μ-*Alida*: expérimentations autour de la catégorisation multi-classes basée sur *Alida*

Adil El Ghali[1] Yann Vigile Hoareau[1,2]
(1) LUTIN UserLab, Cité des Sciences, 75019 Paris
(2) Université Paris 8, 93200 Saint Denis
elghali@lutin-userlab.fr, hoareau@lutin-userlab.fr

Résumé.

Dans article nous présentons le déroulement de notre concours DEFT'10, dans lequel nous nous sommes appuyés sur l'approche *Alida*. Nous introduisons quelques unes des améliorations apportées à l'approche et les illustrons par les résultats des exécutions soumises qui avaient pour but de les tester. Pour la tâche de variation diachronique nous avons réalisé avec l'aide de nos volontaires [1] une correction du corpus pour tester les effets de corpus bruités sur nos systèmes.

Abstract.

This paper presents our work in the context of the DEFT contest. We introduce some of the enhancements to *Alida*, the approach used. And we illustrate some of them with the results of the submitted runs. In the context of the task 1, we made some correction on the original corpus in order to observe the effects of noisy data on our systems.

Mots-clés : Catégorisation de documents, Random indexing, Alida.

Keywords: Classification, Random indexing, Alida.

1 Introduction

Le thème de la sixième édition du DÉfi Fouille de Textes (DEFT'10), est l'étude des variations diachroniques et géographique du français. Deux tâches nous ont été proposées par les organisateurs. La première consistait à identifier la décennie de publication d'articles de journaux sur une période comprise entre 1800 et 1944. La deuxième à identifier l'origine géographique de chaque document (pays d'origine) dans un corpus de presse rassemblant des titres provenant de France et du Québec.

Notre travail pour cette édition du DEFT'2010, dans la lignée de l'édition 2009 (Hoareau *et al.*, 2009b), a consisté à tenter d'apporter des améliorations à notre approche cognitive de catégorisation de textes basée sur l'exploitation des espaces sémantiques : *Alida*. Nous nous sommes principalement attelé à élaborer et à tester en utilisant le corpus du DEFT'10 des méthodes permettant de combiner plusieurs instances d'Alida, pour la catégorisation multi-classes de documents.

Cette article est organisé comme suit, dans une première partie nous rappelons les fondements d'*Alida*,

1. Sylvain Baron (Bytewise), Louis-Gabriel Pouillot (Hibox) et Kaoutar El Ghali

particulièrement sur les algorithmes d'attribution de catégories qui ont été développées et testées pour les besoins du DEFT'10. Nous présenterons ensuite les méthodes de combinaison de plusieurs instances d'*Alida*. Dans une deuxième partie, nous décrivons le déroulement de notre DEFT'10, en présentant les traitement réalisés sur le corpus et l'application de notre approche aux deux tâches. Nous conclurons notre article par une discussion des résultats et quelques perspectives de notre recherche.

2 Alida : une approche cognitive de la catégorisation de textes

Alida, l'approche issue des travaux (Hoareau *et al.*, 2009a; El Ghali *et al.*, 2009) que nous avons développé à partir de notre précédente participation au DEFT'09, se base sur une représentation des mots et des documents d'un corpus dans un espace sémantique (Karlgren & Sahlgren, 2001) implantant l'hypothèse distributionelle de (Harris, 1968). Cet espace est construit en utilisant Random Indexing (Sahlgren, 2006, 2005) et son implantation `semanticvectors` (Widdows & Ferraro, 2008).

Dans espace sémantique, sont représentés par leurs vecteurs, les documents de toutes les catégories. La phase d'apprentissage consiste à construire un vecteur prototype pour chaque catégorie en sommant l'ensemble des vecteurs de ses documents, puis à partitionner chaque catégorie en sous-catégories que nous appelons *cibles* représentant différents sous-prototypes de la catégorie.

Une fois les cibles constitués pour chaque catégorie, il s'agit de proposer des algorithmes pour attribuer la bonne catégorie à un vecteur-sonde (un document à catégoriser).

