Utilisabilité d'une ressource propriétaire riche dans le cadre de la classification de documents

Résumé. Dans ce papier, nous nous intéressons à l'utilisation d'une ressource linguistique propriétaire riche pour une tâche de classification. L'objectif est ici de mesurer l'impact de l'ajout de ces ressources sur cette tâche en termes de performances. Nous montrons que l'utilisation de cette ressource en temps que traits supplémentaires de classification apporte un réel avantage pour un ajout très modéré en termes de nombre de traits.

Abstract. In this paper, we focus on the use of a proprietary resource for a document classification task. The objective is here to measure the impact of the addition of this resource as input for classification features. We show that the use of this resource impacts positively the classification results, for a limited impact on the feature number.

Mots-clés : classification de documents, classification automatique, ressources
Keywords: document level classification, automatic classification, resources

Introduction

1.1 Problématique

La classification de documents selon les thématiques protées par ceux-ci est un problème qui a été très largement étudié par la communauté scientifique. Les approches par classification automatique se sont révélées au cours des dernières années une solution tout à fait adaptée, à la fois en termes de performances et de rapidité/coût de mise en place.

Néanmoins, l'optimisation de ces résultats est toujours possible pour une problématique donnée, notamment via une sélection pertinente des attributs utilisés pour la classification, ainsi que par l'utilisation de primitives linguistiques fines et fiables.

Dans ce papier, nous nous intéressons à comment ces résultats peuvent être optimisés à partir de ressources linguistiques propriétaires génériques sémantiquement riche. L'objectif est ici de mesurer l'impact de l'ajout de ces ressources sur une tâche donnée de classification en termes de performances.

1.2 Travaux existants

La classification automatique thématique de documents est un domaine qui a été très largement étudié. Les travaux sur le sujet peuvent être divisés entre deux approches : la classification sur un jeu de catégorie connu a priori – citons par exemple les travaux de (Chai et al., 2002), et la classification sans jeu de classes, par regroupement de textes proches ou clustering – citons par exemple (Manning et Shütze, 1999), (Pappuswamy et al., 2005).

Notre approche se situe dans la première catégorie, plus simple, mais correspondant à un certain nombre de cas d'application concret (par exemple, la labellisation automatique d'articles de presse suivant une taxinomie d'articles prédéfinie. Notre objectif pour ce travail n'est pas tant d'obtenir un classifieur parfaitement optimisé – les résultats obtenus peuvent très certainement être amélioré – mais de mesurer l'impact de ressources existantes et la progression des résultats avec l'ajout de traits issus de ces ressources.

2 Notre approche

Nous avons choisi de tester l'adaptation des ressources linguistiques propriétaires sur une tâche de classification d'articles de presse. Nous décrivons en section 2.1 la méthodologie que nous avons employée. La section 2.2 décrit en détails les outils et ressources utilisées.

2.1 Méthodologie

Notre approche se déroule en deux phases. Tout d'abord, nous testons, sur un corpus de développement, plusieurs algorithmes de classification automatique. Cette phase nous permet de sélectionner le classifieur le plus adapté a priori pour la tâche, et d'adapter son paramétrage. Cette phase s'effectue sur un corpus plus restreint avec un jeu de traits donnés.

Dans un second temps, nous testons l'ensemble des combinaisons de traits à notre disposition pour le classifieur choisi. Cette approche en deux phases nous permet de limiter la combinatoire des tests effectués. Ceci est d'autant plus important que certains algorithmes testés en phase 1 sont particulièrement longs pour l'apprentissage.

2.2 Ressources linguistiques et outils de classification

Les ressources linguistiques utilisées sont celles qui sont commercialisées par Synapse Développement. Parmi les outils mis à disposition par cette société, nous avons à notre disposition deux outils : l'analyseur syntaxique de Synapse Développement, et l'extracteur de mots-clés, concepts-clés, et noms propres clés de Synapse Développement. Ce dernier est basé sur la taxinomie des concepts générique propriétaire de Synapse Développement.

L'analyseur syntaxique de Synapse Développement a été reconnu comme l'un des meilleurs pour le Français, notamment via plusieurs évaluations (Laurent et al. 2009). Nous utilisons pour notre tâche la version anglaise de l'analyseur, dont les performances sont au niveau de l'état de l'art.

