionally carried out by the system in order to adjust the preliminary ranking. Again, such operations can be independently activated through a separate configuration file. The parameters have been introduced to deal with so-called *nested* key-concepts (Frantzi *et al.*, 2000), i.e. those that appear within other longer candidate key-concepts. After the first ranking, which is still influenced by the key-concept frequency, *nested* (shorter) key-concepts tend to have a higher ranking than the containing (longer) ones, because the former are usually more frequent than the latter. However, in some settings, for example in scientific articles, longer key-concepts are generally preferred over shorter ones because they are more informative and specific. In such cases, the user may want to adjust the ranking in order to give preference to longer key-concepts and to reduce or set to zero the score of nested key-concepts.These operations are allowed by activating the following parameters :

Shorter concept subsumption : It happens that two concepts can occur in the key-concept list, such that one is a specification of the other. Concept *subsumption* and *boosting* (see below) are used to merge or rerank such couples of concepts. If a key-concept is (stringwise) included in a longer key-concept with a higher frequency-based score, the score of the shorter key-concept is transferred to the count of the longer one.

For example, if 'expression verbale' has frequency 4 and 'expression verbale des émotions' has frequency 6, by activating this parameter the relevance of 'expression verbale des émotions' is 6 + 4 = 10, while the relevance of 'expression verbale' is set to zero. The idea behind this strategy is that nested key-concepts can be deleted from the final key-concept list without losing relevant information, since their meaning is nevertheless contained in the longer key-concepts.

Longer concept boosting : This parameter applies in case a key-concept is (stringwise) included in a longer key-concept with a lower relevance. Its activation should better balance the ranking in order to take into account that longer n-grams are generally less frequent, but not less relevant, than shorter ones. The parameter is available in two different versions, having different criteria for computing such boosting. With the *first option*, the average score between the two key-concepts relevance is computed. Such score is assigned to the less frequent key-concepts and subtracted from the frequency score of the higher ranked one. With the *second option*, the longer key-concepts is assigned the frequency of the shorter one. In none of the two variants key-concepts is deleted from the relevance list, as it happens by activating the *Shorter concept subsumption* parameter.

For example, if 'expression verbale' has score 6 and 'expression verbale des émotions' has score 4, by activating the first option of this parameter the relevance of 'expression verbale' becomes 6 - ((6 + 4) / 2) = 1, while the relevance of 'expression verbale des émotions' is set to 5, i.e. (6 + 4) / 2 .

With the second option, both the relevance of 'expression verbale des émotions' and of 'expression verbale' is set to 6.

As shown in the examples above, these parameters can change the output of the ranking by deleting some entries and boosting some others. Note that after applying one cycle of subsumption/boosting, the order of the concepts can dramatically change, producing the conditions for further subsumption/boosting of concepts.

The number of iterations for the application of this re-ranking mechanism can be set by the user, and each cycle increases the impact of the re-ranking on the key-concept list. The parameters can be activated together and in different combinations. If all parameters are set, the short concept subsumption procedure is applied first, then the longer concept boosting is run on the output of the first re-ranking, so that the initial relevance-based list goes through two reordering steps.

4 Similarity score assignment

After a key-concept summary has been extracted from each document, we compute a similarity score between such summary and each candidate abstract. The final goal is to match the document/abstract pair achieving the highest similarity. In particular, we believe that the similarity between a key-concept summary and an abstract can be measured in terms of *coherence*, i.e. the amount of information in the key-concept summary that is present also in the abstract, and *completeness*, i.e. the amount of information in the abstract that is represented through the key-concept summary. In other words, the key-concept summary should ideally contain *all and only* the information in the abstract.

The coherence and completeness criteria can be cast in terms of *precision* and *recall* of the key-concepts extracted from a text, taking a manual abstract as gold standard of the extraction task. This approach is inspired by the methodology originally proposed by (Pianta, 2011) for the *PatExpert* system [1] to evaluate the quality of the key-concepts extracted from a patent document. Since every patent in the PatExpert collection was provided with an abstract, the author proposed to compute the *precision* of a key-concept summary as the percentage of the top-ranked key-concepts that are mentioned in the abstract, and *recall* as the percentage of words in the abstract that appear among the top-ranked key-concepts. Then, F1 was additionally computed following the standard formula :

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

We rely on the same assumption that high precision, recall and F1 correspond to a high similarity between key-concept summary and abstract, but apply the metrics as a similarity score instead of as an evaluation metric.

1. http://www.patexpert.org/

Therefore, we refer to them as Sim_{Prec}, Sim_{Rec} and Sim_{F_1}. Following this idea, we consider Sim_{Prec} as a measure of coherence, and we define it as :

The percentage of n-grams in the key-concept summary that are contained also in the abstract.

On the other hand, we see Sim_{Rec} as a measure of completeness and we define it as :

The percentage of tokens in the abstract, excluding stopwords, that occur also in the key-concept summary.

After carrying out some tests on the DEFT training set, we assess that Sim_{F_1} is the best similarity score to use to our purpose, while Sim_{Prec} and Sim_{Rec} alone do not achieve comparably satisfactory results.

Sim_{Prec} and Sim_{Rec} are calculated according to the procedure described in Algorithm 1 :

Algorithm 1 Compute Sim_{Prec} and Sim_{Rec} for Sim_{F_1} assignment

Require: Key-concept summary K and abstract A
  initialise $count = 0$
  for all $k \in K$ **do**
    if k appears in A **then**
      $count \Leftarrow count + 1$
      remove all occurrences of k in A
    else
      repeat
        take longest sub-phrase $sub \in k$
      until
      sub appears in A
      $count \Leftarrow count + \frac{subLength}{kLength}$
      remove all occurrences of sub in A
    end if
  end for
  $Sim_{Prec} = \frac{count}{|K|}$
  remove stopwords from A
  $Sim_{Rec} = \frac{AInitialLength - AFinalLength}{AInitialLength}$

Given the set of key-concepts K extracted from an article and an abstract A, we compare them as follows : if $k \in K$ matches a phrase in A, than we remove all occurrences of such phrase from A. Otherwise we look for the longest subphrase of k matching a phrase in A, and again if a match is found, all matching phrases are removed. If we can match a full key-concept this is counted as 1 score. If we can only match a subphrase, we count a fraction of 1 depending on the length of the matching sub-phrase. For instance if the full key-concept includes 3 tokens and we can only match a subphrase of length 2, the score is calculated as 2/3.

Once this match-and-delete procedure is carried out for the top-ranked key-concepts, we calculate the percentage of $k \in K$ that are included in A, which corresponds to Sim_{Prec} between K and A. Then, we take the words that are left after the deletion of matching key-concepts and subphrases, we further delete any stopword, and we compare the length in words of the abstract before and after the match-and-delete procedure (i.e. $AFinalLength$ vs. $AInitialLength$). The number of deleted words ($AInitialLength - AFinalLength$) divided by the number of tokens in the abstract ($AInitialLength$) excluding stopwords gives an estimation of the concepts in the abstract that have been actually extracted by KX. In short, it measures the recall of the algorithm. In this way, we calculate also the Sim_{Rec} for each K - A pair. Finally, we use the two measures to compute Sim_{F_1}.

5 Algorithms for best match selection

After assigning a similarity score to each possible pair of key-concept summary and abstract, we need to decide which are the best document - abstract matches. Since the output of the previous step is composed of two partitions,

the documents D and the abstract collection AC, where each source node $d \in D$ is connected to each target node $a \in AC$ and vice versa by a weighted link, we consider the pair selection task as a matching problem in a bipartite graph.

More formally, a weighted bipartite graph is a graph $G = (V, E)$ where V is the node set partitioned into two nonempty sets V_1 and V_2, and every node edge $e \in E$ connects a node in V_1 to a node in V_2 with a weight. In our task, V_1 and V_2 correspond respectively to the set of documents D and the set of abstracts AC. Besides, the graph G is *complete*, i.e. all nodes in D and in AC are connected to each other by a weight ≥ 0 computed as described in Section 4. Such weight expresses the similarity between a document and an abstract.

The initial configuration of the complete graph is riported in Figure 3 (left graph). Our goal is to reduce the graph to a perfect matching subgraph, where each $d \in D$ is mapped to one $a \in AC$ and vice versa (Fig. 3, right graph).

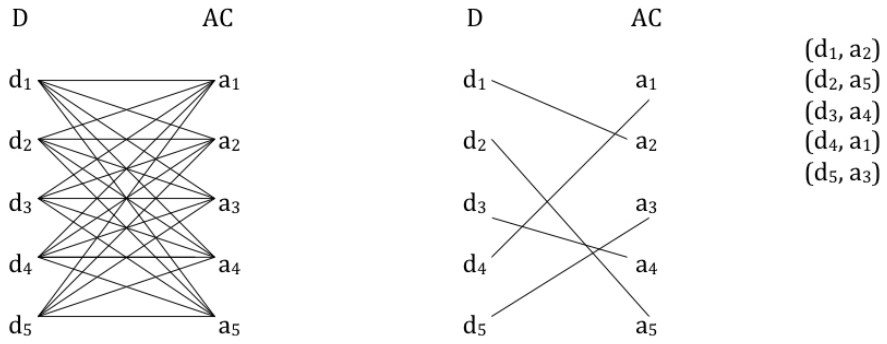


FIGURE 3 – From a complete bipartite graph to a perfect matching between (d_i, a_i) pairs

In order to select the best (d_i, a_i) pairs, we test two different matching algorithms : one is the so-called *Hungarian Algorithm* (Kuhn, 1955), a classical combinatorial optimization algorithm used for maximum assignment problems [2]. This algorithm is exploited when, given a complete weighted bipartite graph, it is necessary to find a perfect matching with maximum cost, i.e. the subgraph that maximizes the total weight of the matches. The intuition behind it is to find the set of edges that at *global* level achieves the highest weight.

We also test a second approach, which we call *Local Match* (*LocMatch*), aimed at finding the set of edges that achieves the highest weight at *local* level. In short, we start the match by choosing the (d_i, a_i) pair having the highest weight in the graph, and we iteratively repeat the assignment by choosing at each stage the edge with the maximal increase of the weight.

In order to explain the different behaviour of the two algorithms, we report an example in Fig. 4. Given a set of sample documents $\{d_1, d_2, d_3\}$ and a set of abstracts $\{a_1, a_2, a_3\}$ connected by weighted edges, we can represent this graph through a rectangular matrix, where the weight associated to each link is reported. The matches assigned by the algorithm are highlighted in grey. With the *Hungarian Algorithm* we obtain the perfect match represented in the central matrix of Fig. 4. In this case, the algorithm finds all the edges with the optimal total weight, i.e. $8 + 4 + 5 = 17$.

With the second algorithm (matrix on the right), we start from matching the (d_i, a_i) pair with the highest weight, i.e. d_1 and a_3. Consequently, we delete the third columns and the first row of the matrix, because d_1 and a_3 cannot be paired with other candidates. Then, we match the pair with the highest weight in the remaining matrix, i.e. d_2 and a_1. Finally, the only possible pairing left is between d_3 and a_2. In this case, the total match weight is $8 + 6 + 2 = 16$, which is lower than the weight obtained with the other algorithm. However, more relevance is given to weights at *local* level.

2. We use the Perl implementation available at `http://search.cpan.org/dist/Algorithm-Munkres/`

	a_1	a_2	a_3
d_1	3	0	8
d_2	6	4	3
d_3	5	2	7

	a_1	a_2	a_3
d_1	3	0	8
d_2	6	4	3
d_3	5	2	7

	a_1	a_2	a_3
d_1	3	0	8
d_2	6	4	3
d_3	5	2	7

FIGURE 4 – Matrix representing the initial complete bipartite graph (left), assignments done with the *Hungarian algorithm* (middle) and with the second matching algorithm (right).

6 Experimental Setup and Evaluation

In the DEFT task for “Abstract/article matching", two subtasks are proposed : the first one consists in identifying the best matches between abstracts and full scientific articles, while in the second subtask the matches must be found between abstracts and articles where introduction and conclusions have been removed.

The training corpus comprises 300 abstracts and 300 articles (full and shortened) from 5 different journals, explicitly indicated for each abstract and each article. The test corpus comprises 198 abstracts and 198 articles from 6 journals, i.e. those already used in the training corpus plus a new one.

Since our matching system does not require supervised learning, we use the training set as a development set, to perform tests and tune the required parameters.

6.1 Training phase

We report in Table 1 the best parameter combination with the corresponding results obtained on the training set. For details about KX parameters, see the explanation in Section 3.

	Subtask 1 : Full articles	Subtask 2 : Short articles
KX Parameters		
N. of key-concepts extracted from each article	60	30
$MinCorpus$	8	8
$MinDoc$	2	2
Use $corpusIdf$	No	Yes
Multiply relevance by key-concept length	Yes	No
Consider position of first occurrence	Yes	No
Shorter concept subsumption	Yes (1 cycle)	Yes (1 cycle)
Longer concept boosting (assign frequency of shorter key-concepts to longer ones)	Yes (1 cycle) Yes	Yes (1 cycle) Yes
Similarity function for similarity score assignment	Sim_{F_1}	Sim_{F_1}
Graph matching algorithm	$LocMatch$	$LocMatch$
Journal-based match	No	No
P, R, F1 on training set	P 0.976, R 0.966, **F1 0.971**	P 0.943, R 0.940, **F1 0.941**

TABLE 1 – Best parameter combination for training set

As expected, the final results obtained using the full articles outperform those based on the articles without introduction and conclusions. This shows that the information at the beginning and at the end of an article is usually crucial to the identification of the most relevant concepts of the document.

Another difference is the number of top-ranked key-concepts considered for the match : in shorter articles 30 terms are enough, while in full ones 60 key-concepts seem to capture better the document content. Besides, in the classification of shorter articles, the integration of $corpusIdf$ information (computed over all training documents)

in the ranking process improves on the matching performance, while for full articles is does not provide additional useful information. As expected, in the second subtask the information about the position of the first occurrence of the key-concept does not help, because the introduction has been removed. Also, longer key-concepts are more relevant to the match of full articles than of shortened ones (see parameter *Multiply relevance by key-concept length*).

As for the similarity function, we tested also Sim_{Prec} only and Sim_{Rec} only, but the best results are achieved using their weighted average Sim_{F_1}.

We also experimented with both graph matching algorithms, but the $LocMatch$ always outperforms the *Hungarian Algorithm* (best F1 is 0.68 on full articles and 0.69 on shortened articles).

Finally, we tested two different experimental settings, one including all documents in a single corpus, and one subdividing them into smaller corpora according to the corresponding journal, and performing the abstract - article match only among documents coming from the same journal. For example, 60 articles and 60 abstracts in the training set have been extracted from the journal “Anthropologie et Sociétés", so we have evaluated also the performance of our system on this subclass of documents (see parameter *Journal-based match*). Surprisingly, this kind of approach does not improve on the system performance on the training set, because we achieved as best F1 0.965 on full articles.

6.2 Test phase

In the test phase, we submitted three system runs on the first subtask and three runs on the second one. For both tasks, one run was based on the parameter configuration in Table 1, a second run was computed by considering also the $corpusIdf$ (this time computed over all training and test documents), and a third one by including the $corpusIdf$ in a journal-based setting. Evaluation results are displayed in Table 2. Note that this time we report only one accuracy score obtained with the official task scorer, which corresponds to the percentage of correct matches.

	Subtask 1 : Full articles	Subtask 2 : Short articles
Setting as in Table 1	0.960	0.934
+ $corpusIdf$	0.975	**0.964**
+ journal-based match	**0.990**	**0.964**

TABLE 2 – System evaluation on test set

The performance on the test set outperforms the results obtained on the training set, probably because of the smaller amount of documents to match. The $corpusIdf$ is now relevant to the matching quality because it is computed on a larger corpus, comprising 996 documents (498 full articles and 498 abstracts), while in the training phase only 600 documents were available. This means that the idf information is important in the matching task only if it is computed on a large corpus. Also, the journal-based setting on the test set improves on the system performance with full articles because it significantly reduces the number of possible matches. However, the improvement is not achieved on shortened articles, probably because the content of the documents coming from the same journal tends to be more homogeneous and more difficult to discriminate if information-rich parts such as introduction and conclusions are removed.

7 Conclusions

In this paper we have presented a system for abstract - article match. It first extracts a list of key-concepts from each document, then a similarity score is applied between the key-concept lists and the abstracts, and finally a graph match algorithm is applied in order to identify the best matches.

The key-concept extraction component was created as a standalone system and was previously evaluated in the SemEval-2010 campaign on English texts with good results. For DEFT, we developed an extension of the system

which is able to handle French texts without performing any syntactic analysis. Besides, the system does not need a large traning corpus because only a small development set for parameter tuning is required.

The system performed well also in this case, since it was able to correctly match almost all pairs in the test set. In the future, we plan to extend the key-concept extraction component also to new languages, and we are currently working at the integration of the French morphological analyzer *Morfette* (Chrupala *et al.*, 2008) into the extraction pipeline.

Acknowledgements

This work has been partially funded by the European Commission under the contract number FP7-248594, PES-CaDO project [3]. We thank Elena Cabrio for the support with the French language.

