Interactions entre le calcul de collocations et la catégorisation automatique de textes

Rémi Lavalley — Patrice Bellot — Marc El-Bèze

Laboratoire Informatique d'Avignon (UPRES 931)
339, chemin des Meinajaries
Agroparc – B.P. 1228
F-84911 Avignon cedex 9
{remi.lavalley, patrice.bellot, marc.elbeze}@univ-avignon.fr

RÉSUMÉ. *Nous proposons dans cet article d'étudier les interactions entre l'extraction de collocations et la catégorisation automatique de textes. C'est-à-dire, dans un premier temps, utiliser la répartition des textes dans les différentes classes afin d'extraire des chaînes spécifiques à chacune (calculées par agglutination de collocations) ; puis, dans un second temps, utiliser ces chaînes spécifiques pour améliorer la catégorisation.*

ABSTRACT. *In this paper we describe some interactions between collocations and automatic text categorization. First, we use the differents categories to extract strings (through collocations agglutinations) related to each categorie. Then we use these categories-specific strings to improve categorization.*

MOTS-CLÉS : *collocations, catégorisation automatique de textes.*

KEYWORDS: *collocation, automatic text categorization.*

Rémi Lavalley, Patrice Bellot, Marc El-Bèze

1. Introduction

La masse croissante de textes disponibles nous pousse continuellement à chercher des méthodes facilitant leur manipulation. La catégorisation automatique de textes fait partie des nombreuses tâches de la recherche d'informations. Le problème consiste à rattacher un texte à une ou plusieurs catégories prédéfinies, ces catégories pouvant être par exemple le sujet du texte, son thème, l'opinion qui y est exprimée, ... Nous diposons pour cela d'un ensemble de textes pour lesquels la catégorie est connue (corpus d'apprentissage) et qui nous servent à entraîner nos modèles, modèles que nous exécuterons par la suite pour étiqueter automatiquement des documents de catégorie indéterminée (corpus de test).

Il existe un très grand nombre de méthodes numériques de catégorisation automatique de textes, qui se basent pour cela sur les mots contenus dans le texte. Des exemples de méthodes sont fournis dans (Yang *et al.,* 1999). La méthode proposée ici présente la particularité de n'utiliser non plus les mots isolés composant le texte, mais d'en regrouper certains (par la recherche de collocations) afin d'obtenir des termes ou expressions plus porteurs de sens.

Pour l'évaluation de nos propositions, nous avons utilisé le corpus de la campagne d'évaluation Défi Fouille de Textes 2007 (DEFT 07 (Grouin *et al.,* 2007)) portant sur la classification de textes selon l'opinion qui y est exprimée. Une partie du défi était basée sur des textes de critiques de jeux vidéos (4000 critiques, soit 28,3 Mo), pour lesquelles il fallait, en se basant sur le texte d'une critique, détecter si son auteur avait attribué une note bonne (classe 2), neutre (classe 1) ou mauvaise (classe 0) à ce jeu. Nous avons utilisé ce corpus car nous possédions déjà des systèmes de classification écrits spécifiquement pour lui lors de la campagne : le système présenté par le Laboratoire Informatique d'Avignon (LIA) (Torres-Moreno *et al.,* 2007) consistait en une combinaison de neuf classifieurs. Nous avons, pour notre travail, utilisé le classifieur *LIA_cosine* (calcul d'une distance - cosinus - entre les représentations vectorielles des documents (les poids des vecteurs sont fournis par TF.IFD (Salton *et al.,* 1988), combiné avec un facteur discriminant inspiré du critère d'impureté de Gini)). Par ailleurs, la campagne TREC 07 comportait une tâche de détection d'opinion dans les blogs, proche de celle étudiée ici. Un tour d'horizon de méthodes de détection d'opinion employées pour la tâche *blog track* de la campagne TREC 07 est fourni dans (Macdonald *et al.,* 2007).

Notre approche comporte deux parties : dans un premier temps (corpus d'apprentissage), on utilise la répartition des textes dans les différentes classes afin d'extraire des collocations (regroupements de mots) propres à chacune ; puis, dans un second temps, on utilise ces collocations pour améliorer la catégorisation du corpus de test. Ce travail offre de plus la possibilité d'indiquer à l'utilisateur quels sont les regroupements influents dans le choix de l'attribution d'une classe plutôt que d'une autre.

Des travaux similaires ont été proposés par (Wiebe *et al.,* 2001) (extraction de collocations pour la détection d'opinion), (Ferret, 2002) (utilisation pour la segmentation de textes), ou encore (Roche, 2006) (pour l'extraction de connaissances dans les

textes). Nous proposons pour notre part d'étudier l'influence qu'elles peuvent avoir sur la catégorisation de textes, en extrayant pour cela des collocations propres à chacune des classes.