La taxinomie des concepts de Synapse Développement est une ressource lexicale générique qui associe à chacun des termes et syntagme reconnus par l'analyseur syntaxique une ou plusieurs catégories, et ce suivant le sens reconnu si le terme est polysémique. Cette taxinomie se rapproche des synsets de WordNet, à la différence qu'ici le nombre de catégories conceptuelles est borné : la taxinomie est un arbre comptant quatre niveaux, avec 3387 catégories feuilles de rang 4, pour 256 catégories de rang 3. La figure ci-dessous montre un exemple en français de catégorisation pour l'adjectif polysémique « cher ».

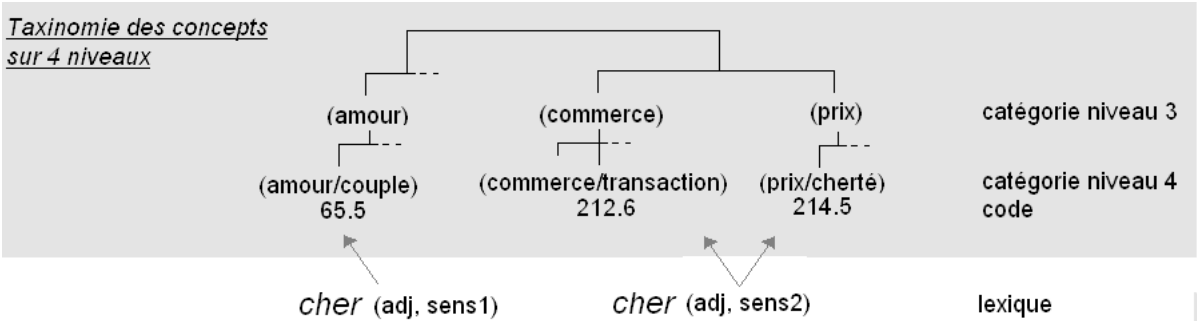

Figure 1 : exemple de catégorisation pour les deux sens de l'adjectif cher

La taxinomie de Synapse Développement étant disponible en multi-lingue (mêmes catégories), nous nous sommes appuyé pour les expérimentations la version anglaise de la ressource. L'extracteur de mots clés, concepts clés et noms propres clés est un composant s'appuyant sur cette taxinomie. À partir d'un texte, le composant extrait les mots clés, les noms propres clés, et les concepts de niveau 3 et 4 associés au document, avec un score de confiance entre 0 et 1 donnant l'importance du terme ou concept clé dans le document.

Enfin, nous avons utilisé pour les expérimentations les implémentations de la plate-forme Weka (Hall et al., 2009) pour la classification automatique.

3 Expérimentations

3.1 Données

Le corpus que nous avons utilisé afin de tester les technologies de Synapse et ce qu'elles peuvent apporter à la classification est une sous partie des nouvelles de l'agence de presse anglaise REUTERS[1] datées entre le 20-08-1996 et le 19-08-1997. Ces nouvelles sont déjà classée par l'agence en 125 catégories dont la plupart peuvent être apparentées à des sous-catégories de la classification IPTC17[2].

Ces documents sont classés en une classe principale et éventuellement plusieurs classes secondaires suivant le ou les thèmes abordés. La tâche que nous avons choisie est la reconnaissance de la classe principale du document. Afin de limiter les problèmes d'affectation d'un document à une classe secondaire, nous nous sommes limité pour la constitution des instances de test aux document ne comportant qu'une seule classe. Parmi les 125 catégories, nous avons choisi de ne garder que les 15 classes les plus occurrentes dans le corpus. Ceci nous permet notamment de disposer de suffisamment d'instances de documents ne possédant qu'une seule classe.

Le corpus de développement est constitué de 660 documents parmi l'ensemble des documents du corpus REUTERS complet. Ce corpus est utilisé dans la phase de sélection et paramétrage du classifieur. Pour la phase d'évaluation des traits le corpus d'entraînement de 2340 documents, et le corpus de test est constitué de 390 documents. Les documents ont été sélectionnés au hasard quant à leur contenu précis, mais de manière à équilibrer les différentes classes présentes.

