Extraction probabiliste de chaînes de mots relatives à une opinion

Rémi Lavalley[*, **] — **Chloé Clavel**[**] — **Patrice Bellot**[*]

[*] *LIA, Université d'Avignon et des Pays de Vaucluse*
Agroparc, BP 91228, 84911 Avignon Cedex 9 (France)
remi.lavalley, patrice.bellot@univ-avignon.fr

[**] *EDF R&D*
1, avenue du Général de Gaulle, 92141 Clamart (France)
chloe.clavel@edf.fr

RÉSUMÉ. *Nous proposons une méthode permettant d'extraire automatiquement des chaînes de mots relatives à des opinions à partir de corpus étiquetés. Il s'agit dans un premier temps d'améliorer les performances de systèmes de catégorisation automatique utilisés pour retrouver l'opinion (positive, négative ou neutre) rattachée à un texte. Dans un deuxième temps, la visualisation de ces chaînes permet d'avoir un aperçu des critiques fréquemment rencontrées. Cette méthode est testée sur des corpus en français ou en anglais de critiques de jeux vidéo et de films et sur un corpus d'enquêtes téléphoniques de satisfaction clients. Nous présentons des exemples de chaînes de mots extraites et les améliorations observées pour la catégorisation.*

ABSTRACT. *We present a probabilistic method aimed at extracting opinion-related strings from corpora labeled according to customer mind. These strings first allow us to improve text categorization systems according to opinions (positive, negative or neutral). Second, we use them to display easily what are the frequent comments made by customers about products or services. We test the method on two critical corpora written by internet users about video games and movies (respectively in French language and in English language) and on a customer satisfaction phone survey. For each of them, we present some examples of extracted word chains and the observed improvement obtained for opinion-oriented text categorization task.*

MOTS-CLÉS : *apprentissage automatique, classification d'opinion, Deft, extraction de collocations, Movies polarity dataset, questionnaires téléphoniques, SVM.*

KEYWORDS: *automatic opinion-oriented text categorization, collocation extraction, customer satisfaction phone survey, opinion-related strings, Movies polarity dataset, SVM.*

1. Introduction

La fouille automatique d'opinion connaît un essor important ces dernières années. Cet essor est rendu nécessaire par l'accroissement considérable des informations disponibles, conséquence de la démocratisation des nouvelles technologies de l'information et de la communication, et est soutenu par les avancées de la recherche notamment dans le domaine du traitement automatique des langues. En effet, l'accès à Internet permet à un grand nombre de personnes d'exprimer leur avis sur des sites, forums de discussion, blogs, etc. et favorise la collecte d'opinions par les entreprises grâce aux questionnaires électroniques. La fouille d'opinion est donc utile pour extraire « rapidement », de manière automatisée, les opinions des gens à propos de produits ou services. Ceci est particulièrement intéressant dans le cas des grandes entreprises, qui possèdent un grand nombre de données textuelles (enquêtes, centres d'appel, mails etc.) où les clients expriment leur opinion.

Ainsi, les travaux présentés dans cet article sont utilisés par EDF, qui possède plus de 40 millions de clients, et s'intéresse depuis plusieurs années à l'emploi de méthodes de TAL pour l'amélioration de la gestion de la relation clients (Cailliau et Giraudel, 2008 ; Bozzi *et al.*, 2009 ; Kuznick *et al.*, 2010 ; Danesi et Clavel, 2010).