Notre approche peut aussi être comparée à la classification à base de SVM à noyau de type séquence de mots (Gaussier *et al.,* 2003) : celle-ci permet en effet de considérer des n-grammes de mots en tant que composantes des vecteurs. Cependant, cette prise en compte des co-occurrences par les SVM opère de façon implicite. De ce fait, il est pratiquement impossible de visualiser les traits les plus discriminants. À l'inverse, l'agglutination de mots découlant des collocations offre de façon explicite la possibilité d'identifier les chaînes qui contribuent le plus à l'attribution d'une classe plutôt qu'une autre, ce qui peut notamment être intéressant dans un cadre de détection d'opinion. Bien sûr, il ne serait pas impensable de ré-utiliser ensuite ces chaînes en tant que composantes des vecteurs en entrée d'un autre système de classification (SVM par exemple).

La démarche est la suivante : le programme de calcul de collocations est inclus dans un système de catégorisation automatique de documents adapté à DEFT 07 (ce classifieur utilisait auparavant les mots pris séparément pour créer ses modèles et effectuer par la suite une répartition des textes dans les différentes classes) : une fois la classification effectuée, le programme de calcul de collocations (voir section 2) est lancé, proposant une liste de termes qu'il juge être des collocations avec un score associé. Puis, nous avons ajouté à ce système un programme englobant, qui permet d'automatiquement itérer, les meilleures propositions étant ré-injectées dans le système qui les applique (agglutination des termes composant une collocation dans l'ensemble des corpus), recommence la classification, ... De cette façon, le système utilise, à la place de mots isolés, les termes agglutinés (collocations), ainsi que les mots non-inscrits dans une collocation. On a donc pu suivre, au fur et à mesure des itérations, l'évolution des résultats de la classification suivant les propositions faites. De plus, on a obtenu - par agglutination d'agglutinations - des collocations allant au-delà de deux mots (par exemple : *graphisme-particulièrement-soigner, acheter-en-connaissance-de-cause* (les corpus ont été préalablement lemmatisés). L'évaluation des résultats est faite par le calcul du F-score (voir section 3).

Le but n'est pas ici de chercher à obtenir une analyse des textes selon une norme grammaticale (vraie dans l'absolu), mais leur modélisation selon un ensemble d'usages de la langue (propres au corpus considéré), qui ont une influence sur la classification, avec la possibilité pour l'utilisateur de voir quels sont ces éléments influents. Les résultats font d'ailleurs apparaître un ensemble de chaînes spécifiques parlantes pour l'utilisateur, bien que ne possédant pas forcément d'appellation dans la classification linguistique.

Dans un premier temps, nous définirons les collocations et la méthode utilisée pour les extraire, puis nous étudierons l'impact de ces collocations sur les résultats de notre système de catégorisation et présenterons des exemples de chaînes spécifiques à chaque classe, enfin nous énoncerons les problématiques restantes et les ouvertures possibles.

2. Collocations

Calculer des collocations consiste à trouver des mots qui "vont ensemble" (mots qu'il est naturel de trouver proches dans le langage, ces mots pouvant être contigus ou non). Il existe un certain nombre de méthodes pour trouver des collocations (voir par exemple (Yu *et al.,* 2003) et (Smadja *et al.,* 1990) pour des méthodes numériques, ou encore (Seretan *et al.,* 2004) pour une méthode utilisant des filtres syntaxiques appliqués au corpus du Web), la plus simple étant celle qui retourne les bigrammes les plus fréquents[1]. Nous allons présenter ici une méthode numérique se basant sur le rapport de vraisemblance.

De même, plusieurs méthodes ont déjà été proposées pour tenter d'évaluer les collocations (notamment des méthodes utilisant des dictionnaires dans (Thanopoulos *et al.,* 2002) ainsi que (Pearce, 2002) qui propose d'ailleurs une discussion sur la façon d'évaluer des propositions de collocations).

Nous allons expliquer pourquoi nous pensons que le fait de considérer ensuite ces mots comme une seule entité permet d'améliorer les performances d'un système de classification. Tout d'abord pour augmenter la significativité du terme : par exemple, si on arrive à repérer l'expression *effet particulièrement désagréable* dans un texte, on pourra supposer que la critique est négative, alors qu'un système classique aurait pris les mots séparément et aurait pu juger par exemple que :

– *effet* fait pencher vers une critique positive (comme dans l'expression "majestueux effets" ou "ce jeu fait bon effet") ;

– *particulièrement* vers une classe indéterminée ;

– *désagréable* vers une critique négative.

Ainsi, en considérant l'expression dans son intégralité nous pensons augmenter son pouvoir discriminant. En fait, l'agglutination de mots composants une collocation peut même intégralement remplacer l'utilisation d'un modèle n-grammes. Il s'agit alors d'un modèle unigramme dont les unités de base sont des n-grammes avec n variable. Le problème posé par le regroupement de termes est de savoir où s'arrêter (nombre de termes agglutinés), car créer des regroupements trop grands entraîne des problèmes de couverture (faible probabilité d'apparition de ces regroupements). La seconde raison qui nous pousse à penser que l'on peut améliorer les résultats vient du fait que l'on s'est ici servi de la répartition des textes dans les catégories, il est donc par exemple envisageable de créer des collocations propres à une catégorie.

