Participation du LINA à DEFT 2012

Florian Boudin Amir Hazem Nicolas Hernandez Prajol Shrestha
Université de Nantes
prénom.nom@univ-nantes.fr

RÉSUMÉ
Cet article présente la participation de l'équipe TALN du LINA au défi fouille de textes (DEFT) 2012. Développé spécifiquement pour la seconde piste du défi, notre système combine les sorties de trois différentes méthodes d'extraction de mots clés. Notre système s'est classé à la $2^{ième}$ place sur un total de 9 systèmes avec une f-mesure de 21,3%.

ABSTRACT
LINA at DEFT 2012

This article presents the participation of the TALN group at LINA to the défi fouille de textes (DEFT) 2012. Developed specifically for the second task, our system combines the outputs of three different keyword extraction methods. Our system ranked 2^{nd} out of 9 systems with a f-measure of 21,3%.

MOTS-CLÉS : extraction de mots clés, deft 2012, combinaison de méthodes.

KEYWORDS: keyword extraction, deft 2012, combining methods.

1 Introduction

L'indexation automatique consiste à identifier un ensemble de mots clés (e.g. mots, termes) qui décrit le contenu d'un document. Les mots clés peuvent ensuite être utilisés, entre autres, pour faciliter la recherche d'information ou la navigation dans les collections de documents. L'édition 2012 du défi fouille de textes (DEFT) porte sur l'extraction automatique de mots clés à partir d'articles scientifiques parus dans le domaine des Sciences Humaines et Sociales (SHS).

L'objectif du défi est de retrouver, à partir du contenu des documents (i.e. articles scientifiques), les mots clés qui ont pu être choisis par les auteurs. Deux différentes pistes ont été proposées. La première piste consiste à identifier dans une terminologie, les mots clés qui ont été assignés aux documents. Cette terminologie regroupe l'ensemble des mots clés utilisés dans la collection. La seconde piste, de prime abord plus complexe, consiste à extraire les mots clés directement à partir du contenu des documents. Cet article décrit notre participation à la seconde piste du défi.

Le reste de cet article est organisé comme suit. La section 2 décrit l'ensemble de données utilisé pour la campagne d'évaluation. La section 3 présente les différentes méthodes que nous avons développées spécifiquement pour la seconde piste du défi. Nous décrivons ensuite en section 4 nos résultats expérimentaux avant de présenter les méthodes que nous avons testées et qui

JEP-TALN-RECITAL 2012, Atelier DEFT 2012: DÉfi Fouille de Textes, pages 61–68,
Grenoble, 4 au 8 juin 2012.

qui ont eu un impact nul ou négatif sur les résultats. La section 6 conclut cet article et donne quelques perspectives de travaux futurs.

2 Description de la campagne DEFT 2012

L'ensemble de documents utilisé pour le défi 2012 est constitué de 234 articles scientifiques parus dans le domaine des SHS. Ces articles ont été publiés entre 2001 et 2008 dans quatre revues différentes. L'ensemble d'apprentissage contient 60% des documents (soit 141 articles), et celui de test contient les 40% restants (soit 93 articles). La répartition des quatre différentes revues dans les deux ensembles est uniforme.

Du point de vue technique, les articles sont au format XML. Ils sont structurés en deux parties : le résumé et le corps de l'article. Chaque article contient également le nombre de mots clés indexant son contenu. Les mots clés assignés à chaque article sont disponibles pour chacun des articles de l'ensemble d'entraînement.

Les systèmes participant au défi sont évalués à l'aide des mesures classiques de précision, rappel et f-mesure. Pour chaque article, les mots clés générés par les systèmes sont comparés aux mots clés de référence (assignés par les auteurs). Afin de limiter les problèmes liés aux différentes variations orthographiques, plusieurs traitements de normalisation (i.e. normalisation de la casse et lemmatisation) sont appliqués au préalable aux mots clés. Chaque participant peut soumettre jusqu'à trois exécutions par piste.