La fouille d'opinion regroupe un certain nombre de sous-tâches se situant à plusieurs niveaux de granularité :

– au niveau du document, c'est-à-dire la tendance de l'avis exprimé dans un texte, est-elle positive, négative, positive et négative, neutre ? Les travaux au niveau du document sont régulièrement abordés lors de campagnes d'évaluation – la tâche « *opinion finding task* » de *TREC blog track* introduite en 2006 (Ounis *et al.*, 2007) consiste à retrouver des articles de blogs exprimant une opinion sur un produit, une personne donnés ; le défi fouille de textes Deft07 (Grouin *et al.*, 2007) portait sur l'attribution d'une opinion (positive, négative ou éventuellement neutre) sur des corpus de critiques de livres, spectacles, jeux vidéo, relectures d'articles scientifiques et débats parlementaires – et font l'objet de nombreuses recherches de la part d'entreprises qui souhaitent par exemple connaître les opinions exprimées par leurs clients au cours d'entretiens téléphoniques, d'une manière générale ou sur des points précis, comme la politesse ou l'efficacité des employés (Camelin *et al.*, 2006).

– les approches plus détaillées (à l'échelle de la phrase ou de l'expression) ont aussi donné lieu à de nombreux travaux, comme par exemple (Turney, 2002) sur la détection de la polarité au niveau de syntagmes (qui peuvent ensuite servir à définir si le document est une critique à tendance positive ou négative), proche des travaux présentés dans (Popescu et Etzioni, 2005) portant sur des critiques de produits et consistant dans un premier temps à repérer leurs caractéristiques, puis les syntagmes porteurs d'opinions sur ces caractéristiques et enfin à leur attribuer une polarité, (Wiebe *et al.*, 2001) sur la recherche de collocations subjectives (exprimant une opinion), ou encore (Wilson *et al.*, 2005) où les auteurs cherchent à déterminer si une expression est porteuse d'une opinion ou non, et si oui laquelle.

Nous proposons ici de traiter deux aspects importants de la fouille d'opinion, por-

tant sur les deux niveaux décrits ci-dessus :

– l'avis général exprimé dans un document : nous abordons cette tâche comme un problème de catégorisation automatique de textes ; la catégorisation automatique de textes (Sebastiani, 2002) consiste à rattacher des textes à une ou plusieurs classes (catégories) prédéfinies. Dans les expériences qui suivent, les catégories correspondent à des opinions (positive, négative ou neutre), le but est donc d'attribuer à chaque texte une de ces catégories, reflétant l'opinion exprimée majoritairement dans ce document, comme cela a par exemple été fait dans (Dave *et al.*, 2003) ;

– la recherche de chaînes de mots caractéristiques de ces opinions, permettant ainsi d'extraire les remarques, critiques ou points souvent formulés.

Notre travail se décompose en deux parties : dans un premier temps, nous nous servons de la répartition des textes dans les différentes catégories afin d'extraire des chaînes de mots relatives à chacune des opinions, par un processus itératif fondé sur une méthode d'extraction de collocations. Ces chaînes peuvent être visualisées pour obtenir un aperçu des remarques positives ou négatives fréquemment rencontrées pour un type de produit ou de service. Dans un second temps, nous utilisons ces chaînes pour améliorer des systèmes de catégorisation automatique supervisée de textes, en agglutinant dans les textes les chaînes que l'on y trouve, les classifieurs ne considérant ainsi plus uniquement des mots isolés comme vecteurs d'entrée, mais aussi des expressions caractéristiques des opinions qui sont par essence plus discriminantes.

La méthode permettant d'extraire les chaînes de mots et leur intérêt pour la catégorisation d'opinion sont présentés en section 2. Les techniques utilisées pour la catégorisation sont décrites en section 3. La méthode est testée sur deux corpus de catégorisation d'opinions en français (critiques de jeux vidéo extraites de la campagne Deft07 et transcriptions de réponses à des questionnaires téléphoniques proposés à des clients d'EDF), et un corpus en langue anglaise de critiques de films, tous trois présentés en section 4.