Pour extraire les collocations présentes dans le corpus d'apprentissage, nous nous sommes appuyés sur la méthode du Rapport de Vraisemblance (*likelihood ratio*), car elle se distingue des autres par son efficacité selon (Daille, 1996) et (Dunning, 1993).

1. cette méthode ayant une tendance à plutôt proposer les combinaisons de mots-outils (Manning *et al.,* 2000)

Cette méthode permet d'évaluer la vraisemblance d'une hypothèse par rapport à une autre, les deux hypothèses étant ici :

– *les occurrences du mot* m_1 *sont indépendantes de celles du mot* m_2 ;

– *les occurrences du mot* m_1 *sont dépendantes de celles du mot* m_2 *- cas de collocation*.

Le logarithme de ce rapport (LRV) se calcule ainsi (développement de la formule exposée dans (Manning *et al.,* 2000)) :

$$
\begin{aligned}
log\Lambda = 2 \times ([C_{12} * \log \frac{C_{12}}{C_1} + (C_1 - C_{12}) \times \log (1 - \frac{C_{12}}{C_1})] \\
+[(C_2 - C_{12}) * \log \frac{C_2 - C_{12}}{N - C_1} + ((N - C_1) - (C_2 - C_1)) \times \log (1 - \frac{C_2 - C_{12}}{N - C_1})] \\
- [C_{12} * \log \frac{C_2}{N} + (C_1 - C_{12}) \times \log (1 - \frac{C_2}{N})] \\
- [(C_2 - C_{12}) * \log \frac{C_2}{N} + ((N - C_1) - (C_2 - C_{12})) \times \log (1 - \frac{C_2}{N})])
\end{aligned}
$$

[1]

avec :
C_1 le nombre d'occurrences de m_1 dans le corpus ;
C_2 le nombre d'occurrences de m_2 dans le corpus ;
C_{12} le nombre d'occurrences du bigramme $m_1 m_2$ dans le corpus ;
N le nombre de mots du corpus.

Ce logarithme fait correspondre un score à un couple de mots, score qui nous permettra par la suite de juger si ce couple peut être considéré comme une collocation significative.

Nous avons choisi, puisque l'on travaille sur une tâche de catégorisation, d'utiliser cette répartition des textes pour améliorer nos collocations, en extrayant des collocations propres à chacune des catégories. C'est-à-dire que le programme est lancé plusieurs fois (trois dans notre cas), en ne considérant pour chaque exécution que les textes du corpus d'apprentissage appartenant à une des catégories. Grâce à cela, nous obtenons trois sous-listes de collocations, spécifiques à chaque catégorie. Ces sous-listes seront regroupées en une seule lors de l'utilisation (agglutination des collocations possibles dans l'ensemble des corpus).

3. Expériences

Des exemples de collocations obtenues sont fournis dans le tableau 1. Il s'agit des collocations proposées après une itération du programme : des collocations sont calculées sur le corpus d'apprentissage, celles obtenant les meilleurs scores pouvant alors être utilisées comme pré-traitement du texte à la prochaine itération pour améliorer la catégorisation. On agglutinera alors les mots composants les collocations retenues, à la fois dans le corpus d'apprentissage et dans celui de test. Ceci nous permettra éventuellement, au fil des itérations, d'agglutiner des mots à des collocations déjà repérées, et ainsi d'obtenir des chaînes agglutinées de plus de deux mots. La figure 1 présente un schéma général du système.

Après expériences, on a choisi empiriquement de ré-injecter pour chaque classe toutes les propositions de collocations ayant un score supérieur à 200 et apparaissant plus de 10 fois dans le corpus d'apprentissage. Comme le montre le tableau 2, on a assez vite épuisé toutes les propositions que le système pouvait retourner (pas d'ajout de nouvelles collocations au-delà de la sixième itération).

classe 0			**classe 1**			**classe 2**		
score	**mot1**	**mot2**	**score**	**mot1**	**mot2**	**score**	**mot1**	**mot2**
6769	note	général	15674	note	général	12029	note	général
4537	bande	son	10717	bande	son	7733	bande	son
2490	de	vie	5777	de	vie	4101	de	vie
2291	durée	de	5151	il	faillir	4083	il	faillir
2076	ne-est	pas	5015	durée	de	3867	de	de

Tableau 1. *Les 5 premières propositions retournées après une itération par la méthode LRV pour les classe 0 (mauvaise note), 1 (note neutre), 2 (bonne note).*

Les exemples visibles dans le tableau 1 correspondent à des collocations propres au domaine considéré et que l'on peut retrouver dans les différents textes, quelle que soit l'opinion qu'ils expriment (commentaires sur la note attribuée, la bande-son du jeu, etc.). Ce sont celles qui ont obtenu le meilleur score. Mais, au-delà des premiers résultats, on pourra observer des rassemblements qui caractérisent les catégories. Ainsi, pour la catégorie 2 (bonne note), on trouve en 21^e position la collocation *très bon* (score de 1786) qui possède encore un bon score, mais est absente de la liste de la catégorie 1 et a un très mauvais score pour la catégorie 0 (82) ; à l'inverse, une expression comme *manquer cruellement* a un score de 617 pour la catégorie 0 et un score de 192 pour la catégorie 2.