Références

CHRUPALA G., DINU G. & VAN GENABITH J. (2008). Learning Morphology with Morfette. In *Proceedings of the 6th International Conference on Languages Resources and Evaluations (LREC 2008)*, Marrakech, Morocco.

FRANTZI K., ANANIADOU S. & MIMA H. (2000). Automatic recognition of multi-word terms : the C-value/NC-value. *Journal of Digital Libraries*, **3**(2), 115–130.

JONES S., LUNDY S. & PAYNTER G. (2002). Interactive Document Summarisation Using Automatically Extracted Keyphrases. In *Proceedings of the 35th Hawaii International Conference on System Sciences*, Hawaii.

KIM S. N., MEDELYAN O., KAN M.-Y. & BALDWIN T. (2010). SemEval-2010 Task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of SemEval 2010, Task 5 : Keyword extraction from Scientific Articles*, Uppsala, Sweden.

KUHN H. W. (1955). The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, **2**, 83–97.

PIANTA E. (2011). Content Distillation from Patent Material. Draft of a Book Chapter on the PatExpert system (to appear).

PIANTA E., GIRARDI C. & ZANOLI R. (2008). The TextPro tool suite. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC)*, Marrakech, Morocco.

PIANTA E. & TONELLI S. (2010). KX : A flexible system for Keyphrase eXtraction. In *Proceedings of SemEval 2010, Task 5 : Keyword extraction from Scientific Articles*, Uppsala, Sweden.

RICCA F., TONELLA P., GIRARDI C. & PIANTA E. (2004). An empirical study on keyword-based web site clustering. In *Proceedings of the 12th IWPC*, Bari, Italy.

3. http://www.pescado-project.eu/

Indexer, comparer, apparier des textes et leurs résumés : une exploration.

Martine Cadot(1, 2), Sylvain Aubin (3), Alain Lelu (2, 4, 5)

(1) Université de Nancy, Campus scientifique, BP 239, 54506 Vandoeuvre cedex – France
(2) LORIA, Campus scientifique, BP 239, 54506 Vandoeuvre cedex – France
(3) Diatopie SA, 27 Bd. St. Martin 75003 Paris
(4) LASELDI, 30 rue Mégevand – 25030 Besançon cedex
(5) ISCC, 20 r. Berbier-du-Metz 75013 Paris
martine.cadot@loria.fr, sylvain.aubin@diatopie.com
alain.lelu@univ-fcomte.fr

Résumé : Nous présentons ici la démarche qui nous a valu un score de 100% de réussite au défi DEFT 2011 dans la tâche d'appariement de résumés avec des articles dépourvus d'introduction et de conclusion : nous avons testé plusieurs types d'indexation et de distance résumé-texte, et mis au point une méthode d'appariement, en univers fermé, robuste et sans nécessité d'information extérieure. En combinant quatre variantes de la distance de compression, indépendante de la langue et du type de codage, elle permet d'atteindre 93% ; les 100% sont atteints avec la distance de Hellinger appliquée à des textes indexés par des noms lemmatisés et des termes composés, distance qui surpasse ici la classique TF-IDF. Nous suggérons son application en univers ouvert, avec plus de textes que de résumés, et des résumés sans texte.

Abstract: We develop here the approach which enabled our 100% success score in the DEFT 2011 challenge for the task of mating abstracts to their respective papers in the domain of social sciences and humanities. These papers were preliminarily reduced by their introduction and conclusion: we have tested several indexing methods and inter-text distance formulas between abstracts and papers. We then settled a mating method specific for the case when their relation is bijective. This method proved robust, and needs no external information source. Using the normalized compression method resulted in a 93% success score; using a crude lemmatizer/tagger, a key-phrase extractor and the Hellinger distance resulted in a 100% score. This distance proved to behave slightly better than the Salton's TF-IDF, and to be more fitted to incremental "open" corpuses processing, a more realistic mating problem for which we suggest a variant of our method.

Mots-clés : similarité textuelle, distance de compression, distance de Hellinger, TF-IDF, lemmatisation, extraction de termes composés, indexation, étiquetage morpho-syntaxique.

Keywords: text similarity, compression distance, Hellinger distance, TF-IDF, lemmatization, key-phrase mining, indexing, morpho-syntactic tagging.

1 Introduction : le problème à résoudre

Le défi DEFT 2011 comportait deux tâches à résoudre, un ensemble de textes anciens à dater d'une part, un ensemble de résumés d'articles de sciences humaines à apparier à leurs textes d'origine d'autre part. Nous avons choisi de nous concentrer sur cette dernière tâche, qui comportait deux « pistes », la plus difficile consistant à apparier les résumés avec des textes amputés de leur introduction et de leur conclusion, alors que l'autre dévoilait les textes complets. Ce défi a présenté pour nous l'intérêt de comparer quantitativement diverses solutions possibles à chaque étape de la chaîne des traitements, à savoir plusieurs méthodes d'indexation, diverses distances entre textes, et de mettre au point une procédure d'appariement, pour finalement évaluer les meilleures combinaisons de ces éléments. En effet, à l'instar de l'option choisie pour un autre défi DEFT (Lelu et al., 2006), nous avons pris le parti de multiplier les points de vues sur les données, ainsi que leurs critères de comparaison, en créant plusieurs chaînes de traitement les plus « orthogonales » possibles. De deux choses l'une : ou bien aucune de ces chaînes ne donnait zéro erreur sur l'ensemble d'apprentissage, et leur combinaison avait toute chance d'améliorer chacun de leurs résultats individuels, par exemple par une procédure de vote pondéré ; ou bien l'une d'elles ne donnait aucune erreur, et nous n'avions aucun élément pour douter de ses seuls résultats dans la phase de test. On verra que c'est cette dernière éventualité qui s'est produite.

Le défi définissait un univers de textes « fermé », au sens où à chaque résumé correspondait un texte, et vice-versa. Comme beaucoup de problèmes réels d'appariement de textes se situent en univers « ouvert », où il s'agit de trouver un ou des texte(s) les plus proche(s) d'un texte donné au sein d'une large collection, nous avons évoqué en conclusion les modifications à apporter à notre méthode d'appariement pour laisser de côté l'information « correspondance biunivoque entre résumés et textes ».

2 Les données

Les données proposées sont tirées de revues de sciences humaines en ligne dans la plateforme québécoise Erudit (*http://www.erudit.org/*). L'ensemble d'apprentissage rassemblait 300 textes des 5 revues *Anthropologie et Sociétés, Études internationales, Études littéraires, Meta, Revue des sciences de l'éducation* et les résumés correspondants. L'ensemble de test comportait en plus des textes et résumés de la revue *Philosophiques*. La difficulté de la tâche tenait en premier lieu pour la piste 1 à l'absence d'introduction et de conclusion, dont les éléments de synthèse sont souvent repris pour tout ou partie dans les résumés. Mais elle tenait surtout à la présence de certains résumés, parfois très succincts, pour de longs textes écrits par le même auteur, sur le même sujet, dans la même revue ! Seuls des spécialistes en traductologie, ou en histoire de la philosophie, par exemple, pouvaient être capables de les apparier correctement au fil d'une lecture attentive.

Il y avait quelques difficultés ponctuelles en plus : un résumé en grec dans l'ensemble d'apprentissage – nous nous sommes contentés d'une « Google-traduction » pour l'intégrer - , un texte vide dans l'ensemble de test – la difficulté était facilement résolue compte tenu de la bi-univocité des appariements.

3 Indexations et distances entre textes

Nous avons opté pour trois définitions contrastées de distances entre textes : deux font appel à un pré-traitement d'indexation, soit basique, soit élaboré, l'autre se veut la plus brute possible et indépendante de la langue. Pour cette dernière l'utilisation de N-grammes de caractères était un bon candidat, d'autant plus que nous l'avions déjà employée avec succès (Delprat et al., 2011). Mais nous avons voulu tester les possibilités d'une autre méthode, encore plus « aveugle », qui ne demande pas même le minimum de pré-traitements nécessaires pour les N-grammes : la distance de compression (Cilibrasi, 2003), indépendante de la langue et du codage textuel utilisé, qui ne nécessite pas même d'ouvrir chaque fichier texte. Nous décrivons tout d'abord cette dernière.

3.1 Distance de compression

La distance de compression normalisée entre deux chaînes de caractères x et y est définie comme suit par les auteurs cités :

$$D_c(x,y) = [Z(xy) – \min(Z(x), Z(y))] / \max(Z(x), Z(y))$$

où Z(x) est la longueur du texte x après compression sans perte, et xy désigne la concaténation des textes x et y. Nous avons utilisé ici le compresseur zip.exe du groupe Info-Zip (version 2.32, en ligne de commande) implantant l'algorithme de déflation (cf. *http://www.info-zip.org/*). A noter que du fait du caractère séquentiel et non-optimal de cet algorithme de compression sans perte, D(x,y) diffère en général de D(y,x) quand les longueurs des deux textes diffèrent fortement, ce qui est le cas ici. Nous avons utilisé les deux configurations « résumé puis texte » et « texte puis résumé » sur les ensembles d'apprentissage et de test.

Compte tenu du déséquilibre entre les tailles des résumés (notés r) et des textes t la distance ci-dessus revient, dans la quasi-totalité des cas, à :

$$D_{c1}(r,t) = [Z(rt) –Z(r)] / Z(t).$$

Dans le cadre de notre volonté de multiplier les points de vues, ne serait-ce que dans le cas d'une seule méthode, nous avons également utilisé la dissimilarité

$$D_{c2}(r,t) = [Z(rt) –Z(t)] / Z(r),$$

normalisée par rapport au seul résumé, a priori plus « sensible » et sujette à de fortes variations entre 0 et 1.

L'avantage de ce type de distance est qu'il ne nécessite absolument aucun pré-traitement et est indépendant de la langue et du type de codage des caractères. L'inconvénient de cet avantage est qu'il est une boîte noire qui ne montre pas le pourquoi de ses résultats, contrairement aux distances vectorielles qui peuvent, si on le désire, expliciter les descripteurs communs à deux textes. Son inconvénient pratique est qu'il nécessite |R|*|T| opérations de concaténation de fichiers, où |R| et |T| sont respectivement les nombres de résumés et de textes, soit ici 90 000 opérations pour l'ensemble d'apprentissage.

3.2 Indexation basique par formes brutes

Nous avons choisi, comme indexation de référence à titre d'intermédiaire entre l'absence d'indexation décrite ci-dessus et l'indexation évoluée de type morphosyntaxique, les chaînes de caractères séparées par des espaces (ordinaires ou insécables) ou un séparateur parmi les caractères de la liste : *.,;:?!'-«»()[]{}*. Nous appellerons « formes brutes » ces chaînes de caractères.

L'avantage de cette approche est d'être la plus simple pour obtenir des vecteurs-documents. Elle n'est opératoire que pour les langues à caractères alphabétiques et non agglutinantes, dans lesquelles il est trivial de séparer les formes. Pour les langues agglutinantes ou sans séparateurs triviaux de mots, la technique des N-grammes de caractères s'impose, ce qui n'est pas le cas ici. Nous avons obtenu un dictionnaire de 26 266 formes brutes, hors hapax, pour l'ensemble d'apprentissage des textes tronqués, de 21 270 pour l'ensemble de test correspondant.

3.3 Indexation élaborée par lemmes et expressions composées

Nous avons utilisé à cet effet le logiciel NeuroNav (commercialisé par Diatopie SA, et décrit dans *http://webu2.upmf-grenoble.fr/adest/seminaires/lelu02/ADEST2001_SA_AL.htm*) qui comporte, outre une interface de consultation d'une base textuelle et navigation dans les « axes obliques » synthétiques qui en sont extraits, deux modules utilisés pour la présente étude :

- Un module d'étiquetage grammatical et lemmatisation, pour le français et l'anglais, qui étiquette de façon minimale en trois catégories : les adjectifs, les verbes et les substantifs. Une quatrième catégorie, celle des particules syntaxiques, est éliminée dans la phase d'indexation. Chaque forme reconnue dans un dictionnaire de formes donne lieu à un lemme étiqueté, selon une heuristique basique, mais globalement efficace : le lemme et l'étiquette sont ceux dont l'emploi est le plus fréquent dans la langue générale. Les formes absentes du dictionnaire sont étiquetées substantif, par défaut.

- Un module d'extraction d'expressions composées (jusqu'à 6 composants), à partir de patrons syntaxiques, comme *Subst* de *Subst*, *Subst Adj*, *Subst* en *Subst*, etc. Des listes d'exceptions permettent de filtrer des expressions de la rhétorique courante, comme *nombreuses façons* en français. Pour l'anglais NeuroNav est orienté davantage vers l'analyse de bases documentaires ou journalistiques que de textes littéraires, et des expressions telles que *excellent results* ou *given context* sont éliminées. Le but est de privilégier la précision sur le rappel, pour ne pas troubler l'utilisateur par la présence de nombreux termes d'indexation non pertinents. Une étude sur une petite base de résumés dans le domaine génomique indexés à la main a montré une précision supérieure à 95%, pour un rappel de termes composés corrects de l'ordre de 50%. Ce taux de rappel médiocre, dû aussi pour beaucoup au vocabulaire très particulier du domaine, est rendu acceptable par la redondance présente dans tout texte, et dans notre cas aucun des 300 résumés, puis des 200 de la phase de test, aussi court soit-il, ne s'est vu attribuer aucun terme, malgré l'élimination des mots et expressions hapax, inutiles pour les comparaisons résumés/textes à effectuer.

Au final, l'indexation des résumés et textes tronqués de l'ensemble d'apprentissage s'est traduite par un total de 23 331 termes lemmatisés différents, hors hapax, dont 3771 adjectifs, 2062 verbes, 10 412 substantifs et 7286 expressions composées. L'ensemble de test a fourni de son côté 16 365 termes.

3.4 Distances entre textes indexés

Un ensemble de textes indexés peut être représenté par un ensemble de vecteurs dont les composantes sont le nombre d'occurrences du descripteur i dans le texte t (modèle « sac de descripteurs »). L'indicateur de similarité le plus utilisé pour les textes (Banerjee et al., 2005) est le cosinus, obtenu par simple produit scalaire entre deux vecteurs-textes normalisés $<\mathbf{v}_t, \mathbf{v}_{t'}>$, qui permet de comparer des textes de longueurs différentes. La façon de normaliser, ainsi que la pondération éventuelles des composantes, définissent une large palette de possibilités, dont nous retiendrons deux parmi les plus utilisées, et une moins connue dont nous avons déjà montré les mérites (Lelu, 2003). Pour chacun de ces cosinus, nous définissons de façon homogène la distance entre deux documents comme l'arc de leur cosinus, ce qui évite certaines variantes qui peuvent avoir une influence sur les résultats en univers fermé : en effet certains auteurs définissent la « distance du cosinus » comme 1-cos, alors que plus rigoureusement la distance sur l'hypersphère est définie par la fonction ArcCosinus, et la distance de la corde, qui lui est équivalente pour toutes opérations de tri et seuillage, s'écrit $\sqrt{2} *(1-\cos)^{1/2}$ Nous définissons ci-après trois transformations du vecteur-document brut produisant des vecteurs normalisés. La distance correspondante entre deux textes sera définie dans tous les cas comme l'arc de leur cosinus, cosinus obtenu par produit scalaire entre ces deux vecteurs-textes normalisés.

- **Distance euclidienne entre vecteurs-textes normalisés**

Chaque composante k_{it} (occurrence du mot i dans le texte t) du vecteur-texte $\mathbf{v}_t$ est divisée par la norme euclidienne de ce vecteur :

$$\{ k_{it} \} \rightarrow \{ k_{it} / \| \mathbf{v}_t \| \} \text{ où } \| \mathbf{v}_t \| = (\Sigma_i k_{it}^2)^{1/2}$$

Dans ce cas le vecteur de longueur 1 obtenu est colinéaire au vecteur d'origine.

- **Distance « TF-IDF »**

Dans cette distance, très utilisée dans les applications de recherche de l'information, les occurrences du mot i (*Term Frequency*) sont pondérées de façon à diminuer l'importance des termes fréquents (*Inverse Document Frequency*) :

$$\{ k_{it} \} \rightarrow \{ k_{it} \log(N/ n_{i.}) \}$$

où $n_{i.}$ est le nombre de présences (et non d'occurrences) du mot i dans l'ensemble des unités textuelles, et N le nombre de textes. Les composantes de ce nouveau vecteur sont alors divisées par sa norme. Le vecteur de longueur 1 obtenu n'est plus colinéaire au vecteur d'origine.

A noter que cette distance est mal adaptée à l'arrivée incrémentale de nouveaux textes, sauf à supposer une certaine stabilité dans la distribution globale des présences de termes $n_{i.}/N$, ces derniers pouvant modifier le système des distances entre textes utilisé auparavant.