2.1 Attribution de catégories dans Alida

L'une des idées fondatrice d'*Alida*, présentée dans la section précédente, est de considérer pour chacune des catégories une décomposition en cibles. Par example, étant données, des catégories C et D, pour un nombre de cibles n, on calcule la similarité d'un document entrant d avec les cibles $C_1, ...C_n$ et $D_1, ..., D_n$. Se pose alors le problème de combiner de manière efficace ces $2 * n$ scores de similarité pour affecter la bonne catégorie à un document. Pour ce faire, nous avons élaboré et testé plusieurs méthodes de combinaison que nous décrivons brièvement ci-après :

Duel dans cette méthode, pour un document d à catégoriser, on compare deux à deux les valeurs de similarité entre d et les cibles de même rang (C_i avec D_i) et on distribue pour chaque rang i un nombre de points m entre les catégories de telle sorte que si la cible de rang i de la catégorie C : C_i est plus similaire au document à catégoriser que la cible de la catégorie D : D_i alors le nombre de points attribué, pour le document d, à la catégorie C est supérieur au nombre de points attribué á la catégorie D :

$$\mathsf{sim}(d, C_i) > \mathsf{sim}(d, D_i) \Rightarrow \mathsf{score}(d, C_i) > \mathsf{score}(d, D_i)$$

Par exemple, pour deux catégories C et D si le nombre de points à distribuer $m = 1$, si C_i est plus similaire que D_i pour un document d donné, alors $\mathsf{score}(d, C_i) = 1$ et $\mathsf{score}(d, D_i) = 0$.

Le score final d'une catégorie C pour un document d étant la somme des scores obtenu par toutes les cibles C_i de C. Et la catégorie attribuée au document d est celle qui aura obtenu le score le plus élevé.

Duel pondéré cette méthode de combinaison des similarités des cibles étend le principe du duel pour prendre en compte de manière algébrique le poids ce certaines cibles pour une catégorie donné. Cette prise en compte du poids de certaines cibles rend compte de l'importance qu'on accorde durant le processus de catégorisation par *Alida* à la particularité de certaines cibles. On peut, par exemple, favoriser les cibles qui contiennent les documents les plus typiques d'une catégorie en associant un poids élevé aux cibles de rang inférieur (les cibles les plus proches du prototype de la catégorie).

Le score final d'une catégorie C pour un document d est alors donné par la sommes des scores de ces cibles, pondéré par leurs poids respectifs :

$$\mathsf{score}(d, C) = \sum_i poids_i * \mathsf{score}(d, C_i)$$

MaxSim dans cette méthode, le score attribue à chaque catégorie C est la valeur de similarité la plus élevé de ses cibles avec le document à catégoriser. Il s'agit ici de considérer que chaque catégorie est représenté par sa cible la plus similaire au document à catégoriser :

$$\mathsf{score}(d, C) = \max_i \ \mathsf{score}(d, C_i)$$

SumSim (pondéré) dans ces méthodes, le score d'une catégorie C est obtenu en sommant les valeurs de similarité des cibles – éventuellement pondéré par le poids de leurs rangs – de C_i de C pour un document donné :

$$\mathsf{score}(d, C) = \sum_i \mathsf{sim}(d, C_i) \ \text{ resp. } \ \mathsf{score}(d, C) = \sum_i poids_i * \mathsf{sim}(d, C_i)$$

3 μ-*Alida*

Alida tout en étant conçu pour catégoriser des documents suivants plusieurs catégories, obtient de moins bonnes performances quand le nombre de ces catégories est important. Nous avons donc voulu tester si la combinaison de plusieurs instances d'*Alida* sur le même corpus en groupant des catégories dans certaines instances pouvait améliorer les performances. Deux types de combinaisons ont été envisagée :

(i) une combinaison hiérarchique, ou la catégorisation obtenu par une instance d'*Alida* ayant un nombre de catégories i ne pouvait être remise en cause par une instance d'*Alida* ayant un nombre de catégories $j > i$.