L'évaluation de la catégorisation se fait par calcul du F-score (avec $\beta = 1$), selon la formule employée pour le classement des participations au défi DEFT 07 (voir formule 2), c'est-à-dire en utilisant une moyenne non-pondérée des scores de précision (formule 3) et rappel (formule 4) obtenus pour chacune des *n* catégories.

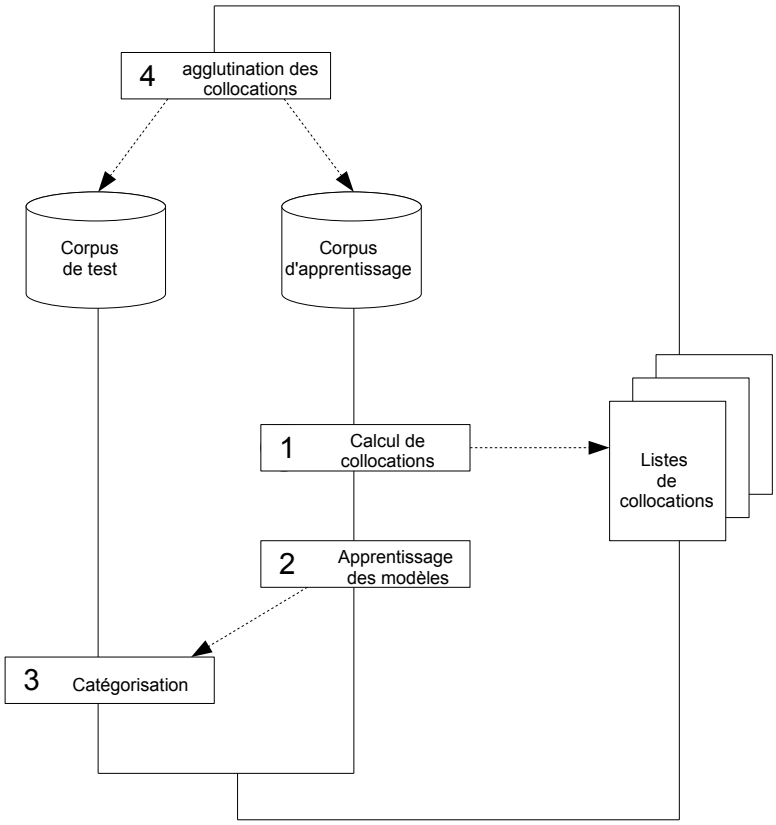

Figure 1. *Fonctionnement du système. Dans l'ordre, à chaque itération : 1) on extrait du corpus d'apprentissage une liste de collocations par catégorie (trois dans notre exemple) ; 2) on entraîne le classifieur sur le corpus d'apprentissage ; 3) on effectue la catégorisation du corpus d'apprentissage (+ calcul du F-score) ; 4) on utilise l'ensemble des listes de collocations réunies en une seule pour agglutiner dans l'ensemble des corpus les termes correspondants.*

$$F_{score}(\beta) = \frac{(\beta^2 + 1) \times Precision \times Rappel}{\beta^2 \times Precision + Rappel} \quad [2]$$

$$Precision = \frac{\sum_{i=1}^{n} Precision_i}{n} \quad [3]$$

$$Rappel = \frac{\sum_{i=1}^{n} Rappel_i}{n} \quad [4]$$

Ainsi, chaque catégorie compte à égalité, ce qui permet d'éviter qu'une catégorie plus peuplée qu'une autre ait un impact plus important sur les résultats.

itération	**modèles**	**F-score**
1	0	0,7530
2	1506	0,7653
3	2101	0,7610
4	2218	0,7672
5	2232	0,7683
6	2234	0,7688
7	2234	0,7688

Tableau 2. *Nombre de modèles de collocations utilisées pour le pré-traitement*

Les résultats de la catégorisation, fournis par la figure 2, sont assez intéressants : on voit que les résultats sont stables dès la 6^e itération (plus aucun ajout de collocations à partir d'ici) et on obtient un meilleur F-score qu'en n'utilisant pas de collocations (correspondant à l'itération 1). Cette méthode a ainsi le triple avantage de :

– converger rapidement ;

– se stabiliser au niveau du meilleur résultat (pas d'oscillations) ;

– fournir de meilleurs résultats qu'une classification sur des mots isolés (le gain observé de 1,5% est en effet important si on considère que notre classifieur obtenait déjà des résultats relativement élevés (voir section 4.1)).