- **Distance de Hellinger**

Cette distance semble très proche de la distance euclidienne sur la sphère, voire redondante avec elle, puisque chaque vecteur-texte est normalisé sans intervention des fréquences ou présences globales de descripteurs $k_{i.}$ ou $n_{i.}$. Chaque vecteur-texte est normalisé comme suit :

$$\{ k_{it} \} \rightarrow \{ (k_{it} / k_{.t})^{1/2} \}$$

Elle est également adaptée au cas de l'analyse d'un corpus de façon incrémentale, au fil de l'arrivée de nouveaux documents indexés, qui ne changent pas les valeurs des composantes des vecteurs précédents. A noter que chaque vecteur de longueur 1 obtenu $\mathbf{v}_t$ n'est plus colinéaire à son vecteur d'origine.

La distance de Hellinger D_H est la longueur de la corde correspondant à l'angle $(\mathbf{v}_t, \mathbf{v}_{t'})$ - égale au plus à 2 quand ces deux vecteurs normalisés sont opposés, égale à $\sqrt{2}$ quand ils sont orthogonaux.

$$D_H(t,t') = \sqrt{2} *(1-<\mathbf{v}_t, \mathbf{v}_{t'}>)^{1/2}$$

Plusieurs propriétés théoriques la rendent intéressante de notre point de vue :

- Elle est particulièrement adaptée aux « données directionnelles » (Banerjee et al., 2005) que sont les données textuelles, pour lesquelles seuls sont pertinents les angles entre vecteurs.

- Elle est liée à la mesure du gain d'information de Renyi d'ordre ½ (Renyi, 1966) apporté par une distribution $\mathbf{x}_q$ quand on connaît la distribution $\mathbf{x}_p$:

$$I^{(1/2)}(\mathbf{x}_q / \mathbf{x}_p) = - 2 \log_2 (\cos(\mathbf{z}_p, \mathbf{z}_q)) = - 2 \log_2 (1 - D_H^2/2)$$

- et surtout (Escofier, 1978) et (Domengès et Volle, 1979) ont montré qu'elle satisfaisait à la même propriété d'équivalence distributionnelle que la distance du khi-deux utilisée en Analyse Factorielle des Correspondances : si on fusionne deux descripteurs de mêmes profils relatifs, les distances entre les unités textuelles sont inchangées. En d'autres termes, dans le cas où les descripteurs sont des mots et les unités décrites des textes, cette propriété assure la stabilité du système des distances entre textes au regard de l'éclatement ou du regroupement de mots de distributions proches. Ceci peut expliquer la considérable supériorité de ses performances qu'on constatera plus bas par rapport à la distance euclidienne sur la sphère, et qui confirme des constats faits par ailleurs – ex. : (Legendre et Gallagher, 2003).

4 Notre méthode d'appariement

4.1 La première étape de notre méthode d'appariement

Nous décidons d'apparier résumés et textes à l'aide d'une méthode s'inspirant des voisins réciproques dont le principe est : on affecte à un résumé *r1* le texte *t* le plus proche, puis on affecte au texte t le résumé *r2* le plus proche, et si le résumé r1 se trouve être le résumé r2, on estime avoir réussi l'appariement entre le résumé et le texte. Nous allons illustrer ce principe basique sur les distances obtenues par compression entre les résumés et les textes, en utilisant la variante : distance 2, configuration « résumé puis texte ». Pour cela nous avons créé une première liste de couples (résumé, texte) candidats à l'appariement en cherchant pour chaque résumé le texte le plus proche, et une deuxième liste de couples en cherchant pour chaque texte le résumé le plus proche, afin de les confronter pour la recherche des voisins réciproques. Mais nous constatons un grand nombre de « doublons ». En effet, parmi les 300 couples de la deuxième liste, le même résumé (r90) était affecté à 295 des 300 textes alors qu'il aurait dû n'être affecté qu'à un seul texte, le résumé r182 était affecté à 3 textes, et les 2 derniers textes de la liste produisaient les 2 couples (r103, t235) et (r145, t234), associations qui se sont avérées justes. La première liste était plus intéressante, car seulement 22 des 300 textes se trouvaient être les plus proches de plusieurs résumés (2 ou 3), ce qui laissait 252 associations possibles entre un résumé et un texte. Parmi celles-ci, une seule était fausse. En combinant les deux listes pour obtenir les voisins réciproques, nous n'avions plus qu'un couple. Devant ce résultat, nous avons décidé de garder les voisins réciproques, qui ont formé les appariements de qualité 1, mais aussi les couples de « qualité 2 », c'est-à-dire présents dans une des deux listes seulement *à condition qu'ils ne vienne pas en contradiction avec ceux de l'autre liste*, en définissant les couples en contradiction des couples qui ont un élément commun, comme (r, t') et (r, t''), ou (r', t) et (r'', t). Ainsi les deux listes qui avaient l'une deux couples et l'autre 252 couples ont été combinées en une liste de 253 couples : 1 couple de qualité 1 (voisins réciproques), et 252 de qualité 2. Sur les 300 couples attendus, on en a trouvé 252 justes et 1 faux, donc un taux de reconnaissance correcte de 84%. On peut voir dans le *Tableau 1* le résultat de cette première étape pour les 4 variantes de la méthode.

Variante	Distance	Ordre	Taux de reconnaissance (bien reconnus/tous)	Nombre d'erreurs
1	1	résumé, texte	84,00%	1
2	1	texte, résumé	55,33%	0
3	2	résumé, texte	41,67%	0
4	2	texte, résumé	37,00%	1

Tableau 1 : Résultats sur l'ensemble d'apprentissage

Les résultats pour les 3 autres variantes de la méthode de compression sont moins bons, mais les 4 variantes ont en commun leur très petit nombre d'erreurs (0 ou 1), ce qui va permettre d'augmenter leur performance par combinaison, comme indiqué dans le Tableau 2. On a réparti dans ce tableau les 300 associations attendues, correspondant donc aux 300 résumés, en indiquant en colonne le résultat final (-1 : appariement faux, 0 : non apparié, 1 : appariement juste) et en ligne la qualité (-1 : appariements proposés en contradiction, 1 : appariement avec accord des 4 variantes, 2 : accord de 3 variantes, 3 : accord de 2 variantes, 4 : 1 variante , et 5 : aucune variante donc pas d'appariement proposé).

On a indiqué par une étoile dans le tableau le nombre d'appariements proposés par plusieurs variantes qui ont produit une contradiction. Il y en a un seul. Il est situé sur la ligne de qualité -1 et dans la colonne de validation 0 car il n'a pas abouti à une association. Il s'agit du résumé 67, qui a été associé au texte 38 dans les deux premières variantes, à aucun dans la troisième, et au texte 160 dans la quatrième. On a choisi de ne pas lui affecter de texte, du fait de la contradiction, mais si on avait choisi le texte par un vote à la majorité, on aurait eu un appariement juste. Sur les 300 appariements possibles, on en a proposé 279 et 278 se sont avérés justes, soit un taux de reconnaissance de 92,67%, avec une seule erreur. Le seul appariement faux est entre le résumé r122 et le texte t51 au lieu du texte attendu t107. Il n'a été proposé que dans une des variantes, comme l'indique la valeur 4 de sa qualité. Le fait qu'il n'y ait aucune erreur

dans les associations de qualité 1, 2 et 3 tend à augmenter la confiance qu'on peut avoir dans l'indice de qualité de la comparaison.

Nous venons de voir que la méthode des voisins réciproques a été modifiée pour donner une méthode de prédiction honorable des couples (résumé, texte) attendus, et qu'elle peut être améliorée par combinaison. Mais nous avons laissé de côté des doublons qui peuvent être en partie récupérés. C'est la deuxième étape de notre algorithme d'appariement.

Nombre d'associations	Validation			
qualité	-1	0	1	Total
-1		1*		1
1			46	46
2			68	68
3			100	100
4	1		64	65
5		20		20
Total	1	21	278	300

Tableau 2 : Combinaison des résultats des 4 variantes sur l'ensemble d'apprentissage

4.2 La deuxième étape de notre méthode d'appariement

Dans l'étape 1 de l'algorithme, nous avons laissé de côté le résumé r90 car il était le plus proche de 295 textes, mais il y a certainement parmi ceux-ci le bon texte à lui apparier, appelons-le t,. Quand nous avons cherché les résumés les plus proches du texte t, le premier était le résumé r90, à la distance d1, et le second, appelons-le r, était à la distance d2 > d1, par construction[1]. Nous supposons que la différence d2-d1 est plus grande pour le texte t que pour tous les 294 textes associés à r90, ce saut important entre les deux résumés indiquant que c'est le résumé r90 qui lui est associé, et non le résumé r. Selon ce principe, nous reprenons tous les doublons et nous créons deux nouvelles listes de couples. Dans notre exemple, la première liste contiendra les couples associés aux résumés r90 et r182, et l'autre les 22 couples correspondant aux 22 textes à l'origine de doublons. Et ces deux listes sont fusionnées de la manière habituelle. Dans notre exemple, un seul des deux couples de la première liste appartient à la seconde, ce qui donne un couple de qualité 3, l'autre disparaît car il est en contradiction avec un de ceux de la liste, et il en reste 20 dans l'autre liste qui sont de qualité 4.

S'il reste encore des textes et résumés qui n'ont pas été appariés, la troisième étape permet d'essayer de les apparier.

4.3 La troisième étape de notre méthode d'appariement

On prend tous les résumés qui n'ont pas été appariés, et on cherche pour chacun le texte correspondant au saut maximum de distance, mais en se limitant aux k plus petites distances (les meilleurs résultats ont été obtenus avec k=10). On créée ainsi une première liste de couples. Puis on procède de la même façon avec chaque texte non apparié pour créer la deuxième liste de couples, et on fusionne les deux listes de la

[1] On a supposé, pour réaliser cet algorithme, que les distances entre un texte et les résumés n'étaient jamais égales, pas plus que celles entre un résumé et les textes. Mais le fonctionnement de l'algorithme n'est pas gêné par des éventuelles égalités.

manière habituelle. On contrôle ensuite la cohérence de cette nouvelle liste avec les précédentes, et on ne garde que les nouveaux couples qui ne contredisent pas d'anciens couples. Dans notre exemple, cela n'a produit que 2 nouveaux textes auxquels on a attribué la qualité 4.

5 Résultats dans le cadre du défi (en univers fermé)

On a fait figurer dans le Tableau 3 les résultats obtenus sur les 300 résumés et textes de l'ensemble d'apprentissage en appliquant notre algorithme d'appariement sur les 4 méthodes et leurs variantes, séparément.

Méthode	Paramètres	Distance	Étape 1				Étapes 2 et 3				Total	
			Vq1	Fq1	Vq2	Fq2	Vq3	Fq3	Vq4	Fq4	Vrai	Faux
Compression	res_txt	Dist2	1	0	251	1	1	0	22	0	275	1
Compression	res_txt	Dist1	1	0	124	0	4	0	30	6	159	6
Compression	txt_res	Dist2	1	0	165	0	0	0	52	4	218	4
Compression	txt_res	Dist1	2	0	109	1	3	0	29	1	143	2
Formes		Hellinger	184	0	72	0	28	0	13	0	297	0
Formes		**TF-IDF**	**265**	**0**	**9**	**0**	**14**	**0**	**8**	**0**	**296**	**0**
Formes		Euclidienne	18	0	125	16	11	0	40	18	194	34
Lemmes	**cat 127**	**Hellinger**	**269**	**0**	**5**	**0**	**17**	**0**	**9**	**0**	**300**	**0**
Lemmes	cat 127	TF-IDF	253	0	13	0	21	0	10	0	297	0
Lemmes	cat 127	Euclidienne	220	0	28	1	37	0	10	0	295	1
Lemmes	cat 27	Hellinger	263	0	11	0	18	0	4	0	296	0
Lemmes	cat 27	TF-IDF	252	0	14	0	19	2	7	1	292	3
Lemmes	cat 27	Euclidienne	216	0	27	0	39	1	12	1	294	2
Lemmes	**cat 12347**	**Hellinger**	**265**	**0**	**10**	**0**	**16**	**0**	**7**	**0**	**298**	**0**
Lemmes	cat 12347	TF-IDF	259	0	11	0	19	0	7	0	296	0
Lemmes	cat 12347	Euclidienne	209	0	33	0	40	0	11	2	293	2
Lemmes	cat 2347	Hellinger	257	0	15	0	19	0	7	0	298	0
Lemmes	cat 2347	TF-IDF	259	0	12	0	18	1	5	1	294	2
Lemmes	cat 2347	Euclidienne	210	0	32	0	39	0	12	3	293	3

Tableau 3: Résultats sur l'ensemble d'apprentissage

Il y a une ligne par variante, d'abord les 4 variantes de la méthode de compression, puis la méthode de comparaison utilisant les formes avec les 3 distances, et enfin la méthode utilisant les lemmes avec divers ensembles de catégories de lemmes (1 : mots composés, 2 : noms en dehors du dictionnaire, 3 : adjectifs,

4 : verbes, 7 : noms du dictionnaire). En colonnes on a le nombre de couples justes et faux appariés à chaque étape de l'algorithme, les couples de qualité 1 et 2 de la première étape (Vq1 pour les couples justes de qualité 1, Fq1 pour les couples faux de qualité 1, etc.) , ceux de qualité 3 de la deuxième étape, de qualité 4 de la troisième et le nombre total de couples appariés. On peut voir que tous les couples appariés de qualité 1 (ce sont les voisins réciproques) sont justes, quelle que soit la méthode considérée. On souhaite combiner les méthodes produisant le moins d'erreurs, ce sont donc les méthodes produisant le plus grand nombre de couples de qualité 1. On peut voir en gras les trois meilleures méthodes selon ce critère, et parmi celles-ci, la méthode NeuroNav d'indexation par noms et mots composés, combinée à la distance d'Hellinger, qui donne exactement 300 associations entre résumés et textes, toutes justes, donc 100% de réussite.

Méthode	Paramètres	Distance	Étape 1				Étapes 2 et 3				Total	
			Vq1	Fq1	Vq2	Fq2	Vq3	Fq3	Vq4	Fq4	Vrai	Faux
Compression	res_txt	Dist2	0	0	29	2	0	0	128	6	157	8
Compression	res_txt	Dist1	0	0	85	0	0	1	24	4	109	5
Compression	txt_res	Dist2	0	0	49	1	0	0	84	0	133	1
Compression	txt_res	Dist1	0	0	82	2	0	1	22	3	104	6
Formes		Hellinger	87	0	83	2	15	0	6	1	191	3
Formes		**TF-IDF**	**171**	**0**	**9**	**0**	**10**	**0**	**8**	**0**	**198**	**0**
Formes		Euclidienne	1	0	32	23	4	0	20	22	57	45
Lemmes	**cat 127**	**Hellinger**	**180**	**0**	**3**	**0**	**9**	**0**	**4**	**0**	**196**	**0**
Lemmes	cat 127	TF-IDF	170	0	8	0	15	0	5	0	198	0
Lemmes	cat 127	Euclidienne	161	0	12	1	14	0	9	1	196	2
Lemmes	cat 27	Hellinger	182	0	2	0	9	0	5	0	198	0
Lemmes	cat 27	TF-IDF	165	0	11	0	17	0	5	0	198	0
Lemmes	cat 27	Euclidienne	161	0	13	1	15	0	5	1	194	2
Lemmes	**cat 12347**	**Hellinger**	**178**	**0**	**6**	**0**	**9**	**0**	**4**	**0**	**197**	**0**
Lemmes	cat 12347	TF-IDF	177	0	5	0	10	0	6	0	198	0
Lemmes	cat 12347	Euclidienne	153	0	21	0	19	0	4	0	197	0
Lemmes	cat 2347	Hellinger	178	0	8	0	8	0	3	0	197	0
Lemmes	cat 2347	TF-IDF	178	0	4	0	9	0	7	0	198	0
Lemmes	cat 2347	Euclidienne	152	0	22	0	19	0	4	0	197	0

Tableau 4 : Résultats sur l'ensemble de test

Dans le Tableau 4, on a fait figurer les mêmes résultats pour l'ensemble Test. On ne connaissait pas le résultat attendu, mais les trois méthodes choisies au moment de l'apprentissage ont donné respectivement 198, 197 et 196 couples sur les 198 attendus. La combinaison des trois méthodes a consisté à contrôler que

les 196 couples communs étaient les mêmes, que le 197ème d'une des deux méthodes coïncidait bien avec l'un des deux restants de l'autre méthode, et donc il ne restait plus de doute pour le dernier.

6 Conclusions, perspectives

Ayant obtenu 0 erreur pour la piste 2, la plus difficile, il n'était pas étonnant que l'on obtienne le même niveau de performance pour la piste 1 (appariement résumés-textes complets), ce qui nous a amené à la première place ex-aequo du classement. En dehors de cette satisfaction, plusieurs conclusions peuvent être tirées :

- Pour la tâche d'appariement avec des textes amputés de l'introduction et de la conclusion, il n'était pas évident au départ que l'ordinateur pourrait faire au moins aussi bien, et bien plus rapidement, qu'un lecteur intelligent et spécialiste d'un domaine difficile des sciences humaines. C'est pourtant ce qui s'est produit.

- Nous avons mis au point une méthode d'appariement robuste qui serait utilisable sans grandes modifications en univers ouvert (textes plus nombreux que les résumés, certains résumés sans textes, …) en se contentant de retirer les phases de calculs de distances partant des textes vers les résumés. Vers un prochain défi ?