(ii) une combinaison algébrique, ou les résultats des catégorisations des différentes instances est projetté sur les suivantes. Nous avons choisi d'implanter ce type de combinaison pour le DEFT'10.

3.1 Projection algébrique

Soientt $\{C_1, ..., C_n\}$ l'ensemble des catégories d'un corpus, et $n_1, ..., n_i \in [2, \frac{n}{2}]$ des diviseurs de n. Le principe de μ-*Alida* est de créer $i + 1$ instance de *Alida*, la première correspond aux catégories d'origine, les i autres instances ayant pour catégories pour $j \in [1, i]$, $\{C'_1, ...C'_{n_j}\}$ avec C'_k la catégorie composée par l'union de $\frac{n}{n_j}$ catégories C_m successives de rang k.

Par exemple, pour $n = 4$, on aura une instance d'*Alida* avec comme catégories $\{C_1, C_2, C_3, C_4\}$ et une deuxième instance ayant pour catégories $\{C'_1, C'_2\}$ avec $C'_1 = C_1 \cup C_2$ et $C'_2 = C_3 \cup C_4$.

Chaque instance d'*Alida* permet d'associer à un document inconnu un score dans chacune des catégories (cf. 2.1). On obtient donc pour chaque document un tableau de scores de longueur n_j. Il ne nous reste plus qu'à projeter les tableaux de scores, après les avoir normalisé, les uns sur les autres en commençant par les instances ayant le plus petit nombre de catégories, comme le montre la figure 1.

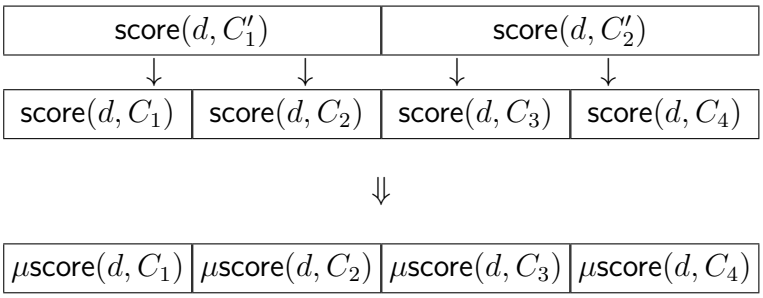

FIGURE 1 – Projection algébrique

4 Tâche 1 : Variation diachronique

4.1 Nettoyage du corpus

Le corpus diachronique qui nous a été fourni était issu de l'OCRisation de documents. il intègrait des erreurs de reconnaissance de caractères, telles que la suppression d'espace, des mots mal reconnu, ...

Nous avons décidé de tirer partie de cette particularité pour évaluer l'effet de ce genre d'erreurs sur nos algorithmes. Pour ce faire, une étape de correction partielle du corpus OCRisé à été réalisé avec l'aide précieuse de nos partenaires [2].

Plusieurs traitements ont été réalisé sur le corpus diachronique. Premièrement, une correction manuelle d'échantillons du corpus a été effectuée. Ensuite, ces corrections on été propagée sur l'ensemble du corpus. Enfin, des corrections à partir d'un correcteur orhographique opensource [3] ont été réalisés. Ces corrections sont de deux types :

1. la désagglutination : de nombreuses séquences étaient agglutinées, le passage du correcteur a permis de désagglutiner un grand nombre d'occurences.
2. le repérage d'erreurs systématiques : une modification des règles du correcteur a été nécessaire pour prendre en compte les erreurs les plus courantes de l'OCR. Par exemple, les mots contenant un caractère de ponctuation ont été traité par un ensemble de règles permettant d'obtenir des substitutions telles que : *!es* → *les*, *e)le* → *elle*, ...