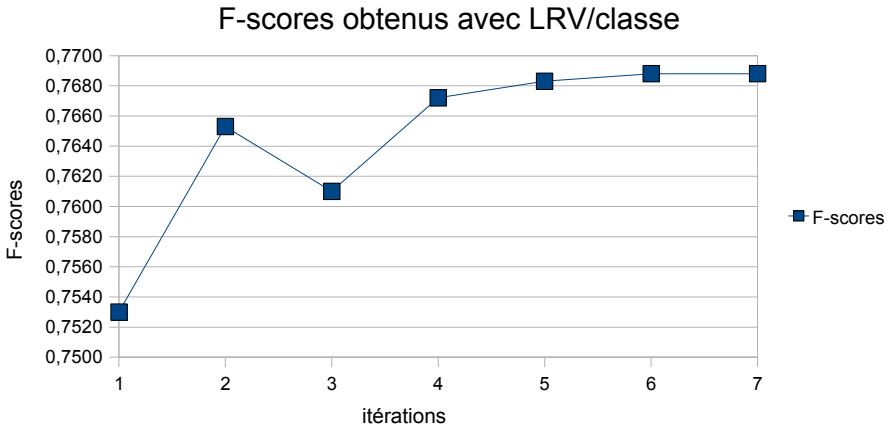

Figure 2. *Ce graphe indique les F-Scores obtenus selon l'itération. À chaque itération on ré-injecte les meilleures propositions de collocations calculées avec le LRV pour chaque catégorie.*

4. Résultats

4.1. *Résultats comparatifs*

Le tableau 3 permet de situer les résultats obtenus par rapport à d'autres méthodes.

système	**F-score**
LIA_cosine	0,7530
LIA_cosine + LRV	0,7634
LIA_cosine + LRV par classe	**0,7688**
LIA (vainqueur Deft)	0,7840
LGI2P (2^e à Deft)	0,7830
moyenne Deft	0,6638
LIA_cosine + manuel	**0,7882**
SVM	0,7410

Tableau 3. *Résultats obtenus par divers systèmes sur la tâche jeuxvideo de DEFT 07. Notre système a obtenu un F-score de 0,7688 - le meilleur score obtenu (sélection manuelle de collocations propres aux différentes catégories) est de 0,7882*

Les systèmes présentés sont :

– LIA_cosine : il s'agit de la base de notre classifieur (évoqué en introduction), utilisé sans applications de collocations ;

– LIA_cosine + LRV : le système précédent, prenant cette fois en compte des collocations calculées sur l'ensemble du corpus d'apprentissage, sans distinction de classe - cette méthode obtient de moins bons résultats que des collocations calculées par catégorie, mais possède l'avantage de pouvoir être entraînée sur l'intégralité du corpus (apprentissage + test), puisqu'on ne considère pas la catégorie pour le calcul de ce que peut être une collocation (voir section 5.2) ;

– LIA_cosine + LRV par classe : notre méthode : prise en compte de l'ensemble des collocations calculées séparément pour chacune des classes ;

– LIA (vainqueur Deft) : la participation du LIA au défi DEFT 07 (Torres-Moreno *et al.,* 2007) : fusion de 9 systèmes de classification (dont *LIA_cosine*, un système de boosting (BoosTexter), des machines à vecteurs de supports, un algorithme des k plus proches voisins, des arbres de classification sémantique, un modèle de probabilités n-gramme, ...) - il s'agit de l'équipe ayant remporté le défi[2] ;

– LGI2P (2^e à Deft) : la participation du LGI2P+LIRMM au défi DEFT 07 (Plantié *et al.,* 2007) : pré-traitements (lemmatisation, anti-dictionnaire, réduction par in-

2. Nous n'avons, pour l'instant, testé notre méthode que sur un seul des classifieurs (*LIA_cosine*) que nous avons utilisés lors de DEFT 07. Il serait intéressant d'observer le gain possible si nous utilisions nos chaînes agglutinées pour l'ensemble des neuf systèmes. De même, nous n'avons effectué nos tests que sur le corpus jeuxvideo de DEFT 07 : nous pourrions réaliser des tests sur les trois autres corpus fournis pour le défi.

formation mutuelle) et catégorisation par SVM - il s'agit de l'équipe ayant terminé deuxième du défi sur cette tâche ;

– moyenne Deft : moyenne de l'ensemble des participations à DEFT 07 pour la tâche portant sur les critiques de jeux vidéos ;

– LIA_cosine + manuel : sélection manuelle des collocations sur 50 itérations ;

– SVM : catégorisation par Machines à Vecteurs de Supports (bilbiothèque LIBLINEAR (Fan *et al.,* 2008)) entraînées spécifiquement pour cette tâche (noyau linéaire, antidictionnaire[3] de 726 entrées, les poids sont les TF.IDF normalisés (compris dans l'intervalle [0 ;1]), les paramètres *c=16* et *e=0,03125* ont été calculés sur le corpus d'apprentissage en validation croisée (5-folds) par tests de valeurs exponentielles successives $c=2^{-5}$, 2^{-4}, ..., 2^{6} et $e=2^{-6}$, 2^{-5}, ..., 2^{3}), sans prise en compte de collocations.

Les résultats *LIA (vainqueur Deft)*, *LGI2P (2^{e} à Deft)* et *moyenne Deft* sont ceux obtenus officiellement au défi DEFT 07 par les équipes participantes (Paroubek *et al.,* 2007) ; les autres correspondent à des expériences que nous avons réalisées dans le cadre de ce travail.