- Les choix qui ont sous-tendu la conception du logiciel NeuroNav sont confirmés : pour les calculs de distance entre textes, l'utilisation des seuls lemmes de substantifs et termes composés ; pour le type de distance, celle de Hellinger plutôt que celle du TF-IDF, aux performances un peu inférieures et peu compatible avec un univers ouvert, ou que la distance euclidienne sur la sphère, bien moins performante. Une augmentation de la qualité de l'analyse morphosyntaxique, quelque peu rudimentaire aujourd'hui, laisserait espérer des performances voisines de 100% en univers ouvert

- De façon surprenante, la distance de compression donne 93% de réussite sur l'ensemble d'apprentissage, 88,4% sur l'ensemble de test, avec 6 erreurs dont une due au texte vide, performances qui sont loin d'être négligeables pour une méthode aveugle, multilingue et multi-codages, sans aucune intelligence ajoutée ! Des méthodes de compression plus efficaces que celle, ancienne, de déflation pourraient encore augmenter ces scores et rivaliser avec les méthodes morphosyntaxiques « intelligentes ».

Références

BANERJEE A., DHILLON. I., GHOSH J.AND SRA S. (2005). Clustering on the Unit Hypersphere using Von Mises-Fisher Distributions. *Journal of Machine Learning Research* (JMLR), vol. 6, 1345-1382.

CILIBRASI R. (2003). Clustering by compression. IEEE *Trans. on Information Theory*, 51(4) : 1523–1545.

DELPRAT B., HALLAB M., CADOT M., LELU A. (2011). Processing a Mayan Corpus for Enhancing our Knowledge of Ancient Scripts, *4th. International Conference on Information Systems and Economic Intelligence* (SIIE 2011) February 17, 18 and 19th., 2011, Marrakech (Maroc).

DOMENGÈS D. ET VOLLE M. (1979). Analyse factorielle sphérique : une exploration. *Annales de l'INSEE*, N° 35, P. 3-83.

ESCOFIER B. (1978). Analyses factorielles et distances répondant au principe d'équivalence distributionnelle. *Revue de Stat. Appliquée*, 26(4):29-37, Paris.

LEGENDRE P., GALLAGHER E. (2001). Ecologically meaningful transformations for ordination of species data. *Oecologia*: 129: 271-280.

LELU A. (2003). Evaluation de trois mesures de similarité utilisées en sciences de l'information. *Information Sciences for Decision Making* 6 14–25.

LELU A., CADOT M., AUBIN S. (2006). Coopération multiniveau d'approches non supervisées et supervisées pour la détection des ruptures thématiques dans les discours présidentiels français. *Semaine du Document Numérique 2006 / DEFT'06*, 21-22/9/2006, Fribourg, Suisse.
http://www.lri.fr/~aze/fdt/DEFT06/articles/DEFT06-Lelu-Cadot-Aubin.pdf

RENYI A. (1966). *Calcul des probabilités*, Paris, Dunod, 620 p.

Matching Texts with SUMMA

Horacio Saggion
TALN
Department of Information and Communication Technologies
Universitat Pompeu Fabra
C/Tanger 122
Barcelona - 08018
Spain
horacio.saggion@upf.edu

Résumé. On décrit notre approche au problème de l'appariement de résumés/articles scientifiques proposé par le programme DÉfi Fouille de Textes (DEFT). Nous avons développé un algorithme d'appariement de textes qui utilise des ressources quasiment indépendantes de la langue. L'algorithme crée des representations de documents tout en utilisant le système SUMMA et les compare grâce à une mesure de similarité cosinus qui nous permet de sélectionner le meilleure candidat pour former la paire. Nos résultats indiquent que cette approche est très précise et qu'elle pourrait s'appliquer à d'autres langues.

Abstract. We describe our solution to the abstract-document matching problem proposed in the DÉfi Fouille de Textes (DEFT) evaluation programme. We have developed a text matching algorithm using quasi language independent resources. Our algorithm creates document representations using the SUMMA system and compares representations using a cosine similarity measure selecting the best matching candidate. Results indicate that the solution is highly accurate and could be applied to other languages.

Mots-clés : Système SUMMA, résumé automatique, similarité textuelle.

Keywords: SUMMA System, Text Summarization, Text Similarity.

1 Introduction

The DÉfi Fouille de Textes (DEFT) evaluation programme focuses on natural language processing technologies for text mining problem solving in the French language. Different challenges have been put forward in previous editions of DEFT such as that of opinion mining (Grouin *et al.*, 2009) or text classification (Grouin *et al.*, 2008).

The DEFT 2011 evaluation chapter proposed two different text mining tasks for participating teams : (i) identify the publication year of a given French article and (ii) identify the source document (out of a pool of documents) for a given text abstracts ("abstract document matching" problem).

In our first participation in DEFT, we concentrated on the abstract-document matching problem only. The data set for the abstract-document matching problem is a set of scientific articles and their abstracts. These were published in reviews in the field of humanities. The corpus was transformed into the following three components :

– *ART* : the set of articles with their author abstracts removed ;
– *RES* : the set of author's abstracts ;
– *TXT* : same as ART but where introduction and conclusion sections have been removed from the articles.

Information about article's authors and titles was removed. Examples of matching documents are shown in Figures 1 (abstract), 2 (article), and 3 (article w/o introduction/conclusion).

Two subtasks were proposed for the abstract-document matching problem : (i) identify for each abstract in *RES* the article in *ART* where the abstract comes from, and (ii) identify for each abstract in *RES* the text in *TXT* where

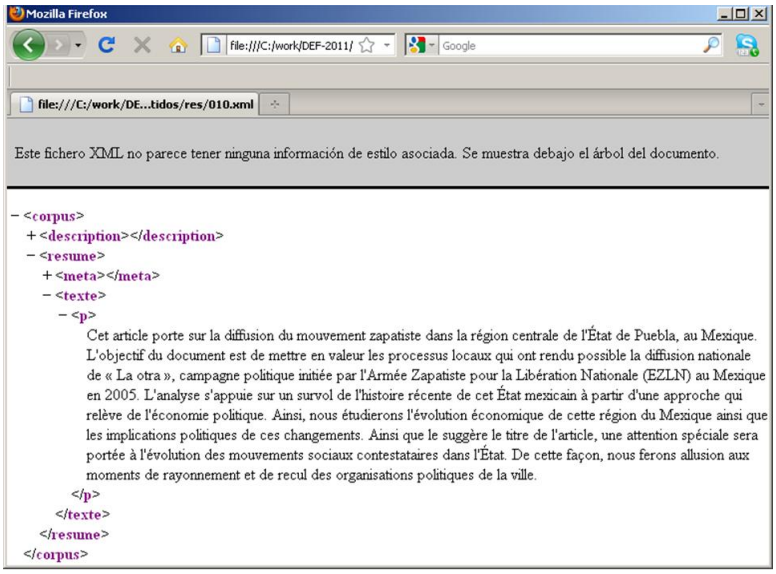


FIG. 1 – Abstract used during training

the abstract comes from. This problem is related to the abstract-descriptors matching problem we have proposed for manual evaluation of automatic text summarization systems (Saggion & Lapalme, 2002).

In DEFT, in addition to providing a single answer per abstract, there was the possibility of providing a set of matching documents, each with an associated confidence score where the confidence scores for a given abstracts have to add up to 1 (i.e., a probability distribution).

We developed our "abstract to text" matching solution in a very short period of time taking advantage of our SUMMA toolkit (Saggion, 2008a) which can be used to compare different document representations and which has been used successfully in previous evaluation campaign such as multi-document summarization (Saggion & Gaizauskas, 2004) or text clustering (Saggion, 2008a) which require document representations to be compared.

In the rest of this short communication we first describe how we have developed our system using available components and how we propose a solution (Section 2), we then describe the evaluation framework and results we have obtained (Section 3), and finally close the paper with a discussion and outlook (Section 4).

2 Document Processing

For basic document processing we have used functionalities available in the SUMMA toolkit (Saggion, 2008b) and GATE system (Maynard *et al.*, 2002).

2.1 SUMMA Functionalities in DEFT

SUMMA operates on GATE document representations and computes feature values for different text units. For example, for text summarization it computes different features for sentences to measure sentence relevance (i.e., sentence position, sentence-title similarity, sentence-document centroid similarity, etc.) ; these are implemented components (e.g., processing resources) which can be integrated in stand-alone applications. Features and annotations computed in SUMMA are added to the document representations and in this way passed from one module to the next one. The main SUMMA components used in DEFT were : (i) a module to create corpus statistics, (ii) a

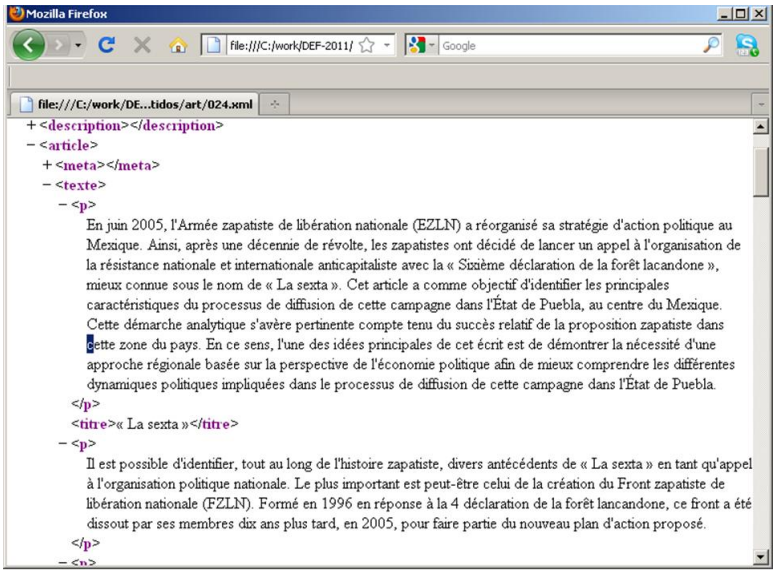


FIG. 2 – Article used during training

module to compute word document statistics, (iii) a flexible module to compute document vectors, and (iv) a module to compute document-document similarity measures. The DEFT algorithm is very simple and uses SUMMA as a Java library.

2.2 Processing Steps

Each document collection (ART, RES, TXT) was processed as illustrated in Figure 4 :

- First the collection of documents is transformed into XML format (for DEFT this was a simple document renaming procedure necessary for subsequent processes) and saved to data stores for more efficient processing since GATE needs considerable amount of memory for the documents ;
- Second, a basic document processing step takes place to identify different types of words in the document. In order to identify words in the documents we used a default tokenizer available in the GATE system. We have tried to use the French language tools from GATE (e.g. named recognition, etc.), but we finally gave up because these are very limited, containing mostly English resources which are of little help in the analysis of French language ;
- Third, an inverted document frequency table is created based on the set of documents to be processed (i.e., there is a table computed for each document collection). The table is created with functionalities available in the SUMMA summarization toolkit. Given a corpus of tokenized documents, an inverted document frequency table is created and stored to disk. The value of inverted document frequency for term t is $idf(t) = log(N+1/M_t+1)$ where M_t is the number of documents containing t and N is the number of documents in the collection ;
- Finally, a vector representation for each document in the collection is created. We also use SUMMA which implements the vector space model for this purpose (Salton, 1988). The tool computes token statistics including term frequency – the number of times each term occurs in the document (tf). Each vector contains for each term occurring in the text fragment, the value tf(t)*idf(t) (term frequency * inverted document frequency for term t). The vector creation component in SUMMA is flexible enough to allow specification of types of tokens to exclude, or a list of stop words to ignore, etc. For the experiments reported here we have not included punctuations, symbols, or numerical tokens in the vector representations. Figure 5 presents a document analysed with the tools and shows two vectors computed for the document : a tf*idf vector and a normalized tf*idf vector where the normalized values are obtained dividing the tf*idf of each term by the vector's Euclidean norm.

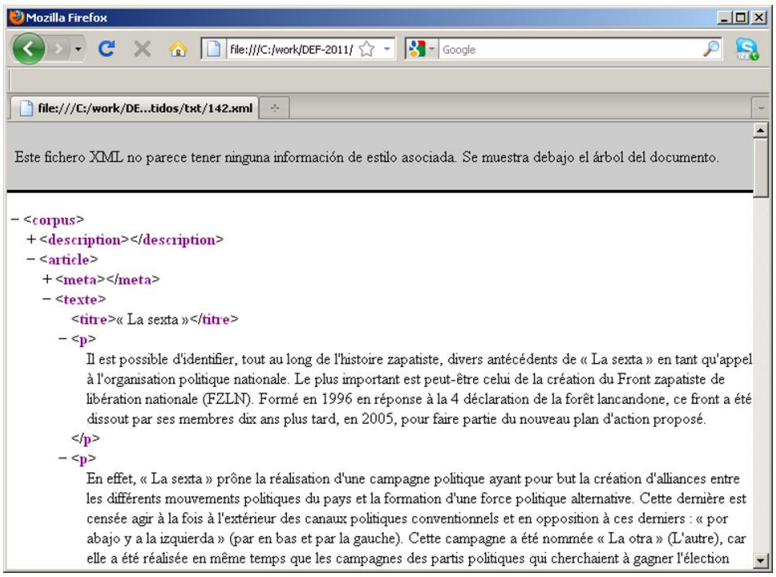


FIG. 3 – Article w/o introduction/conclusion used during training

2.3 Text Matching

Given two document collections $A = \{A_1, ..., A_n\}$ and $D = \{D_1, ..., D_n\}$ we carry out pair-wise comparison of documents in A with documents in D and create a similarity matrix (Figure 4) $M_{i,j}$ for $1 \leq i, j \leq n$ containing the similarity between document i (e.g., an abstract) and document j (e.g., a full article or article without introduction/conclusion). The similarity metric we use in this work to compare the documents is the cosine of the angle between two vectors. This metric gives value one for identical vectors and zero for vectors which are orthogonal (no related). The exact formula we use is as follows :

$$\text{cosine}(d_1, d_2) = \frac{\sum_{i=1}^{n} w_{i,d_1} * w_{i,d_2}}{\sqrt{\sum_{i=1}^{n} (w_{i,d_1})^2} * \sqrt{\sum_{i=1}^{n} (w_{i,d_2})^2}}$$

Here d_1 and d_2 are document vectors and w_{i,d_k} is the weight of term i in document d_k (i.e., the tf*idf values). For the experiment reported here we have used the normalized vectors.

System ID	Score
1 ; 5 ; 8 ; 11	1.000
6	0.995
2	0.980
3 ; 4	0.975
10	0.970
7	0.965
9	0.909

TAB. 1 – Results for systems matching abstracts (RES) with articles (ART)

2.4 Selecting a Candidate Summary

For each abstract A_i in A we select a document D_j in D such that $M_{i,j} \geq M_{i,k} \forall k$. For the response without confidence scores, we returned a set of pairs (Abstract$_i$,Document$_j$), for the response with confidence scores,

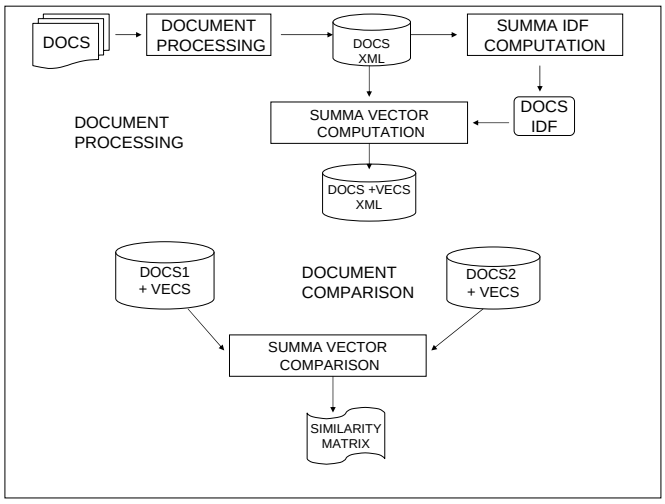


FIG. 4 – Document processing for DEFT 2011

System ID	Score
5 ; 8	1.000
11	0.995
6	0.990
4	0.964
1 ; 3	0.959
2	0.954
7	0.919
10	0.904
9	0.873

TAB. 2 – Results for systems matching abstracts (RES) with articles without introduction or conclusion (TXT)

we returned the best (in terms of similarity to the document) 5 triples (Abstract$_i$,Document$_{j_k}$,Confidence$_{i,j_k}$) for $k = 1, ...5$) for each abstract ; where Confidence$_{i,j_k}$ is the normalized similarity computed as :

$$\text{Confidence}_{i,j_k} = \frac{M_{i,j_k}}{\sum_{l=1,...5} M_{i,j_l}}$$

3 Experimental Results

Each system response (response$_i$ = *(Abstract$_i$,Document$_j$)*) is compared to the true response and evaluated according to the following formula :

$$\text{score(response}_i) = \begin{cases} 1 & \text{if the match is correct} \\ 0 & \text{otherwise} \end{cases}$$

The system final score is computed as :

$$\text{score(S)} = \frac{\sum_{i=0}^{n} \text{score(response}_i)}{n}$$

System ID	Score
3	0.512
10	0.417

TAB. 3 – Results for systems matching abstracts (RES) with articles (ART) with confidence scores

System ID	Score
3	0.402
10	0.368

TAB. 4 – Results for systems matching abstracts (RES) with articles w/o introduction/conclusion (TXT) with confidence scores

where n is the number of responses.