2. Méthode

2.1. *Extraire des chaînes relatives aux diverses opinions*

La méthode que nous proposons pour extraire des chaînes de mots relatives à chacune des catégories d'opinions est fondée sur une méthode d'extraction de collocations. Même si le terme « collocation » a plusieurs définitions dans la littérature, nous suivrons celle qui les présente comme une combinaison de mots cooccurrant plus souvent que par le fruit du hasard (Smadja, 1993). Nous nous sommes donc intéressé aux méthodes statistiques de recherche de collocations (Yu *et al.*, 2003 ; Smadja et McKeown, 1990), bien qu'il existe aussi des méthodes symboliques (Seretan *et al.*, 2004), mais celles-ci se prêtent moins facilement à une utilisation sur différents domaines ou sur des langues plus ou moins fortement dégradées (corpus bruités tels que ceux des blogs ou issus de transcriptions automatiques).

Les collocations obtenues par ces méthodes sont généralement évaluées par comparaison avec des dictionnaires de collocations (Thanopoulos *et al.*, 2002 ; Pearce, 2002). Cependant, notre but ici n'est pas de reproduire un dictionnaire de collocations, ni de produire des collocations syntaxiquement correctes, mais de rechercher des chaînes d'un nombre de mots indéterminé, relatives à une opinion, grâce à une méthode largement indépendante de la langue, puis de les exploiter pour une tâche de catégorisation selon l'opinion. Nous avons pour cela choisi une méthode purement statistique s'appuyant sur la notion du rapport de vraisemblance, introduite dans (Dunning, 1993) et réputée pour être la plus performante (Daille, 1996).

Parmi les travaux approchants, on peut citer Drouin (2004) qui extrait des termes isolés propres à un certain domaine, en comparant des textes d'un domaine spécifique par rapport à un corpus général. Nous nous différencions de cela, comme de Patin (2010), essentiellement par le fait que nous extrayons des chaînes, potentiellement de grande taille, relatives à des classes non thématiques plus que par l'approche centrale elle-même. Dans un but de constitution de terminologie, et non de détection d'opinion comme c'est le cas ici, Dias *et al.* (2003) présentent une architecture logicielle incluant un extracteur probabiliste, SENTA, qui produit des chaînes de longueur non fixée. Par rapport à notre approche qui est détaillée plus bas, la leur présente la caractéristique de ne pas avoir de seuil à fixer puisque sont retenues toutes les chaînes pour lesquelles la force d'association est diminuée lorsqu'un mot au moins leur est enlevé. Si cela constitue un avantage *a priori*, rien ne dit que les chaînes ainsi produites sont efficaces dans un but de catégorisation d'opinion et peut constituer, pour nous, une perspective intéressante. Par la suite, Dias et Vintar (2005) utilisent une fenêtre de 11 mots pour extraire des petits multimots (2 ou 3 mots), cependant ils utilisent les catégories grammaticales des mots, ce que nous ne faisons pas, car cela permet une plus grande indépendance vis-à-vis de la langue et une meilleure robustesse lorsque les données sont bruitées (corpus de blogs, données issues de l'oral...). Frantzi *et al.* (2000) proposent une méthode intéressante d'extraction de termes multimots, s'appuyant à la fois sur des indices linguistiques et probabilistes, la linguistique permettant un filtrage des expressions candidates.

Nous suivons, en ce qui nous concerne, l'idée introduite par Damerau (1993) qui proposait d'extraire des paires de mots spécifiques à un domaine et que nous appliquons au problème de la détection d'opinions, indépendamment des thèmes, nombreux et variés, abordés dans les textes. Notre méthode consiste principalement à extraire des collocations spécifiques à chacune des catégories (les opinions) par un calcul du logarithme du rapport de vraisemblance (LRV) appliqué sur les documents rattachés à une des opinions. Ces collocations peuvent ensuite être exploitées par différentes approches de catégorisation automatique.