Afin d'évaluer la significativité des résultats, nous avons découpé le corpus de test en 10 sous-ensembles de taille identique (9 sous-ensembles de 170 critiques et le dernier de 164) et effectué le test de catégorisation - sans collocations ou avec la liste de 2234 règles d'agglutination obtenue à l'itération 6 de la méthode *LIA_cosine + LRV par classe* - sur les sous-ensembles pris séparément. La catégorisation avec collocations a été meilleure pour 8 des sous-ensembles. Les intervalles de confiance à 95% de ces deux tests sont [0,7119 ; 0,7742] sans prise en compte de collocations et [0,7378 ; 0,7826] avec. Les résultats se distinguent donc de façon significative, même si l'amélioration reste faible.

On observe ainsi que si le gain que nous avons obtenu pouvait sembler dans un premier temps quelque peu léger, cela provient notamment du fait que l'on parte avec un système qui donne déjà d'assez bons résultats (F-Score de 0,7530 alors que la moyenne des participations se situe à 0,6638 et les deux premiers à 0,7840 et 0,7830. Afin de situer notre système par rapport à un classifieur se basant sur les mots isolés, nous avons effectué des tests avec des Machine à Vecteurs de Support qui obtiennent un résultat de 0,7410. Enfin, la ligne intitulée "LIA_cosine + manuel" correspond à un travail de sélection manuelle des collocations sur 50 itérations. Cette sélection "propre" permet de nous représenter un objectif qu'il doit pouvoir être possible d'atteindre avec des méthodes numériques (gains en temps de travail, d'adaptation à un corpus différent, ...).

3. http ://www.up.univ-mrs.fr/~veronis/data/antidico.txt

4.2. *Aspects qualitatifs*

Au-delà du gain obtenu sur la catégorisation, l'autre avantage de notre méthode est la possibilité de présenter à l'utilisateur les chaînes (agglutinations de collocations) utiles pour la répartition dans les classes (i.e. les chaînes qui, présentes dans un texte, influenceront l'attribution d'une classe plutôt que d'une autre). Le tableau 4 présente quelques-uns des exemples retournés.

chaîne	**classe correspondante**
de-nombreux-défaut	0
tout-ce-que-il-y-de-plus-classique	0
aspect-répétitif	0
aucune-originalité	0
la-désagréable-impression	0
cumuler-défaut	0
beaucoup-trop-limiter	0
décevoir-un-peu	1
rien-de-bien-transcendant	1
bénéficier-d'-un-soin-tout-particulier	2
tenir-en-haleine	2
exempt-de-défaut	2
graphisme-particulièrement-soigner	2

Tableau 4. *Exemples de chaînes spécifiques à chacune des classes.*

On observe ainsi que *de-nombreux-défaut*[4] est une chaîne caractéristique des textes ayant obtenu une mauvaise note (catégorie 0). C'est-à-dire qu'une critique comportant cet ensemble sera très probablement mauvaise. Il en va de même pour les expressions *aucune-originalité*, *cumuler-défaut* ou encore *la-désagréable-impression*. A l'inverse, les expressions *exempt-de-défaut* ou *graphisme-particulièrement-soigner* seront caractéristiques d'une critique positive (catégorie 2).

Cet aspect est particulièrement intéressant dans un contexte applicatif : par exemple extraire, comme ici dans le cas de critiques de produits sur des sites, les points-clés sur lesquels s'expriment les clients, les défauts les plus souvent formulés, ... Par ailleurs, ce travail peut intéresser certaines entreprises dans le cadre de leur Gestion de la Relation Client. Par exemple, pour le traitement de grandes quantités de réponses à des enquêtes de satisfaction ou des remarques d'usagers : notre méthode pourrait leur permettre de détecter l'opinion exprimée par le client (catégorisation automatique de textes) et d'extraire les remarques "pertinentes" (les chaînes spécifiques à chaque classe).

4. les corpus ont été lemmatisés

5. Perspectives

5.1. *Choix des collocations à appliquer*

Un des grands problèmes dans l'utilisation des collocations est de savoir lesquelles retenir : certaines peuvent en effet se recouper, toutes n'ayant pas la même influence sur la classification finale. Actuellement, notre algorithme applique les règles dans l'ordre dans lequel il les rencontre (parcours gauche-droite de la phrase). De plus, on ne traite que des règles agglutinant les éléments deux à deux (et pas directement trois à trois par exemple). Il serait utile de mettre en œuvre une véritable stratégie d'application des règles. En effet, chercher à agglutiner la plus grande chaîne possible (première idée instinctive) n'est pas forcément une bonne idée. Ainsi, si on rencontre la suite de termes *m1 m2 m3 m5*, peut-être qu'agglutiner seulement *m3-m5* d'un côté et *m1-m2* de l'autre offre des meilleurs résultats que si on regroupait directement *m1-m2-m3-m5*. Il faudrait, pour optimiser les choix, disposer d'une information supplémentaire (chiffrage gain/perte) permettant de préférer en contexte une agglutination à une autre.