We also provided a set of 5 answers per abstract each with a confidence score as previously described (response$_{i,j_k}$ = *(Abstract$_i$, Document$_j$, Confidence$_{i,j_k}$)*). In this case the scores are computed as :

$$\text{confidence\_score(response}_{i,j_k}) = \begin{cases} \text{Confidence}_{i,j_k} & \text{if the match is correct} \\ 0 & \text{otherwise} \end{cases}$$

The system final confidence score is computed as :

$$\text{Confidence Score(S)} = \frac{\sum_{i=0}^{n} \text{confidence\_score(response}_{i,j})}{n}$$

where n is the number of responses.

Tables 1 and 2 show respectively results for all systems for matching abstracts with full articles and abstracts with articles without introductions or conclusions. Systems with similar performance are grouped together. Our system ID is number 3 which obtains a reasonable score in both tasks being placed in the middle of both tables. Since the same strategy was used to select ART and TXT responses and the scores for TXT are lower, it appears that this latter task is slightly more difficult. It is however evident that all systems are able to solve the proposed task.

Where the response with confidence scores is concerned, obtained results are shown on Tables 3 and 4. In addition to our system (number 3) only other team provided answers with confidence scores (number 10). Results are much lower than those obtained for unique answers because of the low confidence given to the best matching solution : the first solutions is almost always the correct one and confidence computation should take into consideration this fact.

4 Discussion

This paper has described our first participation in the DEFT evaluation campaign. We have developed a solution able to identify the document where an abstract comes from with very high accuracy using available and *quasi* "language independent" tools, in fact only the tokenization process we have used is language dependent, no sophisticated resources such as morphological analysers or stemmers have otherwise been used. We therefore argue that our solution could be easily ported to similar tasks in other Romance languages and to English. Because most systems were able to obtain good accuracy we argue that the task has to be made harder by considering document collections which are closer in content and form or by considering the cross-lingual matching problem.

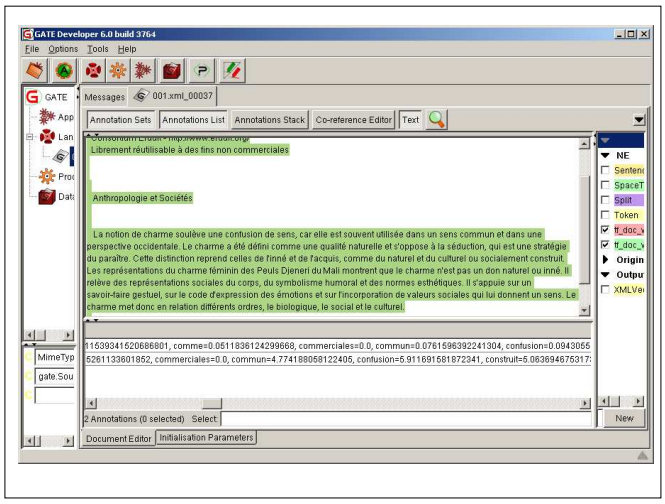


FIG. 5 – Document with normalized and non-normalized vectors computed.

Acknowledgements

We are grateful to Programa Ramón y Cajal 2009 from Ministerio de Ciencia e Innovación, Spain. We are grateful to Simon Mille who helped us with the French abstract. Experiments have been done in the UPF-CIBER-BBN cluster for High Performance Computing, massive storage and software for Biomedical Applications which is part of the Biomedical Research Services of CIBER-BBN `http://www.ciber-bbn.es` and UPF `http://www.upf.edu`.

Références

GROUIN C., BERTHELIN J.-B., AYARI S. E., HURAULT-PLANTET M. & LOISEAU S. (2008). Présentation de deft'08 (défi fouille de textes). In *Actes de JEP–TALN–RECITAL 2008*. 13 juin 2008.

GROUIN C., HAURAULT-PLANTET M., PAROUBEK P. & BERTHELIN J.-B. (2009). Deft'07 : une campagne d'évaluation en fouille d'opinion. *RNTI*, **E-17**.

MAYNARD D., TABLAN V., CUNNINGHAM H., URSU C., SAGGION H., BONTCHEVA K. & WILKS Y. (2002). Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data*, **8**(2/3), 257–274.

SAGGION H. (2008a). Experiments on semantic-based clustering for cross-document coreference. In *Proceedings of the Third Joint International Conference on Natural Language Processing*, p. 149–156, Hyderabad, India : AFNLP AFNLP.

SAGGION H. (2008b). SUMMA : A Robust and Adaptable Summarization Tool. *Traitement Automatique des Langues*, **49**(2), 103–125.

SAGGION H. & GAIZAUSKAS R. (2004). Multi-document summarization by cluster/profile relevance and redundancy removal. In *Proceedings of the Document Understanding Conference 2004* : NIST.

SAGGION H. & LAPALME G. (2002). Generating Indicative-Informative Summaries with SumUM. *Computational Linguistics*.

SALTON G. (1988). *Automatic Text Processing*. Addison-Wesley Publishing Company.

~~L~~*SVM*A : au plus deux composants pour apparier des résumés à des articles

Yves Bestgen

UCL/CECL/IPSY, B-1348 Louvain-la-Neuve, Belgique
yves.bestgen@psp.ucl.ac.be

Résumé

Dans le cadre de la tâche 2 de DEFT2011 qui consiste en l'appariement d'articles scientifiques avec le résumé correspondant, une approche, dénommée LSVMA, est proposée. Elle est basée sur trois composants : l'analyse sémantique latente (LSA), les machines à support vectoriel (SVM) et l'assignation finale selon l'algorithme du meilleur d'abord (MA). Cette approche a permis d'apparier parfaitement les résumés aux articles. Des analyses complémentaires montrent que le composant LSA n'est pas indispensable pour relever efficacement le défi. Par contre, une optimisation de l'assignation effectuée par la SVM est nécessaire, à tout le moins pour les options et paramètres testés. Le caractère superflu de LSA pour la tâche proposée contraste avec le rôle qu'il joue dans les systèmes d'évaluation automatique de résumés. Cette recherche ne permet toutefois pas de décider si cette conclusion est spécifique au présent défi ou si elle peut être généralisée à d'autres tâches mettant en jeu l'évaluation automatique de résumés.

Abstract

Within the framework of DEFT2011 second task, which consists in the pairing of a scientific article with its corresponding abstract, an approach, called LSVMA, is proposed. It is based on three components: Latent Semantic Analysis, Support Vector Machines (SVM) and a final assignment step based on the best-first algorithm (*Meilleur d'Abord* : MA). This approach made it possible to pair the summaries with the articles without any error. Complementary analyses show that LSA is not necessary to take up the challenge effectively while an optimization of the assignment carried out by the SVM is necessary, at least for the options and parameters tested. The unnecessary character of LSA for this task contrasts with the role it plays in automatic summary grading. This study does not make it possible to decide whether this conclusion is specific to the present challenge or can be generalized to other tasks.

Mots-clés : Evaluation automatique de résumés, Machines à support vectoriel, Analyse sémantique latente, Problème d'affectation, Meilleur d'abord
Keywords: Automatic summary grading, Support Vector Machines, Latent Semantic Analysis, Assignment problem, Best-first

1 Introduction

La tâche 2 de DEFT2011 consiste en l'appariement d'articles scientifiques avec le résumé correspondant. Selon le point de vue adopté, de nombreuses approches sont envisageables pour relever ce défi. Si, par exemple, on s'appuie sur le fait que l'auteur d'un résumé est (très probablement) aussi l'auteur de l'article en question, des procédures développées afin d'identifier l'auteur d'un texte sont tentantes. Partir du postulat que nombre de résumés sont rédigés lorsque l'article est terminé et qu'ils réutilisent des passages de cet article conduit à se tourner vers des procédures efficaces dans le cadre de la détection de la "réutilisation" de passages dans différents textes. L'évidente parenté thématique entre le résumé et l'article conduit, quant à elle, à privilégier le recouvrement en termes de contenu. C'est, par exemple, le point de vue choisi par Foltz, Britt et Perfetti (Foltz, 1996) dans leur étude visant à déterminer au moyen d'une procédure automatique, basée sur l'analyse sémantique latente (LSA : Latent Semantic Analysis), quels textes ont le plus influencé les résumés produits après la lecture d'un grand nombre de textes abordant un même sujet.

L'approche évaluée dans ce rapport s'inscrit prioritairement dans cette troisième option parce qu'elle semble être la plus pertinente dans le cadre des travaux sur l'évaluation automatique de la qualité d'un résumé. Comme le soulignent les organisateurs de DEFT[1], évaluer un résumé est une question complexe et importante (Das, Martin, 2007). Elle est également au centre d'un champ de recherches très dynamique en éducation comme l'atteste le développement d'outils pour l'évaluation des résumés produits par des étudiants et celui de tutoriels visant à les aider à améliorer leurs résumés (Franzke et al., 2005 ; He et al., 2009 ; Kintsch et al., 2000 ; Miller, 2003 ; Olmos et al., 2009 ; Wade-Stein, Kintsch, 2004). Cet intérêt pour les résumés trouve son origine dans les processus cognitifs sous-jacents à cette activité qui conduisent l'étudiant à porter une attention toute particulière aux informations les plus importantes d'un texte et à intégrer celles-ci avec ses connaissances antérieures, deux processus primordiaux pour un apprentissage efficace par la lecture (Wade-Stein, Kintsch, 2004). Dans le cadre de ce défi, l'idée à l'origine de l'approche proposée est qu'une procédure qui fonctionne pour évaluer des résumés devrait donner une meilleure "note" au résumé du texte qu'aux résumés d'autres textes. La section suivante décrit cette approche. Les analyses et résultats obtenus sont présentés dans la troisième section. Ensuite, l'utilité des composants LSA et Assignation au meilleur d'abord pour l'efficacité de la procédure est évaluée au travers d'analyses complémentaires.

2 Description de l'approche

Pour évaluer la qualité de résumés produits par des étudiants, l'approche classique consiste à comparer ces résumés à un document de référence au moyen d'une mesure de similarité, les meilleurs résumés étant ceux qui ressemblent le plus au document de référence. Dans ces travaux, la similarité est habituellement obtenue par l'entremise d'une analyse sémantique latente, une technique mathématique qui vise à extraire un espace sémantique à partir de l'analyse statistique des cooccurrences dans un corpus de textes (Deerwester et al., 1990 ; Landauer et al. 1998)[2]. Le corpus employé pour extraire les dimensions sémantiques peut être composé d'un très grand nombre de textes censés être représentatifs des documents lus par des étudiants (Tasa corpus, 11 millions de mots) ou d'un corpus plus petit, mais composé de textes thématiquement similaires à ceux qui doivent être évalués (Kintsh et al., 2000). Il peut également être composé des seuls documents analysés (He et al., 2009). Quant au document de référence, il s'agit soit d'un résumé-idéal, le plus souvent rédigé par un expert du domaine sur lequel le texte porte, ou du texte qui devait être résumé. Lorsque le document de référence est le texte initial, la comparaison peut se faire sur la base du texte considéré comme un seul document ou sur la base des différentes sections ou des paragraphes qui le composent. Ces différentes options ont été combinées de très nombreuses manières (León et al., 2005), le cas le plus extrême étant probablement la technique proposée par He et al. (2009) qui découpe le résumé-cible et le résumé-idéal en phrases, procède à une analyse sémantique latente indépendante de ces deux micro-documents et obtient, sur cette base, les similarités entre les phrases qui composent les deux

[1] http://deft2011.limsi.fr/index.php?id=4&lang=fr

[2] LSA est aussi une des techniques employées pour générer automatiquement le résumé d'un texte, voir par exemple Gong et Liu (2001) ou Steinberge et Jezek (2004), mais ce sujet ne sera pas abordé ici parce que l'approche qui consiste à générer le résumé des articles et à ensuite le comparer aux résumés potentiels n'a pas été suivie.

documents. Les études qui ont comparé ces options mettent en évidence peu de différences entre elles (León et al., 2005 ; Olmos et al., 2009).

Ces observations laissent penser qu'une approche basée sur la comparaison des résumés aux différentes sections des textes en employant un espace sémantique spécifique à ce matériel devrait être efficace pour effectuer la tâche d'appariement. Cette tâche présente toutefois une spécificité par rapport à l'évaluation de résumés qui justifie une adaptation de la procédure : l'existence d'une correspondance biunivoque entre l'ensemble des résumés et l'ensemble des textes. Afin de tirer profit de cette particularité, l'approche classique qui s'appuie sur une mesure de similarité entre les résumés et les segments de textes a été remplacée par une procédure de classification supervisée qui apprend, à partir des informations issues de LSA, à classer correctement des paragraphes en fonction du texte dont ils ont été extraits et qui est ensuite appliquée à la catégorisation des résumés selon les mêmes principes (pour d'autres études qui combinent LSA et SVM, voir, par exemple, Bechet et al. (2008) ou Kwok (1998)). Il est à noter que la procédure supervisée employée ne requiert pas un échantillon d'apprentissage pour lequel l'appariement entre les textes et les résumés est connu. En effet, la procédure est appliquée aux paragraphes des textes pour lesquels l'appariement est toujours connu : chaque texte est considéré comme une catégorie et chaque paragraphe comme une instance de cette catégorie. La prédiction est faite sur les résumés qui sont catégorisés dans une des catégories correspondant à un texte.

2.1 Principaux composants

L'approche proposée, dénommée **L*SVMA***, repose sur trois composants : l'analyse sémantique latente (Latent Semantic Analysis : LSA), les machines à support vectoriel (SVM) et l'affectation par la technique du Meilleur d'Abord (MA) qui tire profit de la relation biunivoque entre les résumés et les articles. Ces trois composants sont décrits dans la présente section.

2.1.1 LSA

Comme indiqué ci-dessus, LSA est une technique mathématique qui vise à extraire un espace sémantique à partir de l'analyse statistique des cooccurrences dans un corpus de textes. Le point de départ de l'analyse est une matrice qui contient le nombre d'occurrences de chaque terme dans chaque document, un document pouvant être un texte, un paragraphe, une phrase ou même une suite de mots d'une longueur arbitraire. Après normalisation (optionnelle) de la matrice de fréquences, celle-ci fait l'objet d'une décomposition en valeurs singulières. La matrice initiale (X) est décomposée en un produit de trois matrices (TSD') comme illustré à la figure 1. S est une matrice diagonale correspondant aux valeurs singulières et T et D sont des matrices orthogonales correspondant respectivement aux vecteurs singuliers pour les termes et pour les documents. En ne retenant que les *k* plus grandes valeurs singulières, on peut reconstruire une version compressée de la matrice originale dans laquelle seules les dimensions les plus importantes ont été conservées. Dans la suite des analyses, ce sont les *k* vecteurs singuliers de la matrice D qui sont employés comme traits dans la SVM afin d'apprendre à classer les paragraphes en fonction de l'article d'origine.

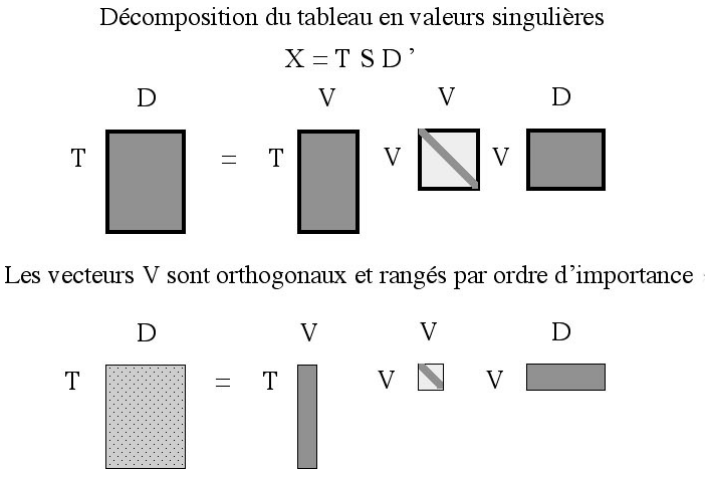


Figure 1 : Décomposition en valeurs singulières d'une matrice de fréquences Termes*Documents

Pour effectuer la décomposition en valeurs singulières de la matrice de fréquences (après pondération par la formule classique de log-entropie, Dumais, 1991), j'ai employé le programme *SVDPACKC*[3] (Berry et al., 1993), procédure LAS2 (*Single Vector Lanczos Method*).