La liste ci-dessous présente quelques unes des difficultés que nous avons identifiées dans les articles de l'ensemble d'entraînement.

– Articles différents ayant le même résumé, e.g. les articles `as_2002_007048ar` et `as_2002_007053ar`.
– Contenu des articles dans des langues différentes et/ou mélangées, e.g. français et anglais dans `ttr_2008_037494ar`, espagnol dans `meta_2005_019927ar`.
– Contenu des articles très bruité avec des problèmes de ponctuation, de caractères unicodes et de segmentation en paragraphes, e.g. ci-dessous un extrait de l'article `meta_2005_019840ar`.

```
<p>Ce langage est au c.ur des préoccupations des juristes, qui
nous rappellent régulièrement</p>
<p>que le droit est affaire de mots. Et cela dans tout l.univers
du droit, vers quelque côté que l.on se</p>
<p>tourne, dans le monde juridique anglophone - où, pour
Mellinkoff (1963&amp;#x00A0;: vii), «&amp;#x00A0;The law is a</p>
<p></p>
<p>dans son ensemble, la technique juridique aboutit, pour la
plus grande part, à une question de</p>
<p>terminologie&amp;#x00A0;». Chacun pourra le vérifier par la
consultation d.ouvrages parmi les plus récents et</p>
```

3 Approches

Les différentes méthodes que nous avons développées utilisent le mot comme unité principale. Nous avons donc appliqué un ensemble commun de pré-traitements aux documents : segmentation en phrases, découpage en mots et étiquetage morpho-syntaxique. L'information structurelle présente dans chacun des documents (i.e. résumé, corps de l'article et paragraphes) est préservée. Chaque paragraphe est segmenté en phrases en utilisant la méthode PUNKT de détection de changement de phrases (Kiss et Strunk, 2006) mise en œuvre dans la boîte à outils NLTK (Bird et Loper, 2004). La tokenisation des phrases est effectuée avec un outil développé en interne utilisant le lexique des formes fléchies du français (lefff)[1] pour l'indentification des unités lexicales complexes (e.g. mots composés). L'étiquetage morpho-syntaxique est obtenu à l'aide du *Stanford POS Tagger* (Toutanova *et al.*, 2003)[2] entrainée sur le *French Treebank* (Abeillé *et al.*, 2003).

3.1 Système 1

Ce système est basé sur du $TF \times IDF$ et trois règles issues du corpus d'apprentissage. La principale question qui se pose ici est : qu'est ce qu'un mot clé ? ou autrement dit, qu'est ce qui fait qu'un terme a plus de chances d'être un mot clé qu'un autre ?

En analysant les documents du corpus d'apprentissage, nous avons relevé trois particularités liées aux mots clés. La première concerne leur localisation dans les documents. Chaque document étant divisé en deux parties qui sont : le résumé (ABSTRACT) et le corps du document (BODY), nous nous sommes donc intéressés à la position des mots clés par rapport à ce découpage. Nous avons pu constater qu'un terme apparaissant à la fois dans le résumé et dans le corps du document avait plus de chances d'être un mot clé. Ainsi, nous avons utilisé cette information comme première règle de notre système (nous appellerons cette règle : R1). Deux stratégies utilisant cette règle ont été adoptées, la première consiste à ne sélectionner que des termes qui apparaissent à la fois dans le résumé et dans le corps du document (nous appellerons cette stratégie : S1), la deuxième consiste à donner la priorité aux termes respectant la stratégie S1 en utilisant une pondération par un paramètre α fixé empiriquement (nous appellerons cette stratégie : S2). Les différents tests conduits ont montré que l'utilisation de la stratégie S1 donnait de meilleurs résultats que l'utilisation de la stratégie S2. Intuitivement, nous aurions tendance à penser le contraire (la stratégie S2 devrait être meilleur que S1), car éliminer des termes n'apparaissant que dans le résumé ou que dans le corps du document nous ferait sans doute perdre des mots clés. L'explication et que la stratégie S1 corrige sans doute les faiblesses de notre système qui renverrait plus de faux positifs que de vrais négatifs.