L'ensemble des documents est rédigé dans un anglais impeccable, ce qui facilite l'analyse syntaxique du texte. Dans le cas contraire, une passe de correction orthographique et grammaticale automatique aurait pu être envisagé en pré-traitement des fichiers.

3.2 Choix et paramétrage du classifieur

Choix du classifieur

Nous avons effectué une première phase de classification sur le corpus de développement dans le but de sélectionner un algorithme adapté à notre tâche. Les algorithmes testés font partie des implémentations classiques de la plate-forme Weka. La version utilisée de l'outil est la version 3.6 (version stable en cours). Les résultats obtenus sont décrits dans le tableau 1 suivant.

[1] http://fr.reuters.com/

[2] http://www.iptc.org/site/Home/

Classifieur	Accuracy
Random Forest	**68.55 %**
J48	52.62 %
IBK	25.60 %
Bagging	46.77 %
Naive Bayes	37.50 %
K-Means	11.90 %
RepTree	53.43 %

Tableau 1 : accuracy sur le corpus de développement (toutes classes) pour chaque algorithme testé

Nous pouvons observé une nette prédominance de l'algorithme Random Forest (Breiman, 2001). C'est donc sur cet algorithme que s'est porté notre choix pour la suite des expérimentations.

Paramétrage du classifieur

L'algorithme Random Forest est basé sur la génération de k arbres se basant sur p traits aléatoires. Une valeur de p est proposée par l'implémentation de l'algorithme que nous avons utilisée. Cette valeur est calculée en fonction du nombre total de traits du corpus.

Nous avons évalué sur ce même corpus de développement les variations de performances de l'algorithme en fonction du nombre d'arbres. En faisant varié entre 10 et 800 le nombre d'arbres, nous avons observé que k=50 arbres était un bon compromis entre rapidité de calcul du modèle et performance, avec une accuracy quasi-maximale. Le maximum global observé pour 775 arbres n'est que très marginalement plus performant pour un temps de calcul très important.

3.3 Comparatifs des différents jeux de traits

Nous nous sommes ensuite basé sur les corpus d'entrainement et de test (décrits en section 3.1) pour effectuer une série de classification, dans le but d'évaluer l'impact de chaque trait sur les performances du classifieur.

Les traits utilisés pour les expérimentations sont les suivants :

- Mots (W) : les mots "bruts" qui composent le document. À chaque mot du corpus correspond un trait, actif s'il est effectivement présent dans le document.

- Lemmes (L) : la forme lemmatisée de ces mots. Comme pour les mots, à chaque lemme distinct présent dans le corpus correspond un trait, actif s'il est effectivement présent dans le document.

- Concepts-clés (CC) : les concepts clés associés au document. Ceux-ci sont extraits par le composant d'extraction des mots, concepts, et noms propres clés décrit en section 2.2. À chaque concept associé à un document est associé un trait booléen. Nous avons utilisé les concepts de niveau 3 et 4 pour la classification, et filtré les concepts clés suivant le niveau de confiance donné par le composant, avec une valeur seuil de 0.5.

- Mots-clés (MC) : les mots-clés associés au document. Ceux-ci sont extraits par le composant d'extraction des mots, concepts, et noms propres clés décrit en section 2.2. À chaque mot-clé d'un document est associé un trait booléen. Comme pour les concepts, nous avons filtré les mots clés suivant le niveau de confiance donné par le composant, avec une valeur seuil de 0.5.

- Noms Propres Clés (NPC) : les noms propres clés associés au document. Ceux-ci sont également extraits par le composant d'extraction décrit en section 2.2. À chaque nom propre clé d'un document est associé un trait booléen. Comme précédemment, nous avons filtré les noms propres clés suivant le niveau de confiance donné par le composant, avec une valeur seuil de 0.5.