2.1.2 SVM

L'algorithme d'apprentissage utilisé est une machine à support vectoriel (SVM) connue pour son efficacité en catégorisation supervisée de textes (Joachims, 2002). Dans sa forme la plus classique, cet algorithme apprend à classer un ensemble d'exemplaires en deux catégories sur la base de traits qui correspondent ici aux vecteurs singuliers issus de LSA. Dans le cas présent, le problème est multicatégoriel puisqu'il s'agit de catégoriser chaque paragraphe (lors de l'apprentissage) et chaque résumé (lors du test) dans une des catégories correspondant à chaque fois à un article. Plusieurs approches ont été proposées pour appliquer les SVM à ce type de situations comme l'emploi d'une série de classifieurs binaires (en "un contre tous" ou en "un contre un") ou d'une approche intrinsèquement multiclasse (Crammer, Singer, 2001). C'est cette dernière option qui a été choisie ici en raison de la simplicité de sa mise en pratique. Pour effectuer cette analyse, le logiciel SVM^{multiclass}[4] de Joachims (Joachims et al., 2009) a été employé. Le paramètre de régularisation *C*, qui détermine le rapport entre la taille de la marge et le nombre d'erreurs tolérées lors de l'apprentissage, a été fixé à 1 (mais voir la section 4 pour une analyse d'autres valeurs).

Une limitation des SVM pour le problème à résoudre est que la classification finale est effectuée de manière indépendante pour chaque résumé, ce qui ne garantit pas que chaque texte ne se verra affecter qu'un seul résumé. Le troisième composant tente d'optimiser l'approche en tirant profit de cette propriété de la tâche.

2.1.3 Appariement par le meilleur d'abord

Pour chaque résumé, la procédure SVM attribue, au travers des valeurs de décision, un score de comptabilité avec chacune des catégories et donc avec chaque article. On obtient donc une sorte de matrice de similarité Résumés*Textes et il s'agit d'apparier le plus correctement possible chaque résumé à un et un seul texte. Cette situation peut être vue comme un classique problème d'affectation dans lequel des tâches doivent être assignées à des agents pour un coût minimal (ou un bénéfice maximal). Le même genre de problème se rencontre en méthodologie de la recherche dans les études observationnelles qui visent à comparer deux groupes comme des personnes "malades" et des personnes "saines" (Rubin, 1973). Afin d'accroître l'efficacité des analyses, on y procède à l'appariement des participants du groupe cible à ceux du groupe contrôle afin de constituer les paires dont les membres sont les plus similaires possible. Différentes techniques ont été proposées par Rosenbaum et Rubin (1985). La plus simple consiste à choisir aléatoirement un membre du groupe cible et à l'apparier au membre du groupe témoin qui lui est le plus similaire. Ensuite, chaque membre de cette paire est retirée du groupe respectif et la procédure est réappliquée jusqu'à ce que tous les membres du groupe cible soient associés à un membre du groupe contrôle. Comme le souligne Rosenbaum (2010), une telle approche n'est pas nécessairement optimale au sens qu'elle ne garantit pas une similarité maximale globale entre les couples ainsi formés parce qu'elle ne prend pas en compte le fait que l'appariement effectué à une étape modifie les appariements possibles lors des étapes ultérieures. Dans certaines situations, elle est néanmoins aussi efficace que l'optimisation globale. On peut penser que le cas considéré ici fait partie de ces situations puisque l'objectif n'est pas de construire un appariement à partir de données disparates, mais bien de retrouver l'appariement pré-existant aux analyses. Pour cette raison, la procédure d'appariement par paire selon l'approche du plus proche disponible (*nearest available pair-matching* ou *greedy*, Hansen, 2004) en suivant l'ordre du *Meilleur d'Abord* (MA) a été utilisée. Elle consiste simplement à associer en premier lieu le résumé et l'article qui ont le meilleur score de compatibilité dans toute la matrice, à retirer ce couple de la matrice et à réitérer cette procédure jusqu'à ce que tous les couples aient été formés. Comme indiqué ci-dessus, cette procédure ne conduit pas nécessairement à une solution globalement optimale et est donc potentiellement optimisable au moyen, par exemple, de la méthode hongroise proposée par Kuhn pour résoudre les problèmes d'assignation (Rosenbaum, 2010).

[3] http://www.netlib.org/svdpack/svdpackc.tgz

[4] http://svmlight.joachims.org/svm_multiclass.html

2.2 Implémentation

Cette section décrit différents aspects de l'implémentation qui n'ont pas été discutés dans la section précédente. Il est à noter que les analyses décrites ci-dessous ont été réalisées séparément pour chaque revue puisqu'une balise spécifiant la revue dont a été extrait le document était associée à chaque résumé et à chaque article et que les règles du défi indiquaient que cette information pouvait être employée.

2.2.1 Prétraitement

Dans un premier temps, les documents ont été lemmatisés au moyen du programme *TreeTagger* de Schmid (1994). La suite des analyses a été effectuée sur les formes lemmatisées, sauf lorsque le mot était inconnu du tagger et, dans ce cas, la forme originale a été conservée. Les signes de ponctuation et les nombres, identifiés par le tagger, ont été supprimés.

Ensuite, les articles ont été segmentés en paragraphes. Cette segmentation est basée sur les balises *<titre>* et *<p>*. Les titres étant très courts, chacun d'entre eux a été systématiquement associé au paragraphe qui le suit directement. Pour les paragraphes eux-mêmes, la situation est un peu plus complexe parce que la balise *<p>* est présente, dans certains textes, au début de chaque ligne et qu'elle est également employée pour isoler des exemples. On a donc décidé de fixer une taille minimale de 75 mots aux paragraphes et de regrouper tout paragraphe trop petit avec celui ou ceux qui le suivent (à l'exception du ou des derniers paragraphes qui, s'ils n'atteignaient pas la longueur minimale, étaient réunis aux paragraphes qui les précèdent). Dans la suite, les unités ainsi formées sont appelées paragraphes. Aucune segmentation n'a été appliquée aux résumés.

Après la segmentation, tous les déterminants définis et indéfinis ont été supprimés. Par contre, tous les autres mots fonctionnels comme les pronoms ou les conjonctions, ont été conservés. Le résultat de cette étape est, pour chaque revue, une matrice de fréquence des termes dans tous les paragraphes et dans tous les résumés. C'est cette matrice, après application de la classique pondération log-entropie, qui a été soumise à LSA. Une autre option, légèrement plus complexe, aurait été de ne soumettre à la SVD que les paragraphes et de positionner ultérieurement les résumés dans cet espace (en calculant la somme pondérée des vecteurs représentant les termes qui les composent).

2.2.2 Options d'optimisation

L'implémentation présentée ci-dessus inclut une série de paramètres et d'éléments optionnels qui peuvent être vus comme autant d'occasions pour tenter d'optimiser la procédure (voir par exemple Bestgen (2004) pour une analyse de l'impact de ces paramètres sur l'efficacité de LSA). Parmi ceux-ci, un paramètre mérite une attention particulière parce qu'il est connu pour affecter l'efficacité d'une procédure basée sur l'analyse sémantique latente et qu'il affecte directement la procédure de catégorisation par la SVM : le nombre de dimensions de l'espace sémantique pris en compte (*k* dans la section 2.1.1). Trois cents est généralement considéré comme un optimum, tout particulièrement lorsqu'on traite un très grand corpus (Landauer et al., 2004). Toutefois, lorsque les analyses sont effectuées sur des corpus spécifiques (textes d'un contenu similaire au texte cible), un nombre plus réduit de vecteurs est conservé (Olmos et al., 2009). Dans le cas présent, le matériel de développpement pour une des revues n'est composé que de 1297 documents (paragraphes et résumés, voir Tableau 1), ce qui correspond à un matériel d'à peine 650 documents pour la phase de test. Il a donc été décidé de comparer l'efficacité de la procédure pour 300 vecteurs singuliers, mais aussi pour 200 et 100.

3 Analyses et résultats

3.1 Analyse du matériel de développement

Le tableau 1 présente le matériel qui a servi pour le développement de la procédure et ce pour les deux pistes du défi, la piste 1 celle qui porte sur le texte complet de l'article et la piste 2 pour laquelle

l'introduction et la conclusion de l'article ont été supprimées du texte à apparier. L'abréviation employée pour faire référence aux différentes revues dans la suite est donnée en dessous du nom de celle-ci.

		Piste 1		Piste 2	
Revue	Couples	Paragraphes	Termes différents	Paragraphes	Termes différents
Anthropologie et Sociétés (ANH)	60	1982	8952	1558	8439
Meta (MET)	59	2569	8652	2330	8500
Revue des sciences de l'éducation (SCI)	60	2427	7072	2078	7003
Études internationales (INT)	60	2546	7984	2004	7848
Études littéraires (LIT)	60	1748	9319	1297	8285

Tableau 1 : Description du matériel de développement

3.1.1 Efficacité de SVM^{multiclass} lors du développement : catégorisation des paragraphes

Le Tableau 2 présente le pourcentage de paragraphes classés correctement par la procédure SVM selon le nombre de vecteurs singuliers employés (Nvec). Il s'agit des valeurs obtenues lors de l'apprentissage et donc sans application d'une procédure de validation croisée puisque la vraie évaluation de l'approche sera effectuée au travers de l'appariement des résumés aux textes. On observe que les vecteurs qui occupent les rangs allant de 101 à 300 apportent une contribution non négligeable à la catégorisation puisque pour une des cinq revue le pourcentage de bien classés n'est que de 81% pour 100 vecteurs alors qu'il est de à 94% pour 300.

Piste	Nvec	ANH	MET	SCI	INT	LIT
1	100	92.58	80.03	93.78	94.85	94.97
	200	96.97	89.96	96.54	96.90	98.40
	300	98.13	94.08	97.73	97.84	98.91
2	100	93.32	80.60	93.41	94.06	94.45
	200	97.56	90.73	96.44	96.56	98.30
	300	98.59	93.73	97.64	97.95	99.23

Tableau 2 : Pourcentage de classifications correctes lors de l'apprentissage

3.1.2 Efficacité de la procédure lors du développement : appariement résumé-texte

Le Tableau 3 présente le pourcentage d'appariements corrects pour les deux pistes selon le nombre de vecteurs singuliers (Nvec) mis à la disposition de la SVM. On observe que la revue *Meta* est la seule à

poser problème à LSVMA et ce seulement pour 100 ou 200 vecteurs singuliers. Cette observation peut être mise en relation avec la moindre performance du classifieur SVM pour cette revue lors de l'apprentissage et suggère qu'il pourrait être pertinent d'accroître le nombre de vecteurs singuliers pris en compte par la SVM. Toutefois, l'obtention d'une performance parfaite avec 300 vecteurs singuliers pour chacune des deux pistes du défi a rendu peu pertinente toute tentative d'optimisation des paramètres.

Piste	Nvec	ANH	MET	SCI	INT	LIT
1	100	100	89.83	100	100	100
	200	100	100	100	100	100
	300	100	100	100	100	100
2	100	100	89.83	100	100	100
	200	100	96.61	100	100	100
	300	100	100	100	100	100

Tableau 3 : Pourcentage d'appariements corrects

3.2 Analyse du corpus de test

Le matériel de test pour les deux pistes a été analysé, au moyen de la procédure décrite ci-dessus et, donc, en faisant varier le nombre de vecteurs singuliers mis à disposition de la SVM (100, 200 et 300). Ces trois analyses ayant produit les mêmes appariements, une seule soumission a été envoyée pour chaque piste. Les résultats, transmis par les organisateurs du défi, indiquent que les appariements proposés par LSVMA sont tous corrects, un niveau de performance également atteint par d'autres équipes, et ce malgré la présence d'une revue qui n'était pas incluse dans le matériel d'entraînement.

4 Evaluation de l'utilité de deux des trois composants

Des trois composants de l'approche proposée, seul SVM est indispensable parce c'est ce composant qui assigne, au moins provisoirement, les résumés aux articles. Il serait bien sûr possible de s'en passer, par exemple en calculant une mesure de proximité entre les paragraphes et les résumés, mais il s'agirait là d'une tout autre approche. Les deux autres composants sont facultatifs. Au lieu d'employer les vecteurs singuliers obtenus par LSA, SVM peut utiliser directement les termes comme traits et il n'est pas prouvé que l'assignation des résumés aux textes effectuée par SVM, n'est pas optimale, ce qui rendrait le passage par la procédure d'assignation au meilleur d'abord inutile. Afin de déterminer si ces deux composants sont utiles ou non pour effectuer la tâche, des analyses complémentaires ont été menées.

4.1 Nécessité de LSA?

Une des principales raisons de l'emploi de LSA pour estimer la similarité entre des documents est que cette technique permet de considérer comme similaires des documents même si ceux-ci n'ont pas de mots en commun (Miller, 2003). Cette propriété est très importante pour l'évaluation automatique de résumés ou le développement de tutoriels visant à aider des étudiants à écrire de meilleurs résumés puisqu'on s'attend à ce que les étudiants emploient leurs propres mots lors de la rédaction. Dans le cadre de ce défi, il est, par contre, peu probable qu'un résumé contienne un vocabulaire différent du texte correspondant. Il s'ensuit que LSA n'est peut-être pas nécessaire pour réaliser efficacement la tâche d'appariement.

Pour évaluer l'impact de LSA sur l'efficacité, ce composant a été retiré de la procédure. La SVM a donc été appliquée à la matrice Paragraphes*Termes (pondérée par log-entropie) issue directement des

prétraitements. Ce sont donc les termes qui servent de traits pour l'apprentissage et non les vecteurs singuliers issus de LSA.

Ces analyses ont d'abord été effectuées sur le matériel de développement en employant plusieurs valeurs pour le paramètre *C* de la SVM. Avec *C* fixé à 1 (comme dans les analyses précédentes), cette version de la procédure obtient un pourcentage d'appariements corrects de 97.32% pour les deux pistes du défi (articles entiers et articles tronqués). Avec un *C* fixé à 100, on obtient plus de 99% d'appariements corrects et 100% pour *C* = 1 000 ou *C* = 5 000 et ce, toujours, pour les deux pistes[5].

Des résultats identiques ont été obtenus avec le matériel de test pour *C* = 1 000 et *C* = 5 000. Il s'ensuit qu'avec une valeur *C* suffisamment grande, la procédure sans le composant LSA est aussi efficace qu'avec ce composant et que, donc, celui-ci n'est pas nécessaire pour la tâche en jeu. C'est la raison pour laquelle le L de ~~L~~*SVM*A a été biffé dans le titre de l'article.

4.2 Nécessité de l'assignation au meilleur d'abord?

Pour évaluer l'utilité du composant d'assignation finale au meilleur d'abord, on a déterminé l'efficacité des procédures LSA+SVM (avec 300 vecteurs singuliers) et SVM sans LSA (pour *C* = 1 000 et *C* = 5 000) avant que le composant Assignation ne soit appliqué. Tant sur le matériel de développement que sur le matériel de test, aucune des procédures n'a donné lieu à une performance parfaite. Il s'ensuit que ce composant améliore bien l'efficacité de la SVM.

4.3 Discussion

Les résultats présentés dans cette section ne peuvent être considérés comme "définitifs" parce, comme indiqué ci-dessus, l'approche proposée inclut un grand nombre d'options et de paramètres et qu'aucune étude systématique de leur impact n'a été réalisée. Il est donc possible que le composant "assignation au meilleur d'abord", déclaré "nécessaire", ne le soit plus lorsqu'un jeu plus optimal de paramètres est employé. Par contre, le fait qu'un composant (LSA dans le cas présent) se révèle non indispensable n'est pas remis en question par cette argumentation. On notera toutefois qu'il est possible que l'analyse sémantique latente se révèle aussi efficace que l'approche présentée ici si elle employée, non au travers d'une procédure SVM, mais par des approches plus classiques d'évaluation de résumés et optimisée dans ce cadre (Olmos et al., 2009).

5 Conclusion

Dans le cadre de la tâche 2 de DEFT2011 qui consiste en l'appariement d'articles scientifiques avec le résumé correspondant, nous avons proposé une approche basée sur trois composants : l'analyse sémantique latente, une machine à support vectoriel et l'assignation finale selon l'algorithme du meilleur d'abord. Cette approche a permis d'apparier parfaitement les résumés aux articles, et ce pour les deux pistes du défi, celle qui porte sur le texte complet de l'article et celle pour laquelle l'introduction et la conclusion de l'article ont été supprimées du texte à apparier. Des analyses complémentaires ont montré que le composant LSA n'est pas indispensable pour relever efficacement le défi. Par contre, une optimisation de l'assignation effectuée par la SVM est nécessaire, à tout le moins pour les options et paramètres testés. Le caractère superflu de LSA pour la tâche proposée contraste nettement avec le rôle qu'il joue classiquement dans les systèmes d'évaluation automatique de résumés. Cette recherche ne permet toutefois pas de décider si cette conclusion est spécifique au présent défi ou si elle peut, en partie au moins, être généralisée à d'autres tâches mettant en jeu l'évaluation automatique de résumés.

[5] Les analyses présentées à la section 3 ont été répétées en faisant également varier le paramètre *C*. Pour le matériel de test, les résultats sont identiques à ceux rapportés à cette section. Pour le matériel de développement, l'emploi d'une valeur *C* très élevée permet d'améliorer l'efficacité de l'approche pour 100 vecteurs singuliers.