La deuxième particularité relative aux n-grammes, découle de la question suivante : est ce qu'un terme simple (1-gramme) a plus de chances d'être un mot clé qu'un terme composé (n-grammes avec $n > 1$) ? De part le corpus d'apprentissage nous avons pu constater qu'il y avait 70% de termes simples et 30% de termes composés. Ainsi, nous avons voulu donner une plus grande importance aux termes simples extraits par notre système. De la même manière que pour la stratégie S2, nous avons introduit un paramètre de pondération β afin de prioriser les termes simples (nous appellerons cette règle R2).

[1] `http://www.labri.fr/perso/clement/lefff/`
[2] Nous utilison la version 3.1.0 avec les paramètres par défaut.

La troisième particularité relève de la simple observation que la quasi totalité des mots clés étaient soit des noms, soit des adjectifs. À partir de cette constatation, nous avons introduit une troisième règle (R3) qui filtre les verbes.

3.2 Système 2

Ce système repose sur l'exploitation d'un existant à savoir l'approche KEA (*Keyphrase Extraction Algorithm*) de (Witten *et al.*, 1999). KEA permet d'une part de modéliser les expressions significatives (composées d'un ou plusieurs mots) du contenu de textes à l'aide de textes et d'expressions clés associées et d'autre part d'extraire les expressions clés d'une collection de textes à l'aide d'une modélisation construite *a priori*. L'approche utilise un classifieur bayésien naïf pour calculer un score de probabilité de chaque expression clé candidate. La construction requiert un ensemble d'expressions clés classées positivement pour chaque texte du corpus d'apprentissage. L'extraction se réalise sur un corpus de domaine similaire au domaine du corpus d'apprentissage.

Les phases de modélisation ou d'extraction des expressions clés fonctionnent toutes deux à la suite de deux phases élémentaires : l'extraction de candidats et le calcul de traits descriptifs des candidats. Les candidats s'obtiennent par extraction de *n*-grammes de taille prédéfinie ne débutant pas et ne finissant pas par un mot outil.

Les traits utilisés pour décrire chaque candidat au sein d'un document sont les suivants : le $TF \times IDF$ (mesure de spécificité du candidat pour le document), la position de la première occurrence (pourcentage du texte précédent l'occurrence), le nombre de mots qui compose le candidat. Les candidats ayant un haut $TF \times IDF$, apparaissant au début d'un texte et comptant le plus de mots sont ainsi considérés comme étant de bons descripteurs du contenu d'un texte.

Les dernières évolutions de KEA permettent d'exploiter des lexiques contrôlés de type thésaurus dans la construction de la modélisation (Medelyan et Witten, 2006).

Nous n'avons pas exploité de ressources extérieures de type lexiques contrôlés dans la construction de notre modélisation. Nous avons utilisé la version 5.0 de l'implémentation de KEA[3] disponible sous licence GNU ; en pratique nous avons utilisé les fonctionnalités d'extraction «libre» présentes dès la version 3.0. Les fonctionnalités développées ultérieurement concernent l'exploitation de lexiques contrôlés. Nos candidats étaient au maximum de taille 5. Nous avons exploité le corpus d'apprentissage fourni pour la seconde piste pour construire notre modélisation. Chaque texte (résumé et corps) a été considéré comme une unité documentaire.