Le tableau 2 suivant présente les résultats obtenus en termes de documents correctement classifiés :

Traits utilisés	**Accuracy obtenue (en %)**
W	56 %
L	60 %
W + MC	63.33 %
L + MC	64.66 %
W + MC + CC	71 %
L + MC + CC	**73 %**
W + MC + CC + NPC	69.66 %
L + MC + CC + NPC	**73 %**

Tableau 2 : accuracy sur le corpus de test (toutes classes) pour chaque jeu de traits testé

3.4 Discussion

Nous pouvons retirer plusieurs observations de ces résultats. Quantitativement, les résultats observés ne sont pas excellents quelque soit la méthode employée. Ceci vient très certainement d'une sélection un peu trop restreinte de textes d'entrainement et de test. L'objectif de ce papier n'est néamoins pas tant d'optimiser les résultats bruts que d'observer l'impact de l'ajout des traits de mots clés, concepts clés, et noms propres clés.

Le classifieur le plus performant sur la tâche présente un intérêt certain dans un contexte industriel. En effet, celui-ci présente l'avantage d'être assez performant en termes de vitesse d'apprentissage et de classification, ce qui réduit les coûts d'adaptation à une problématique client. De plus, le modèle généré étant basé sur un arbre de décision, il est tout à fait envisageable de compléter manuellement cet arbre à tout niveau avec des règles symboliques métiers spécifiques à un besoin client, et ce sans ré-entrainement du classifieur.

Côté traits de classification, la lemmatisation apporte globalement un plus important par rapport aux mots bruts. C'était attendu compte tenu de l'état de l'art, mais cela valide l'intérêt et la qualité du composant de Synapse dans le cadre de la classification.

L'ajout des mots clés apporte un gain de performance notable. Ces éléments sont donc intéressants à retenir pour la classification. Les noms propres clés n'apportent quasi aucune amélioration sur notre corpus. Ceci était également plutôt attendu en raison de la taille des corpus utilisés : pour pouvoir tirer parti de la redondance de la mention d'une même entité nommée entre deux textes de catégories similaires, le nombre d'exemples utilisés pour l'entrainement était sans doute trop restreint. Les concepts-clés apportent quant à eux un réel gain en termes de qualité de classification. On remarque également que même seul, ce trait est tout à fait discriminant pour le cas discuté. Ceci valide l'utilisabilité de cette pré-classification générique des documents pour aller vers une classification métier donnée.

Il est également intéressant de noter que le nombre de traits introduits par ces ressources est très réduit par rapport à des traits lexicaux classiques (ici mots ou lemmes, mais également n-grammes), ce qui permet de limiter l'impact sur le temps de calcul du modèle de classification. Ceci est particulièrement important dans le cadre d'une application industrielle.

4 Conclusion et Perspectives

Nous avons testé dans ce papier l'utilisabilité d'une ressource linguistique propriétaire en tant que source de traits de classification. Nous avons pu voir que l'ajout de concepts et mots clés issus du composant d'extraction de Synapse Développement permettait d'améliorer de manière significative les résultats obtenus.

Les résultats en eux-mêmes peuvent être optimisés en complétant le jeu de traits utilisé avec des traits complémentaires, notamment issus de l'analyse syntaxique : nous n'avons utilisé pour ces expérimentations que les lemmes, laissant de côté les groupes syntaxiques ("chunks") et les informations de désambiguïsation sémantique. Des expérimentations concernant ces traits sont en cours avec des résultats prometteurs.

Remerciements

[Paragraphe temporairement supprimé par les auteurs pour anonymisation]

Références (style *Titre sans numéro*)

BREIMAN L. (2001). Random Forests. *Machine Learning*. 45(1) :5-32.

CHAI KMA., CHIEU HL., Ng HT. (2002) Bayesian online classifiers for text classification and filtering. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* p. 97‑104.

HALL M, FRANK E, HOLMES G, PFAHRINGER B, REUTEMANN P, WITTEN IH. (2009) The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*. 11(1):10 8.

LAURENT D., NEGRE S., SEGUELA P. (2009) L'analyseur syntaxique cordial dans Passage. *Actes de TALN 2009*.

MANNING, C. AND SCHÜTZE, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, US.

PAPPUSWAMY U, BHEMBE D, JORDAN PW, VANLEHN K. (2005) A supervised clustering method for text classification. *Computational Linguistics and Intelligent Text Processing*. Springer p. 704‑14.