Voici deux exemples de problèmes soulevés par notre méthode actuelle, en supposant que l'on ait les règles d'agglutination proposées par le tableau 5.

règle	**T1**	**T2**	**agglutination (T1-T2)**
R1	m1-m2	m3-m4	m1-m2-m3-m4
R2	m1-m2	m3	m1-m2-m3
R3	m3	m5	m3-m5
R4	m2	m3	m2-m3
R5	m1	m2	m1-m2
R6	m3	m4	m3-m4

Tableau 5. *Exemples de règles d'agglutination de collocations, par exemple R6 signifie que si l'on rencontre les mots m3 et m4 côte à côte dans le texte on pourra les agglutiner pour former m3-m4.*

– problème 1 : on n'agglutine que deux termes à chaque passe, ainsi, selon l'ordre dans lequel les règles sont rencontrées, on ne pourra pas créer toutes les agglutinations, par exemple : on a la chaîne *m1 m2 m3 m4* : si, à la première passe, on a agglutiné m2 et m3 (R4), jamais on ne pourra appliquer la règle R1 ;

– problème 2 : cas de recoupement : si pour un même ensemble il est possible de créer plusieurs agglutinations différentes, comment savoir laquelle est la plus intéressante, par exemple : on a la chaîne *m1 m2 m3 m5* : comment choisir entre appliquer R3 *m1 m2 m3-m5* ou appliquer successivement R5 *m1-m2 m3 m5* et R2 *m1-m2-m3 m5* ?

Pratiquement, un exemple de problème qui pourrait se poser avec le corpus présenté serait celui provoqué par l'application des règles exposées dans le tableau 6.

règle	**T1**	**T2**	**agglutination (T1-T2)**
R7	très-utile	pour-mener-bien	très-utile-pour-mener-bien
R8	pour	mener	pour-mener
R9	très	utile	très-utile
R10	très-utile	pour	très-utile-pour
R11	pour	mener-bien	pour-mener-bien
R12	mener	bien	mener-bien

Tableau 6. *Exemples de règles d'agglutination de collocations du corpus jeuxvideo de DEFT 07.*

Ainsi, lors des extractions de collocations, le système a jugé que *pour mener* correspondait à une collocation possible, de même que *mener bien*, puis, au fil des itérations, la chaîne complète *très utile pour mener bien*. Cependant, selon si, au moment d'agglutiner ces collocations dans le corpus, on applique en premier la règle R8 ou le couple R12-R9, on n'obtiendra pas le même résultat. Dans le premier cas, on arrivera au mieux à segmenter ainsi : *très-utile pour-mener bien*. Dans le second cas, on pourra éventuellement arriver jusqu'à agglutiner la chaîne complète *très-utile-pour-mener-bien*. On voit ici les limites d'une méthode qui ne s'appuie pas sur une modélisation de la structure syntagmatique des phrases. Sachant les problèmes posés par une approche syntaxique (robustesse, manque de couverture, temps d'exécution), nous avons délibérément opté pour une analyse probabiliste.

5.2. *Autres perspectives*

En relation avec les problèmes exposés à la section 5.1, nous cherchons aussi une méthode pour calculer des collocations de plus de deux termes, de façon directe et non incrémentale comme nous le faisons actuellement. De plus, nous aimerions avoir un moyen de trouver des collocations "à trous" (termes non-obligatoirement consécutifs).

Par ailleurs, nous aimerions introduire les mêmes mécanismes qui amènent un être humain à lire un texte de façon différente suivant s'il le comprend d'une manière ou d'une autre. Regarder de bout en bout le texte traité sous un angle différent selon chacune des opinions envisagées, pourrait se traduire pour nous par l'emploi d'autant de jeux d'agglutinations qu'il y a de catégories. C'est-à-dire que, pour chaque texte, on effectuerait trois tests de catégorisation, en n'agglutinant pour chacun que les chaînes spécifiques à une des trois catégories. Puis, au regard de ces trois tests, on attribuerait à ce texte la catégorie ayant obtenu les meilleurs résultats. L'idée sous-jacente est que la manière dont les termes sont regroupés (et donc la phrase segmentée) peut faire varier totalement la catégorisation proposée par le système. Ainsi, la phrase *J'apprécie fort peu l'histoire*, fera probablement pencher la catégorisation vers une critique plutôt négative si elle est découpé comme suit : *J'apprécie-fort-peu l'histoire* et plutôt positive en suivant cet autre découpage : *J'apprécie-fort peu l'histoire*. Ces cas ne sont

pas improbables si on considère que nos corpus sont extraits de critiques laissées par des internautes (langage non-forcément académique), lemmatisés et sans ponctuation.

Pour comparaison, il nous semble intéressant, en utilisant la méthode du rapport de vraisemblance globale (collocations apprises sur les textes de toutes les classes confondues), d'extraire des collocations sur l'intégralité du corpus (apprentissage + test), ce qui augmenterait le nombre d'exemples sans toutefois introduire de biais, puisque l'on n'utilise pas la répartition dans les classes.