L'approche KEA est facilement portable à différentes langues du fait qu'elle nécessite peu de ressources. En particulier elle ne requiert pas une pré-analyse syntaxique pour sélectionner des candidats. Nous avons néanmoins porté une certaine attention à nos traitements préliminaires et nous avons constaté qu'une pré-segmentation en token mots ainsi que l'utilisation d'une liste multilingue de mots outils augmentaient la qualité de l'extraction des expressions clés lorsque nous évaluions l'approche par validation croisée sur le corpus d'apprentissage. Concernant la liste des mots outils, nous avons fusionné les listes fournies par KEA pour le français, l'anglais et l'espagnol. Nous l'avons complétée des formes des mots outils pouvant subir une élision du *e* final en français (e.g. «*le*» s'est vu complété de la forme «*l'*», de même pour «*lorsque*» avec «*lorsqu'*»...). Ces formes étaient en effet reconnues par notre segmenteur en mots.

[3] `http://www.nzdl.org/Kea`

3.3 Système 3

Ce système est basé sur une approche par classification supervisée. La tâche d'extraction de mots clés est ici considérée comme une tâche de classification binaire. La première étape consiste à générer tous les mots clés candidats à partir du document. Pour ce faire, nous commençons par extraire tous les n-grammes de mots jusqu'à $n = 4$. Des contraintes syntaxiques sont ensuite utilisées pour filtrer les candidats. Ainsi, seuls les n-grammes composés uniquement de noms, d'adjectifs et de mots outils (excepté en premier/dernier mot du n-gramme) sont gardés.

Pour chaque candidat, nous calculons les traits suivants :

– Poids $TF \times IDF$
– Nombre de mots du n-gramme
– Patron syntaxique du n-gramme (e.g. "Nom Adjectif")
– Position relative de la première occurence dans le document
– Section(s) où apparait le n-gramme (résumé, corps ou les deux)
– Nombre de documents de la collection dans lesquels le n-gramme apparait
– Score de saillance dans l'arbre de dépendances de cohésion lexicale du texte (voir ci-dessous)

Nous construisons ce que nous appelons un «arbre de dépendances de cohésion lexicale» selon une approche décrite par Choi à la section 6.3.1. de sa thèse (Choi, 2002). Une dépendance est présupposée exister entre deux phrases consécutives si celles-ci ont des mots en commun ; l'hypothèse est de considérer la seconde phrase comme une élaboration de la première. En pratique, notre algorithme ne reconnaît pas systématiquement une relation de dépendance entre deux phrases consécutives qui partagent des mots en commun. En effet notre algorithme recherche, pour chaque phrase du texte, la phrase la plus haute dans la chaîne de dépendance de la phrase précédente avec laquelle elle partage des mots en commun. L'arbre est construit en prenant le texte dans son ensemble (résumé et corps) préalablement lemmatisé. Un score de saillance est calculé pour chaque phrase en fonction du nombre de ses dépendances (directes et transitives) normalisé par le nombre de dépendances maximal qu'une phrase peut avoir sur le texte donné. Chaque expression candidate hérite alors du score de la phrase où apparait sa première occurence.

Nous utilisons la combinaison par vote de trois algorithmes de classification disponibles dans la boite à outils Weka (Hall *et al.*, 2009) : NaiveBayes, J48 et RandomForest. Les mots-clés candidats sont ensuite triés selon leurs scores de prédiction.

3.4 Combinaison des systèmes

Les trois systèmes que nous avons développés utilisent différentes méthodes pour capturer l'importance d'un mot clé par rapport à un document. Une combinaison des sorties de ces derniers est donc pertinente.

Nous disposons pour chaque document, de trois listes pondérées de mots clés. La méthode la plus simple consisterait à utiliser la somme des scores des trois systèmes. Cependant, les scores calculés par chacun des systèmes ne sont pas directement comparables. À la place du score, nous utilisons pour chaque mot clé candidat, l'inverse de son rang dans la liste ordonnée.

Deux stratégies de combinaison ont été utilisées. La première, COMBI1 consiste à assigner la

somme de l'inverse des rangs d'un mot clé dans les listes ordonnées des trois systèmes. Pour la seconde stratégie, COMBI2, nous ne considérons que les mots clés apparaissant dans les sorties des trois systèmes. L'idée est de filtrer les mots clés considérés comme important par seulement un ou deux des trois systèmes.