Remerciements

Yves Bestgen est chercheur qualifié du Fonds de la recherche scientifique (FNRS) de la Communauté Wallonie-Bruxelles (Belgique).

Références

BECHET, N., ROCHE, M., CHAUCHE, J. (2008). ExpLSA et classification de textes, Actes des *9es Journées internationales d'Analyse statistique des Données Textuelles*, 167-177.

BERRY, M.. DO, T., O'BRIEN, G., KRISHNA, V., VARADHAN, S. (1993). SVDPACKC: Version 1.0 User's Guide, *Tech. Rep. CS-93-194*, University of Tennessee, Knoxville, TN.

BESTGEN, Y. (2004). Analyse sémantique latente et segmentation automatique des textes, Actes des *7es Journées internationales d'Analyse statistique des Données Textuelles*, 171-181.

CRAMMER, K., SINGER, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines, *Journal of Machine Learning Research* 2, 265-292.

DAS, D., MARTINS, A.F.T. (2007). A survey on automatic text summarization. Literature Survey for the Language and Statistics II course at CMU.

DEERWESTER, S., DUMAIS, S.T., FURNAS, G.W., LANDAUER, T.K., HARSHMAN, R. (1990). Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science* 41, 391-407.

DUMAIS, S. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods* 23, 229-236.

FOLTZ, P.W. (1996). Latent Semantic Analysis for text-based research. *Behavior Research Methods* 28, 197-202.

FRANZKE, M., KINTSCH, E., CACCAMISE, D., JOHNSON, N., DOOLEY. S. (2005). Summary Street ® : Computer support for comprehension and writing. *Journal of Educational Computing Research 33* , 53-80.

GONG, Y., LIU, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. Proceedings of the *24th annual international ACM SIGIR conference on Research and development in information retrieval*, 19-25

HANSEN, B.B. (2004). Full matching in an observational study of coaching for the SAT. *Journal of the American Statistical Association* 99, 609-618.

HE, Y., HUI, S.H., QUAN, T.T. (2009). Automatic summary assessment for intelligent tutoring systems. *Computers & Education* 53, 890–899.

JOACHIMS T. (2002). *Learning to classify text using support vector machines: Methods, theory and algorithms*, Dordrecht, Kluwer.

JOACHIMS, T., FINLEY, T., YU, C. (2009). Cutting-Plane training of structural SVMs, *Machine Learning* 77, 27-59.

KINTSCH, E., STEINHART, D., STAHL, G., LSA RESEARCH GROUP, MATTHEWS, C., LAMB, R. (2000). Developing summarization skills through the use of LSA-based feedback. *Interactive Learning Environments 8*, 87-109.

KWOK, J.T. (1998). Automated text categorization using support vector machine. Proceedings of *ICONIP'98*, 347–351.

LANDAUER, T.K., FOLTZ, P.W., LAHAM, D. (1998), An introduction to latent semantic analysis. *Discourse Processes* 25, 259-284.

LANDAUER, T.K., LAHAM, D., DERR, M. (2004). From paragraph to graph: Latent Semantic Analysis for information visualization. Proceedings of the *National Academy of Science* 101, 5214-5219.

LEON, J., OLMOS, R., ESCUDERO, I., CAÑAS, J., SALMERON, L. (2005). Assessing short summaries with human judgments procedure and Latent Semantic Analysis in narrative and expository texts. *Behavior Research Methods* 38, 616-627.

MILLER, T. (2003). Essay assessment with Latent Semantic Analysis. *Journal of Educational Computing Research* 29, 495-512.

OLMOS, R., LEON, J.A., BOTANA, G.J., ESCUDERO, I. (2009) New algorithms assessing short summaries in expository texts using latent semantic analysis, *Behavior Research Methods* 41, 944-950

ROSENBAUM, P.R. (2010). *Design of Observational Studies*. Springer, New York.

ROSENBAUM, P.R., RUBIN, D. (1985), Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician* 39, 33–38.

RUBIN, D. B. (1973), Matching to remove bias in observational studies, *Biometrics* 29, 159-183.

SCHMID, H., (1994). Probabilistic part-of-speech tagging using decision trees. Proceedings of the *International Conference on New Methods in Language Processing*, 44-49.

STEINBERGER, J., JEZEK, K. (2004). Text summarization and singular value decomposition. *Lecture Notes in Computer Science* 2457, 245–254.

WADE-STEIN, D., KINTSCH, E. (2004). Summary Street: Interactive computer support for writing. *Cognition and Instruction* 22, 333-362.

Couplage d'espaces sémantiques et de graphes pour le Deft 2011 : une approche automatique non supervisée

Yann Vigile Hoareau Murat Ahat Coralie Petermann Marc Bui
CHArt, 41 rue Gay Lussac, 75005 Paris
hoareau@lutin-userlab.fr, murat.ahat@etu.ephe.sorbonne.fr, coralie.peterman@laisc.net, marc.bui@ephe.sorbonne.fr

Résumé. Nous décrivons l'approche mise en oeuvre dans le cadre du Défi de Fouille de Texte 2011 pour la piste 2 qui consistait à identifier, pour un article scientifique donné, le résumé qui lui correspond parmi un ensemble de résumés possibles. Cette approche est basée sur le couplage entre les méthodes d'espaces sémantiques pour la représentation des connaissances sémantiques d'une part, et les graphes pour la décision sur l'affectation d'un résumé à un article, d'autre part. La méthode proposée est entièrement automatique, sans phase de paramétrage, non-supervisée et ne nécessite aucune ressource externe.

Abstract. We describe our approach in Deft 2011 for track 2, which is to identify a corresponding summary, for a given scientific paper, from a set of possible abstracts. The approach is based on coupling the methods on the one hand, of semantic space for the representation of semantic knowledge, and, on the other hand, graphs for the decision on the allocation of a resume to a document. The proposed method is fully automatic, without any particular tuning, unsupervised and requires no external resources.

Mots-clés : Espace sémantique, Graphe, Random Indexing.

Keywords: Semantic Space, Graph, Random Indexing.

Introduction

Dans cette édition du Defi de Fouille de Texte 2011, nous avons appliqué une approche originale qui consiste à mixer deux méthodes de représentation des connaissances : les espaces sémantiques qui sont des espaces vectoriels à grandes dimensions et les modèles de graphes. L'intérêt du couplage des deux approches est de bénéficier d'une part des propriétés d'apprentissage non-supervisé ainsi que des propriétés sémantiques latentes associés aux espaces sémantiques et, d'autre part de la sophistication des mathématiques sous-jacentes à la théorie des graphes. Pour ce faire, la première contrainte à respecter est de produire un graphe ayant les mêmes propriétés que l'espace sémantique en ce qui concerne la représentation des relations sémantiques latentes entre les mots ou les documents (Louwerse *et al.*, 2006). Cette contrainte satisfaite, des applications peuvent alors être réalisées directement à partir du graphe. Un exemple d'application de cette approche mixte est celui de la visualisation des relations sémantiques latentes entre documents au sein de grandes bases de données textuelles (Hoareau *et al.*, 2011).

Dans la suite de cette article, nous décrivons comment nous avons appliqué cette méthode mixte pour la tâche 2 du Deft 2011. Cette méthode a été instanciée de telle sorte à représenter la relation sémantique entre chaque résumé et chaque article dans un graphe construit à partir d'un espace sémantique, puis à transformer ce graphe complet en un graphe biparti dans lequel chaque article est associé à un et un seul résumé.

L'article est organisé de la façon suivante. Dans la première section nous décrivons les espaces sémantiques et ainsi que l'approche qui consiste à représenter les documents sous la forme d'un graphe ayant les mêmes propriétés que l'espace sémantique. Dans la deuxième section, nous décrivons la chaîne de traitement mise en oeuvre pour implémenter notre méthode. Dans le troisième section, nous présentons très brièvement les résultats de notre approche pour les tâches 1 et 2 de la piste 2. Enfin, nous concluons l'article en présentant les perspectives de recherche qui pourraient prolonger le présent travail.

1 Le couplage espace sémantique et graphe

1.1 Les espaces sémantiques

Les modèles de représentation vectorielle de la sémantique des mots sont une famille de modèles qui représentent la similitude sémantique entre les mots en fonction de l'environnement textuel dans lequel ces mots apparaissent. La distribution de co-occurrence de mots est rassemblée, analysée et transformée en espace sémantique dans lequel les mots ou les concepts sont représentés comme des vecteurs dans un espace vectoriel de grandes dimensions. *Latent Semantic Analysis* (LSA) (Landauer & Dumais, 1997), *Hyper-space Analog to Language* (HAL) (Lund & Burgess, 1996) et *Random Indexing* (RI) (Kanerva *et al.*, 2000) en sont quelques exemples. Ces modèles sont basés sur l'hypothèse distributionnelle de Harris (1968) qui affirme que les mots qui apparaissent dans des contextes semblables ont des significations semblables. La définition de l'unité de contexte est un sujet commun à tous ces modèles, même si sa nature dépend du modèle. Par exemple, LSA construit une matrice mot-document dans laquelle chaque cellule a_{ij} contient la fréquence d'un mot donné i dans une unité de contexte j. HAL définit une fenêtre flottante de n mots qui parcourt chaque mot du corpus, puis construit une matrice mot-mot dans laquelle chaque cellule a_{ij} contient la fréquence à laquelle un mot i se retrouve avec un mot j en fonction d'une fenêtre flottante donnée. Différentes méthodes mathématiques et statistiques permettant d'extraire la signification des concepts sont appliquées à la distribution des fréquences stockées dans la matrice mot-document ou mot-mot. Le premier objectif de ces traitements mathématiques est d'extraire la tendance centrale des variations de fréquences et d'éliminer ce qui peut être considéré comme du « bruit » provoqué par la part d'utilisation spécifique de la langue associée à chaque scripteur. LSA emploie une méthode générale de décomposition linéaire d'une matrice en composantes principales indépendantes : la décomposition de valeur singulière (SVD). Dans HAL la dimension de l'espace est réduite en maintenant un nombre restreint de composantes principales de la matrice de co-occurrence. Des représentations vectorielles sont employées pour le stockage et la manipulation de la signification de concepts. À la fin du processus, la similitude entre deux mots peut être calculée selon différentes méthodes. Classiquement, la valeur du cosinus de l'angle entre deux vecteurs correspondant à des mots ou un groupe de mots est calculée afin d'approximer leur proximité sémantique. Une autre méthode équivalente est la distance euclidienne pondérée.

1.2 Random Indexing

Random Indexing (Kanerva *et al.*, 2000) est un modèle d'espace sémantique qui a les mêmes propriétés que les modèles d'espaces sémantiques précédemment décrits comme LSA ou HAL. La différence avec ces deux modèles est que RI ne s'appuie par sur des méthodes de réduction matricielle mais sur les projections aléatoires. La méthode de construction d'un espace sémantique avec RI est la suivante :

– créer une matrice A ($d \times N$), contenant des *vecteurs-index*, où d est le nombre de documents ou de contextes correspondant au corpus et N, le nombre de dimensions ($N > 1000$) défini par l'expérimentateur. Les vecteurs-index sont creux et aléatoirement générés. Ils consistent en un petit nombre de (+1) et de (-1) et de centaines de 0 ;
– créer une matrice B ($M \times N$) contenant les *vecteurs-termes*, où M est le nombre de termes différents dans le corpus. Pour commencer la compilation de l'espace, les valeurs des cellules doivent être initialisées à 0 ;
– parcourir chaque document du corpus. Chaque fois qu'un terme τ apparaît dans un document d, il faut *accumuler* le vecteur-index correspondant au document d au vecteur-terme correspondant au terme τ.

À la fin du processus, les vecteurs-termes qui sont apparus dans des contextes (ou documents) similaires, auront accumulé des vecteurs-index similaires.

Le modèle a démontré des performances comparables (Kanerva *et al.*, 2000) et parfois même supérieures (Karlgren & Sahlgren, 2001) à celles de LSA pour le test de synonymie du TOEFL (Landauer & Dumais, 1997). RI a été aussi appliqué à la catégorisation d'opinion (Sahlgren & Cöster, 2004).

1.3 Le couplage Espace Sémantique–Graphe

Cette section décrit le processus de construction (i) d'un graphe complet représentant les propriétés sémantiques d'un espace sémantique, puis (ii) d'un graphe biparti à partir d'un graphe complet. Le procédé consiste à calculer la distance euclidienne pondérée entre chaque document de l'espace sémantique afin de construite une matrice de connexité. Cette matrice de connexité correspond alors à une représentation de l'espace sémantique sous la forme d'un graphe à N noeuds et N^2 arcs. L'intérêt de cette méthode très simple est de générer automatiquement un graphe qui a les mêmes propriétés que l'espace sémantique et de permettre ainsi d'y appliquer les méthodes issues de la théorie des graphes (Hoareau *et al.*, 2011).

L'algorithme décrit ci-après a pour objectif de construire un graphe biparti à partir du graphe complet construit à partir d'un espace sémantique. Il prend en entrée un ensemble d'articles ou un ensemble de résumés pour les représenter dans un espace sémantique. Une matrice m "article – résumé" est construite. Cette matrice contient dans chaque cellule $m_{i,j}$, la valeur de la distance euclidienne pondérée entre les vecteurs de l'article i et du résumé j. À partir de cette matrice, un graphe g est produit. Ce graphe g peut être ambigu au sens où un résumé donné peut correspondre à plusieurs articles et *vice versa*. Un processus de désambiguisation est appliqué à ce graphe complet afin de produire un graphe biparti où, à un résumé ne correspond qu'un seul article et *vice versa*. Dans le cas d'une ambiguïté, la distance entre le résumé et l'article est initialisée à 0 et le prochain résumé le plus proche de l'article est recherché. Ce processus est itéré jusqu'à obtenir un graphe biparti. Il est illustré dans la section 2.4.

```
Procedure main()
 Var
   A as Article Set;
   R as Resume Set;
   N as number of articles or resumes;
   m as Matrix Article Resume
   g as graph (article --> resume)

 Begin
   spaceSemantic = RandomIndexing(A, R)

   For (i:=1 to N)
     artVector = spaceSemantic(A[i]);
     For (j:=1 to N)
       resVector = spaceSemantic(R[j]);
```

```
      m[i,j] = cosine(artVector, resVector);
    End For; //j
  End For; //i

  g = createGraph(m);
  resolveAmbiguity(g)
End Procedure //main()



Procedure createGraph(m);
  Var
    m as Matrix Article Resume
    g as graph (article --> resume)
  Begin
    g = emptyGraph();
    For (i:= 1 to N)
      j = Max(m[i,:]
      g.add(i,j);
    End
    Return g;
  End Procedure //createGraph()


Procedure resolveAmbiguity(g)
  While ambiguityExist(g)
    Do
      For (k:=1 to N)
        If (g.degree(resNode[k]) > 1) Then
           ambSet = all articles nodes connected to resNode[k];
           For (l:=1 to size(ambSet))
             If (m[ambSet[l],k] is not maximum) Then
               m[ambSet[l],k] = 0;
               g.delete(ambSet[l],k);
               newJ = Max(m[ambSet[l],:]);
               g.delete(ambSet[l],newJ);
             End
           End For
        End If
      End For
    End While
End Procedure
```

2 La chaine de traitement

2.1 Extraction des documents vers une Base de Données

La première étape a été l'indexation de l'ensemble des documents du corpus dans une base de données. Pour cela, une base de données relationnelle a été construite afin de stocker toutes les informations fournies en gardant leur structure et leurs liens. L'idée était de pouvoir facilement accéder à l'ensemble des données et modifier les unités de contextes utilisées pour l'apprentissage de l'espace sémantique (Rehder *et al.*, 1998; Bestgen, 2004). Un ensemble de scripts en langage PHP a été développé afin de réaliser ces tâches. Le système de gestion de base de données est MySql pour sa facilité de connexion avec les divers langages utilisés (Php, Java, etc.).

2.2 Indexation de la base de données

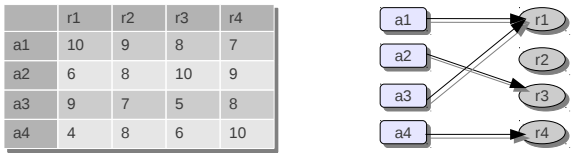
Afin d'automatiser entièrement notre chaîne de traitement, nous avons choisi d'intégrer LuSql [1] pour l'indexation du corpus. LuSql est une application java en ligne de commande permettant de construire des index Lucene à partir de bases de données relationnelles. Il permet de sélectionner précisément les données à indexer en passant en argument de la ligne de commande la requête SQL utilisée. De plus, dans son mode par défaut, il utilise le *multithreading* pour s'exécuter sur plusieurs processeurs et donc optimiser les temps d'indexation. Toujours en vue d'automatiser nos outils, nous avons développé une interface java intégrant LuSql.