6. Conclusion

Si l'apport de la prise en compte des collocations dans la catégorisation des textes (orientée ici détection d'opinion) peut sembler avoir un impact faible au niveau du F-score (gain d'un peu plus de 1,5%), ces résultats sont à relativiser par rapport au score élevé qu'obtenait déjà le classifieur sur lequel nous nous sommes basés. De plus, il ne faut pas voir ici uniquement l'influence sur les résultats de la classification, mais aussi l'apport pratique pour l'utilisateur, c'est-à-dire la possibilité de lui montrer les chaînes spécifiques à chacune des classes. Si ces chaînes sont des agglutinations, le sens qu'elles sous-tendent peut être plus facilement perceptible que s'il s'agissait de termes simples (cas des chaînes telles que *rien-de-bien-transcendant* ou *exempt-de-défaut*). Ceci est particulièrement intéressant dans les cas de détection d'opinion, puisque cela nous permet par exemple de visualiser les reproches les plus souvent formulés. Enfin, il s'agit d'une méthode nouvelle, pour laquelle de nombreuses améliorations sont envisageables.

7. Bibliographie

Daille B., « Study and Implementation of Combined Techniques for Automatic Extraction of Terminology », *The Balancing Act : Combining Symbolic and Statistical Approaches to Language*, vol. 1, p. 49-66, 1996.

Dunning T., « Accurate methods for the statistics of surprise and coincidence », *Computational Linguistics*, vol. 19, p. 61-74, 1993.

Fan R.-E., Chang K.-W., Hsieh C.-J., Wang X.-R., Lin C.-J., « LIBLINEAR : A Library for Large Linear Classification », *The Journal of Machine Learning Research*, vol. 9, p. 1871-1874, 2008.

Ferret O., « Using collocations for topic segmentation and link detection », *Proceedings of the 19th international conference on Computational linguistics*, Taipei, Taiwan, p. 1-7, 2002.

Gaussier N. C. E., Goutte C., Renders J. M., « Word sequence kernels », *The Journal of Machine Learning Research*, vol. 3, p. 1059-1082, 2003.

Grouin C., Berthelin J.-B., Ayari S. E., Heitz T., Hurault-Plantet M., Jardino M., Khalis Z., Lastes M., « Présentation de DEFT 07 (DEfi Fouille de Textes) », *Actes de l'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, p. 1-8, 2007.

Macdonald C., Ounis I., Soboroff I., « Overview of the TREC 2007 Blog Track », *Proceedings of TREC 2007*, Gaithersburg, USA, 2007.

Manning C. D., Schütze H., *Foundations of statistical natural language processing*, MIT Press, p. 151-189, 2000.

Paroubek P., Berthelin J.-B., Ayari S. E., Grouin C., Heitz T., Hurault-Plantet M., Jardino M., Khalis Z., Lastes M., « Résultats de l'édition 2007 du DEfi Fouille de Textes », *Actes de l'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, p. 9-17, 2007.

Pearce D., « A Comparative Evaluation of Collocation Extraction Techniques », *Conference on language Resources and Evaluation*, p. 1530-1536, 2002.

Plantié M., Dray G., Roche M., « Défi DEFT07 : Comparaison d'approches pour la classification de textes d'opinion », *Actes de l'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, p. 57-69, 2007.

Roche M., « Acquisition de la terminologie et définition des tâches à effectuer, deux principes indissociables », *Actes des Journées de Rochebrune (Rencontres interdisciplinaires sur les systèmes complexes naturels et artificiels)*, p. 151-161, 2006.

Salton G., Buckley C., « Term weighting approaches in automatic text retrieval », *Information Processing and Management*, vol. 24(5), p. 513-523, 1988.

Seretan V., Nerima L., Wehrli E., « Using the Web as a corpus for the syntactic-based collocation identification », *Proceedings of International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, p. 1871-1874, May, 2004.

Smadja F. A., McKeown K. R., « Automatically extracting and representing collocations for language generation », *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, p. 252-259, 1990.

Thanopoulos A., Fakotakis N., Kokkinakis G., « Comparative Evaluation of Collocation Extraction Metrics », *The 3rd International Conference on Language Resource and Evaluation*, p. 620-625, 2002.

Torres-Moreno J.-M., El-Bèze M., Béchet F., Camelin N., « Comment faire pour que l'opinion forgée à la sortie des urnes soit la bonne ? Application au défi DEFT 2007 », *Actes de l'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, p. 119-133, 2007.

Wiebe J., Wilson T., Bell M., « Identifying Collocations for Recognizing Opinions », *Proceedings of the ACL/EACL Workshop on Collocation*, Toulouse, France, 2001.

Yang Y., Liu X., « A re-examination of text categorization methods », *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, Berkeley, California, USA, p. 42-49, 1999.

Yu J., Jin Z., Wen Z., « Automatic Detection of Collocation », *The 4th Chinese lexical semantics workshop*, Hong-Cong, 2003.