On considère tous les documents du corpus d'apprentissage étiquetés avec une certaine opinion et on y applique un calcul de LRV. On obtient ainsi une liste de collocations par classe d'opinion, ordonnées selon leur score (LRV) et leur nombre d'occurrences dans le corpus d'apprentissage. Puis, nous agglutinons les collocations dont la force d'association et le nombre d'occurrences sont supérieurs à un certain

seuil (avec un filtre sur le LRV et le nombre d'occurrences), toutes classes confondues, dans l'ensemble du corpus (quelle que soit l'opinion attribuée au document), c'est-à-dire que si deux mots *m1* et *m2* font partie des collocations ainsi sélectionnées, on les considère comme un seul en remplaçant chaque occurrence de « *m1 m2* » par « *m1–m2* ». Puis, nous itérons cette procédure : on effectue un nouveau calcul de collocations à partir du corpus dans lequel les précédentes ont été agglutinées.

Le système va ainsi évaluer si deux « mots » « *m1–m2* » et *m3* ont un LRV et un nombre d'occurrences suffisamment élevé pour proposer la chaîne « *m1–m2–m3* ». Bien entendu, à ce stade, on peut aussi trouver des rassemblements de deux chaînes déjà agglutinées : si les occurrences de « *m1–m2* » et « *m3–m4* » possèdent une force d'association et un nombre d'occurrences suffisant, on les agglutinera pour donner la chaîne « *m1–m2–m3–m4* ». Lors de chaque itération, on agglutine les chaînes par ordre décroissant de taille résultante (on recherche dans le corpus toutes les chaînes de taille 4, puis celles de taille 3...). Au fur et à mesure des itérations on obtient des chaînes de plus en plus grandes. On arrête les itérations quand le système ne propose plus de rassemblements qui satisfont le filtre imposé en termes de LRV et nombre d'occurrences. Nous obtenons ainsi au final une liste de chaînes par catégorie (opinion) et un corpus sur lequel ces chaînes ont été agglutinées. Notons qu'il sera intéressant dans le futur de comparer notre approche à ce que l'on obtient à l'aide d'autres indices numériques tels que C-value (Frantzi *et al.*, 2000) qui pourraient aboutir à des listes de chaînes candidates différentes et plus ou moins nombreuses. Le pseudo-code suivant résume la méthode employée.

Soit L_i la liste des chaînes à l'itération i, $LRV(t1\ t2)$ la valeur du LRV pour le bigramme $t1\ t2$ et $nbOcc(t1\ t2)$ son nombre d'occurrences dans le corpus d'apprentissage.

1. Tant que de nouvelles chaînes sont trouvées ($L_i \neq L_{i-1}$)
2. Pour chaque catégorie C
3. initialiser $liste_C$
4. à partir des textes du corpus d'apprentissage $\in C$
5. Pour chaque bigramme de mots ou chaînes $t1\ t2$
6. Si $LRV(t1\ t2) > seuilLRV$ et $nbOcc(t1\ t2) > seuilNbOcc$
7. $liste_C += t1\ t2$
8. $L_i = L_{i-1} + liste_C$
9. ordonner L_i par taille décroissante
10. Pour tous les textes T (apprentissage + test)
11. Pour toutes les chaînes $ch = m1\ m2...mn$ de L_i
12. Si ch est présente dans T
13. Remplacer $m1\ m2...mn$ par $m1 - m2 - ... - mn$
14. $i++$

2.2. *Utiliser les chaînes pour la catégorisation de textes*

Nous nous proposons de tester l'impact de ces chaînes sur des problèmes de classification d'opinion, abordés comme des tâches de catégorisation automatique de textes. L'idée sous-jacente est que considérer des chaînes plutôt que les mots isolés seuls peut permettre de prendre en compte des expressions, des composants plus discriminants, qui seront plus porteurs de sens. Le repérage de ces chaînes peut désambiguïser certaines parties, voire permettre d'attribuer une étiquette totalement opposée à celle que l'on aurait trouvée en considérant les mots isolés. Par exemple, dans la phrase « *Ce produit n'est pas si bien* », il peut être intéressant de réussir à repérer « *pas si bien* », sans pour autant avoir un système de détection des négations. Ainsi, un tel système, s'il n'est pas assez sophistiqué, ne couvrirait probablement pas l'exemple suivant : « *Cet hôtel est loin d'être un paradis merveilleux* ». Dans cet exemple, un classifieur prenant en compte les mots isolés considérerait séparément les mots « *paradis* » et « *merveilleux* », donnant ainsi deux dimensions à tendance positive, qui feraient pencher la décision en direction de l'étiquette « appréciation positive », alors que si l'on savait repérer une chaîne du type « *loin d'être un paradis (merveilleux)* », le système pourrait réussir plus facilement à attribuer l'étiquette « appréciation négative ».