2.3 Construction de l'espace sémantique

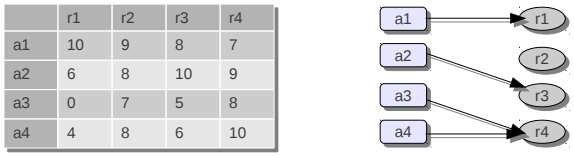
L'espace sémantique a été construit en utilisant la méthode Random Indexing implémenté par la librairie Java SemanticVectors (Widdows & Cohen, 2010). Les unités de contexte servant à l'apprentissage sont les documents tels que définis dans le corpus d'apprentissage : l'article et le résumé. Le corpus n'a fait l'objet d'aucun traitement de lemmatisation (Hoareau *et al.*, 2007; Lemaire, 2008). Les mots vides ont été supprimés. L'espace sémantique RI a été construit avec 1000 dimensions. L'ensemble des traitements a été réalisé sur un ordinateur portable à 2 Go de RAM, intel core2 duo cpu à 2.00ghz.

2.4 Construction du graphe biparti

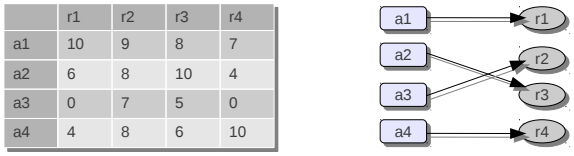
Dans cette étape, nous avons appliqué l'algorithme de la section 1.3. Le graphe construit à partir de l'espace sémantique n'avait que peu d'ambiguïtés. Nous attribuons cela d'une part à la qualité des données et d'autre part, à la robustesse de *Random Indexing*. Le processus de résolution des ambiguïtés a permis construire le graphe biparti avec chaque noeud à un degré de 1. Dans la Figure 1, nous illustrons l'ensemble du processus de la construction du graphe et de résolution des ambiguïtés.



(a) Etape 1 : Graphe avec ambiguïté



(b) Etape 2 : Graphe avec ambiguïté



(c) Etape 3 : Graphe biparti sans ambiguïté

FIGURE 1 – Exemple de construction d'un graphe biparti

1. http ://lab.cisti-icist.nrc-cnrc.gc.ca/cistilabswiki/index.php/LuSql

3 Les résultats

Les performances de la méthode pour les pistes 1 et 2 représentées dans le Tableau 1 sont très satisfaisantes. La méthode réalise un score de 1 pour la tâche 1 et de 0,995 pour la tâche 2.

	Tâche 1 (articles complets)	Tâche 2 (contenu)
Exécution	**1,000**	**0,995**
Moyenne	0,981	0,956
Médiane	0,990	0,959
Ecart-type	0,027	0,042

TABLE 1 – Scores pour les tâches 1 et 2

4 Conclusion

La méthode proposée dans le cadre de notre participation au Deft 2011 repose sur le couplage entre les espaces sémantiques et les graphes. Le faible nombre de documents disponibles pour l'apprentissage constituait une contrainte forte pour notre méthode entièrement basée sur une approche distributionnelle. Les résultats indiquent que la méthode n'a pas souffert de cette contrainte.

Les méthodes d'espaces sémantiques telles que LSA, utilisant des techniques de réductions matricielles nécessitent un nombre important de documents lors de l'apprentissage afin que les processus de réduction puissent s'appliquer de façon efficace. Ainsi, il nous semble fort probable que la robustesse de notre approche serait la conséquence de l'utilisation de *Random Indexing* qui est basé sur les projections aléatoires.

De prochaines expériences seront réaliser afn de tester cette hypothèse car ses implications pourraient s'avérer importantes pour la recherche appliquée : il serait alors possible de tirer profit des propriétés des espaces sémantiques à partir d'un corpus même très limité en nombre.

Références

BESTGEN Y. (2004). Analyse sémantique latente et segmentation automatique des textes. In G. PURNELLE, C. FAIRON & A. DISTER, Eds., *Proceedings of the 9th International Conference on the Statistical Analysis of Textual Data(JADT'08)*, volume 1 of *Cahier du Cental*, p. 171–181, Louvain : Presse Universitaire de Louvain.

HARRIS Z. (1968). *Mathematical Structures of Language*. New York : John Wiley and Son.

HOAREAU Y., GANDON F., GIBOIN A., DENHIÈRE G., JHEAN-LAROSE S., LENHARD W. & BAIER H. (2007). Similarity measurement applied to information research and indexing. In F. WILD, M. KALZ, J. VAN BRUGGEN & R. KOPER, Eds., *Proceedings of the First European Workshop on Latent Semantic Analysis in Technology Enhanced Learning*, p. 5–6, Herleen (Holand).

HOAREAU Y. V., AHAT M., MEDERNACH D. & BUI M. (2011). Un outil de navigation dans un espace sémantique. In A. KHENCHAF & P. PONCELET, Eds., *Extraction et gestion des connaissances (EGC'2011)*, volume RNTI-E-20 of *Revue des Nouvelles Technologies de l'Information*, p. 275–278 : Hermann-Éditions.

KANERVA P., KRISTOFERSON J. & HOLST A. (2000). Random Indexing of Text Samples for Latent Semantic Analysis. In L. GLEITMAN & A. JOSH, Eds., *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, Mahwah : Lawrence Erlbaum Associates.

KARLGREN J. & SAHLGREN M. (2001). From Words to Understanding. In Y. UESAKA, P. KANERVA & H. ASOH, Eds., *Foundations of Real-World Intelligence*. Stanford : CSLI Publications.

LANDAUER T. & DUMAIS S. (1997). A Solution to Plato's Problem : The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge. *Psychological Review*, **104**(2), 211–240.

LEMAIRE B. (2008). Limites de la lemmatisation pour léxtraction de significations. In G. PURNELLE, C. FAIRON & A. DISTER, Eds., *Proceedings of the 9th International Conference on the Statistical Analysis of Textual Data(JADT'08)*, Cahier du Cental, p. 725–732, Louvain : Presse Universitaire de Louvain.

LOUWERSE M., CAI Z., HU X., VENTURA M. & JEUNIAUX P. (2006). Cognitively inspired natural-language based knowledge representations : Further explorations of latent semantic analysis. *International Journal of Artificial Intelligence Tools*, **15**, 1021–1039.

LUND K. & BURGESS C. (1996). Producing high-dimensional semantic space from lexical co-occurence. *Behavior research methods, instruments & computers*, **28**(2), 203–208.

REHDER B., SCHREINER M., WOLFE M., LAHAM D., LANDAUER T. & KINTSCH W. (1998). Using Latent Semantic Analysis to assess knowledge : Some technical considerations. *Discourse Processes*, **25**(2), 337–354.

SAHLGREN M. & CÖSTER R. (2004). Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *COLING '04 : Proceedings of the 20th international conference on Computational Linguistics*, p. 487, Morristown, NJ, USA : Association for Computational Linguistics.

WIDDOWS D. & COHEN T. (2010). The semantic vectors package : New algorithms and public tools for distributional semantics. *International Conference on Semantic Computing*, **0**, 9–15.

One simple formula for losing DEFT with more than 90% of correct guesses

Daniel Devatman Hromada

(1) PhD. student Université Paris 8 – St. Denis
hromi@kyberia.sk

Abstract

A simple scoring method based upon unweighted summing of relative word occurrence probabilities was implemented in order to find the solution to the second problem of data-mining competition "Défi Fouille de Textes". The objective - to couple, one by one, the limited set of scientific articles with the limited set of abstracts summarizing these articles – was attained in two computational passes with >97% hit rate for the training corpus only, >90% for the testing corpus only, >94% for the testing+training corpora combined and 92.9% for the testing corpus exploiting the knowledge acquired from training corpus. These results indicate that relative frequencies of individual words yield useful "features" for coupling the full text with its summarized counterpart.

Résumé

Une méthode simple, basée sur l'addition des probabilités d'occurence des mots, fut mise en place afin de proposer la solution à la deuxième tâche de Défi Fouille de Textes 2011. Cette tâche consistait à identifier à quel article scientifique – parmi un ensemble fermé d'articles - correspondait un résumé appartenant à un ensemble fermé de résumés. La formule que nous proposons atteignit >97% de taux de réussite quand évaluée sur le corpus d'apprentissage seulement, 90% quand évaluée sur le corpus de test seulement, >94% quand évaluée sur les deux corpora combinés, et 92.9% pour l'évaluation du corpus de test prenant en compte les distributions de probabilité obtenus à partir du corpus d'apprentissage. Ces résultats indiquent que les fréquences relatives des mots individuels peuvent être considerées comme les "traits" pertinents pour l'appariement des articles avec leurs résumés.

Mots-clés: appariement résumé / article, fréquence relative, mot vide, hapax
Keywords: full-text / abstract coupling, relative frequency, stopword, hapax

1 Introduction

The goal of the second task of challenge Défi Fouille de Textes (DEFT) 2011 was to couple N scientific articles with N abstracts summarizing the contents of the respective articles, where N=300 for the training corpus and N=200 for the testing corpus. Due to the lack of resources, our team had decided to try to ignore more state of the art approaches based on Artificial Neural Networks, Support Vector Machines, Nearest Neighbors or boosting and decided to confront the DEFT corpora with a very simple, yet intuitively[1] appealing idea.

The basic idea is simple: since abstract is nothing else than the condensation of the full-text, it can be stated that if the word W occurs in the abstract, it will tend to occur in the full-text as well; and more it is present in the abstract, more would one expect to find it in the associated full-text. Since opposite is also the case - i.e. the more W is present in the full-text F , the more one would tend to associate with F the abstract A within which W is frequent - one can reason even

[1] The reason why there are no references in this article is that the method hereby presented does not follow any antecedent study but was based upon pure intuition.

further: if W is frequent in abstract A and full-text F_1 but not frequent in full-text F_2, one would tend to couple A with F_1 and not with F_2.

We have supposed that such "surface information" as relative word frequencies of diverse words are the only "features" needed in order to obtain the list of most plausible [abstract, full-text] candidate couples.

2 Method

The method hereby proposed is based upon very simple frequency counting which occurs in two passes. In the first, so-called "article pass", the total frequency $F_{W,total}$ for every word W in all articles, as well as $F_{W,A}$ denoting a number of occurences of the word W in article A are calculated. Therefore, after this first pass, our algorithm can calculate the relative frequency:

$$P_{W,A} = F_{W,A} / F_{W,total}$$

i.e. "relative probability of word W occuring in article A when compared with the rest of the corpus". It is evident that the highest $P_{W,A}$ =1 will be obtained in case of hapaxes (i.e. $F_{W,A} = 1$; $F_{W,total} = 1$; $P_{W,A} = 1 / 1 = 1$) and "relative hapaxes" (i.e. the words like proper nouns which occur in one article only; $F_{W,A} = N$; $F_{W,total} = N$; $P_{W,A} = N / N = 1$).

In the second, so-called "abstract pass", the algorithm calculates the overall score for every possible (abstract, article) couple by taking, one after another, every word W from every abstract, and, **adding** the to the $P_{W,A}$ value for every possible article A. Thus all the mathematical operations of our method are contained within this line of simple PERL code:

```
$abstract_article{$abstract}{$article}+= ($word_freq_in_article{$word}{$article} / ($word_total{$word})) if $word_total{$word};
```

In other words, the final score for a possible [abstract, article] couple is obtained as **a sum of all $P_{W,A}$** where W is every word present in the abstract-being-scored, and A is any article whatseover. Finally, [abstract, article] couples are sorted in descending order and every abstract is coupled with the article from the top of the list, i.e. having the highest score.

3 Results

The overall results of our tentatives are presented in the Table 1. After having being motivated by encouraging results (>97%) obtained by application of our method upon the training corpus (N=300) , we have applied the same method upon the testing corpus (N=200). While the test yielded 90% hit rate, it earned us the last place in DEFT competition.

Training	**Testing**	**Hit rate – with stopwords**	**Hit rate – without stopwords**
N=300	N=300	292 (97.3%)	293 (97.7%)
N=200	N=200	180 (90%)	**194 (97%)**
N=300+200	N=300+200	471 (94.2%)	469 (93.8%)
N=300+200	N=200	185 (92.5%)	184 (92%)

Table 1 : Obtained results for different combinations of testing & training corpora

Unfortunately, it was only after the announcement of the oficial DEFT results that we have found time to vary our method. Firstly, one can notice (c.f. row 4 of Table 1) that the keeping of the total word frequencies learned from the training corpus can increase the efficacity of our additive scoring when applied upon testing corpus. Secondly, by implementing a list of stopwords from CPAN's Lingua ::StopWords package, we raised the hit rate of our simple formula to 97% which we consider to be a satisfying result, given the simplicity of our approach.

4 Discussion

The theoretical framework of our approach can be characterized as follows: the summarization process during which an author condenses the knowledge contained in the full text characterized by a word-frequency histogram F into much shorter abstract characterized by a word-frequency histogram A can be understood as a surjection of F onto A. In other terms, there exists a mapping function between F and A and the approximative knowledge of this function could allow us to find & evaluate the best candidate (F, A) couples.

Our naïve supposition stating that a very simple unweighted addition of relative probabilities of all the words present in the abstract could be considered as a sufficiently adequate approximation of such a mapping function, allowed us to obtain more than 90% of correct couplings within the scope of the test corpus, nonetheless our team ended up as last in this task of the DEFT competition for which 3 teams have attained 100% hit rate.

In spite of being last in the DEFT competition, we think that our proposal have certain properties which make it worth of interest. Firstly, the method proposed hereby is fully deterministic, no stochastic or quasistochastic element is involved in the scoring nor in the subsequent [article, abstract] coupling. Secondly, diversification of results after exclusion of "stopwords" indicates that the method hereby proposed possibly disposes of various parameters which could be possibly tuned in order to obtain 100% accuracy. Thirdly, the fact that no external corpus was needed in order to obtain the results which were obtained indicates that at least for a certain class of text-summarization problems, one is not obliged to implement "state of the art" machine learning techniques in order to construct robust statistical models of semantic spaces – , but can stick to more empiric "surface features" like relative word frequencies. It is evident that the calculation of such surface features demands less computational resources than more sophisticated techniques.

Lastly, we think that the method, *the idea* proposed hereby could be successfully applied not only upon corpora written in french language but could be used to couple abstracts and articles written in any non-inflectional language like English, for example. We think that this is possible because, as our results indicate, there exist certain quantitative correlations, certain observable morphisms, between distribution of symbols in full-text and distribution of symbols in related abstract.

5 The code

```
#articles are in « art » directory, abstracts are in « res » directory
print '<?xml version="1.0" encoding="utf-8" ?>'."\n<corpus>\n";
#1st pass - creating total & article-relative word frequency histograms for all articles
my %word_freq_in_article;
my %word_freq_in_all_articles;
@artz=glob("art/*.pur");
for $art (@artz) {
 $art=~/^art\/(\d\d\d)/;
 $file=$1;
 open(A,$art);
 while (<A>) {
  @wordz=split(/[^\w]/);
  for $word (@wordz) {
   if (!$word_freq_in_all_articles{$word}) {
    $word_freq_in_all_articles{$word}=1;
    $word_freq_in_article{$word}{$file}=1;
   }
   elsif (!$word_freq_in_article{$word}{$file}) {
    $word_freq_in_all_articles{$word}++;
    $word_freq_in_article{$word}{$file}=1;
   }
   else {
    $word_freq_in_all_articles{$word}++;
    $word_freq_in_article{$word}{$file}++;
   }
  } } }
```

```
#2nd pass – we take every word W from every abstract and then look at the frequencies of W in all articles
my @keylist;
my %abstract_article;
foreach $f (<res/*.res>) { $i{$f} = -s $f };
@re_filez = (sort{ $i{$b} <=> $i{$a} } keys %i);

for $resfile (@re_filez) {
  $resfile=~/^res\/(\d\d\d)/;
  $abstract=$1;
  push @keylist, $abstract;
  open(F,$resfile);
  while (<F>) {
   if (/<p>(.*?)<\/p>/) {
    $content=$1;
    @wordz=split(/[^\w]/,$content);
    for $word (@wordz) {
     for $article (keys%{$word_freq_in_article{$word}}) {
      $abstract_article{$abstract}{$article}=0 if (!$abstract_article{$abstract}{$article});
      #formula which attributes the score to every (abstract, article) couple
      $abstract_article{$abstract}{$article}+= ($word_freq_in_article{$word}{$article} /
($word_freq_in_all_articles{$word})) if $word_freq_in_article{$word}{$article};
     }
    }
   }
  }
}
 our @used;
our @keyz;
sub r {
 $depth=$_[0];
 if (grep($_ eq $keyz[$depth], @used)) {
  r($depth+1);
 } else {
  return $keyz[$depth];
 }
}

for $abstract (@keylist) {
 %abhash=%{$abstract_article{$abstract}};
#descendant ordering of (abstract, article) couples gives us  the best candidates
 @keyz = sort {$abhash{$b} <=> $abhash{$a}} (keys(%abhash));
 $key=r(0);
 if ($abhash{$keyz[0]}>($abhash{$keyz[1]}+0.23)) {
  push @used,$key;
 }
 print "<doc><resume fichier=\"$abstract.res\" /><article fichier=\"$key.art\" /></doc>\n";
 $hit++ if ($resultz{$abstract}==$key);
} print "</corpus>\n";
```

Index des auteurs

Index des mots-clés