4 Résultats

Nous présentons dans cette section les résultats officiels de la campagne DEFT 2012. Nous avons soumis trois exécutions pour chacune des deux pistes. Pour la première piste, nous avons simplement utilisé le Système 1 (décrit dans la section 3.1) et filtré les mots clés candidats à l'aide de la terminologie. Le nombre de mots clés retournés est fixé à 7 pour la première exécution et à 6 pour les deux autres. Les trois configurations utilisent la règle R3.

La première exécution utilise la règle R2 ($\beta = 0,6$). La seconde exécution utilise la règle R2 ($\beta = 0,65$). La troisième exécution utilise la règle R1 avec la stratégie S2 ($\alpha = 0,65$) et la règle R2 ($\beta = 0,65$). Pour la seconde piste, nous avons soumis les exécutions de deux combinaisons (COMBI1 et COMBI2) ainsi que du système 3 (décrit dans la section 3.3). Le nombre de mots clés retournés est fixé à 130% du nombre de mots clés de référence pour COMBI1 et COMBI2 et à 110% pour le système 3. Ces nombres permettent d'obtenir les meilleurs résultats sur l'ensemble d'entraînement.

La table 1 présente les résultats de nos trois exécutions pour la première piste. Les résultats obtenus par les trois exécutions sont moins bons que ceux obtenus sur l'ensemble d'entrainement (f-mesure=0,44 pour la première exécution). Nous constatons que la variation du rappel sur les trois exécutions est faible. La chute de la précision pour la troisième exécution s'explique par l'application de la règle R1 qui limite le nombre de candidats possibles.

Système	Précision	Rappel	f-mesure
1	0,3812	0,4004	**0,3906**
2	0,3759	0,3948	0,3851
3	0,3343	0,4097	0,3682

TAB. 1 – Résultats de nos trois exécutions pour la première piste.

La table 2 montre les résultats de nos trois exécutions pour la seconde piste. Nous pouvons voir que la performance de COMBI2 est largement en dessous de COMBI1. Nous avions constaté le phénomène inverse sur les données d'entraînement. Ceci est du au fait que le nombre de mots clés retournés par COMBI2 peut dans certains cas être inférieur au seuil que nous avons fixé. En effet, l'intersection des listes des 100 meilleurs mots clés candidats de chaque système est très restrainte pour quelque uns des documents de l'ensemble de test. Nous constatons que les scores du système 3, ayant obtenu les meilleurs résultats sur l'ensemble d'entraînement parmis nos trois systèmes, sont faibles en comparaison des deux combinaisons. Ce résultat semble indiquer un problème de sur-entraînement et illustre bien l'utilité de la combinaison.

La table 3 présente, pour chacune des deux pistes, le classement des différentes équipes sur la base de la meilleure soumission. Notre soumission est classée au rang 5 sur 10 pour la première

Système	Précision	Rappel	f-mesure
COMBI1	0,1949	0,2355	**0,2133**
COMBI2	0,1788	0,2128	0,1943
Système 3	0,1643	0,1880	0,1753

TAB. 2 – Résultats de nos trois exécutions pour la seconde piste.

piste et au rang 2 sur 9 pour la seconde piste. Les résultats obtenus par l'équipe 16 sont bien au dessus de toutes les autres équipes et montrent qu'une marge de progression importante est possible pour notre système.