2. Sylvain Baron (Bytewise) et Louis-Gabriel Pouillot (Hibox)
3. hunspell : `http://hunspell.sourceforge.net/`

4.2 Description des exécutions et Résultats

Pour la tâche de variation diachronique, le nombre de catégories était 15. Nous avons soumis trois exécutions, la première est une application d'*Alida* à 15 catégories sur le corpus brut, la deuxième est aussi une application d'*Alida* à 15 catégories sur le corpus corrigé et la troisième est l'application de μ-*Alida* à trois étages avec 3, 5 et 15 catégories sur le corpus brut.

Éxécution	Description	F-mesure	Médiane
#1	*Alida* corpus brut	0.116	
#2	μ-*Alida* corpus brut	0.156	0.181
#3	μ-*Alida* corpus corrigé	0.180	

TABLE 1 – Valeurs des F-mesures pour les exécutions de la tâche 1

La table 1 récupitule les performances des différentes exécutions soumises pour la tâche 1. Les résultats montrent que d'une part que μ-*Alida*, i.e. la combinaison de plusieurs instances d'*Alida*, améliore les performances par rapport à une application d'*Alida* avec le nombre de catégories initial. Et d'autre part, que les corrections du corpus améliorent aussi les performances.

5 Tâche 2 : Origine géographique

5.1 Description des exécutions et Résultats

Trois exécutions ont été soumise pour la tâche d'origine géographique, dans le but de tester l'effet de la méthode d'attribution des catégories dans les instances d'*Alida* sur les performances de μ-*Alida*.

La table 2 récupitule les performances des différentes exécutions soumises pour la tâche 2. Les résultats montrent que μ-*Alida* utilisant la méthode d'attribution Duel donne de meilleurs performance que μ-*Alida* avec SumSim. μ-*Alida* avec Duel pondéré donne les meilleurs performances pour la détermination du Pays tandis que μ-*Alida* avec Duel donne les meilleures performance pour la détermination du Journal.

Exécution	Description	Pays		Journal	
		F-mesure	Médiane	F-mesure	Médiane
#1	μ-*Alida* SumSim	0.762		0.424	
#2	μ-*Alida* Duel	0.798	0.792	0.446	0.462
#3	μ-*Alida* Duel pondéré	0.792		0.462	

TABLE 2 – Valeurs des F-mesures pour les exécutions de la tâche 2

6 Conclusion

Dans cette édition du DEFT'10, nous avons voulu tester un certain nombre d'optimisations de l'approche *Alida*, notamment en ce qui concerne les différentes méthodes d'attribution de catégorie. Nous avons aussi introduit un algorithme de combinaison de plusieurs instances d'*Alida*, qui permet d'améliorer les performances par rapport à une simple exécution d'*Alida*, dans le cas ou le nombre de catégories est important. Nous avons aussi eu l'occasion de tester les effets de la correction d'un corpus bruité sur les performances du système.

Remerciements

Nous tenons à remercier Sylvain Baron (Bytewise), Louis-Gabriel Pouillot (Hibox) et Kaoutar El Ghali pour leur aide précieuse sur la correction de corpus.

Références

EL GHALI A., HOAREAU Y. & EL GHALI K. (2009). The Episodic Memory Metaphor for Opinion Judgment Categorization. In *IADIS International Conference WWW/Internet (2)*, Rome.

HARRIS Z. (1968). *Mathematical Structures of Language*. New York : John Wiley and Son.

HOAREAU Y., EL GHALI A. & TIJUS C. (2009a). Detection of opinions and facts. a cognitive approach. In *Proceeding of Recent Advances in Natural Language Processing RANLP'09*, Borovets, Bulgaria.

HOAREAU Y. V., EL GHALI A. & LEGROS D. (2009b). Approche multi-traces et catégorisation de textes avec random indexing. In *Actes de l'atelier DEFT'09*, Paris.

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.

SAHLGREN M. (2005). An introduction to random indexing. *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*.

SAHLGREN M. (2006). *The Word-Space Model : Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis, Department of Linguistics Stockholm University.

WIDDOWS D. & FERRARO K. (2008). Semantic Vectors : A Scalable Open Source Package and Online Technology Management Application. In *Proceeding of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.