Rang	Piste 1	Piste 2
1	Équipe 16 (0,9488)	Équipe 16 (0,5874)
2	Équipe 05 (0,7475)	**Équipe 06 (0,2133)**
3	Équipe 04 (0,4417)	Équipe 05 (0,2087)
4	Équipe 02 (0,3985)	Équipe 02 (0,1921)
5	**Équipe 06 (0,3906)**	Équipe 01 (0,1901)
6	Équipe 01 (0,2737)	Équipe 13 (0,1632)
7	Équipe 13 (0,1378)	Équipe 04 (0,1270)
8	Équipe 17 (0,1079)	Équipe 17 (0,0895)
9	Équipe 03 (0,0857)	Équipe 03 (0,0785)
10	Équipe 18 (0,0428)	-

TAB. 3 – Classement de DEFT 2012 sur la base de la meilleure soumission de chaque équipe pour chacune des deux pistes. Notre classement est indiqué en gras (équipe 06).

5 Ce qui n'a pas marché

Nous décrivons ici les méthodes qui ont eu un impact nul ou négatif sur les résultats.

Traits ayant un impact négatif sur la performance du système 3 : la dispersion d'un mot clé dans le document, mots appartenant à des phrases contenant des citations, noms des auteurs les plus cités dans le document (spécifique aux articles commençant par "as").

Suppression de la redondance : nous avons constaté un niveau de redondance important des mots clés dans les sorties de nos systèmes. Par exemple, les mots clés "jardins collectifs", "jardins" et "collectifs" sont tous les trois présents dans le top 10, ce qui fait baisser le rappel. Plusieurs stratégies ont été expérimentées pour supprimer cette redondance (e.g. suppression d'un n-gramme si tous les mots qui le composent sont également présents parmi les 10 meilleurs candidats). Une dégradation des résultats est cependant observée indiquant que la stratégie à adopter est dépendante des documents.

Modèle de pondération à base de graphe : nous avons implémenté l'approche proposée dans (Mihalcea et Tarau, 2004). Il s'agit de représenter chaque document sous la forme d'un

graphe de mots connectés par des relations de co-occurrences. Des algorithmes de centralité sont ensuite appliqués pour extraire les mots les plus caractéristiques. Les résultats obtenus par cette méthode sont inférieurs à ceux obtenus à l'aide d'une pondération par la mesure $TF \times IDF$.

6 Conclusions

Nous avons décrit la participation du LINA à DEFT 2012. Notre système est le résultat de la combinaison des sorties de trois différentes méthodes d'extraction de mots clés. Les résultats obtenus par ce dernier sont toujours meilleurs que ceux obtenus par chacune des trois méthodes individuellement. Pour la seconde piste, notre système s'est classé à la $2^{ième}$ place sur un total de 9 systèmes avec une f-mesure de 21,3%.

La stratégie que nous avons employée pour combiner les sorties des différentes méthodes n'est cependant pas optimale. Nous envisageons d'étendre ce travail en proposant d'autres stratégies comme par exemple l'utilisation d'un meta-classifieur.

Références

Abeillé, A., Clément, L. et Toussenel, F. (2003). Building a treebank for French. *Treebanks : building and using parsed corpora*, pages 165–188.

Bird, S. et Loper, E. (2004). NLTK : The natural language toolkit. *In ACL*, Barcelone, Espagne.

Choi, F. Y. Y. (2002). *Content-based Text Navigation*. Thèse de doctorat, Department of Computer Science, University of Manchester.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. et Witten, I. (2009). The weka data mining software : an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Kiss, T. et Strunk, J. (2006). Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Medelyan, O. et Witten, I. H. (2006). Thesaurus based automatic keyphrase indexing. *In Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '06, pages 296–297, New York, NY, USA. ACM.

Mihalcea, R. et Tarau, P. (2004). Textrank : Bringing order into texts. *In* Lin, D. et Wu, D., éditeurs : *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Toutanova, K., Klein, D., Manning, C. et Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. *In Proceedings of the 3rd Conference of the North American Chapter of the ACL (NAACL 2003)*, pages 173–180. Association for Computational Linguistics.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C. et Nevill-Manning, C. G. (1999). Kea : Practical automatic keyphrase extraction. *CoRR*, cs.DL/9902007.