Ceci peut s'apparenter à une approche de type n-grammes, qui permet de prendre en compte, sans connaissance *a priori*, certains de ces phénomènes. Cependant, dans notre approche, le n n'est pas fixe, il s'adapte en fonction du besoin, pour repérer des chaînes de longueur importante (il pourra être égal à 3 pour repérer la chaîne « pas si bien » et égal à 6 pour repérer la chaîne « loin d'être un paradis merveilleux »). En plus de cette variabilité d'échelle, les chaînes nous permettent aussi de nous affranchir des problèmes des modèles n-grammes lorsque n devient trop grand (grande complexité pour des gains faibles, notre méthode extrayant des chaînes de plus de 10 mots).

D'après Moschitti et Basili (2004), l'utilisation de « multimots » n'apporte que peu voire pas du tout d'amélioration au niveau de la classification de textes. Cependant, nous avons choisi de tout de même tester l'influence de nos chaînes sur des systèmes de catégorisation pour les raisons suivantes :

– contrairement à celle présentée dans (Moschitti et Basili, 2004), notre méthode d'extraction de chaînes ne nécessite pas de calculs complexes et n'utilise pas de structures linguistiques ; elle possède ainsi l'avantage de pouvoir s'appliquer aisément sur un nouveau corpus (tâche et/ou langue différentes) et n'est pas consommatrice de ressources, il est donc toujours intéressant de l'employer, même si le gain est faible ;

– les chaînes que nous extrayons ne sont pas seulement relatives à un domaine, mais à une des catégories : elles sont donc plus discriminantes dans un contexte de catégorisation automatique de textes ; ce phénomène est accentué par l'emploi d'une pondération renforçant l'influence des termes les plus discriminants (voir formule 1) ;

– les multimots font généralement référence à des expressions de 2 ou 3 mots ; les chaînes que notre système retourne peuvent être de plus de 10 mots, couvrant ainsi des expressions, des portions de phrases potentiellement plus intéressantes.

3. Protocole pour la catégorisation

3.1. *Apprentissage et catégorisation*

Nous disposons d'un système qui extrait les chaînes, les applique et effectue une catégorisation, en fonctionnant par itérations qui correspondent à la recherche de chaînes de plus en plus longues. Ceci correspond à l'ajout d'une phase d'apprentissage des modèles, de catégorisation et d'évaluations des résultats entre les instructions 9 et 10 du pseudo-code présenté en section 2.1. Ainsi, à l'itération 1, on effectue une première catégorisation en n'utilisant aucune chaîne (mots isolés seulement) puis on effectue le calcul des collocations, à partir du corpus d'apprentissage, qui nous fourniront des chaînes de taille 2 ; à l'itération 2, on applique ces chaînes de taille 2 dans les corpus d'apprentissage et de test (on agglutine les mots les composant), on apprend les nouveaux modèles qui en découlent et on effectue une nouvelle catégorisation, puis on calcule des collocations en s'appuyant sur celles déjà agglutinées dans le corpus : on obtiendra donc des chaînes de tailles 2, 3 et 4, qui seront utilisées à l'itération 3. Ainsi, on peut voir l'évolution des résultats de la catégorisation en fonction de la longueur et la quantité des chaînes proposées.

Nous avons effectué la catégorisation de textes avec les deux classifieurs suivants :

– cosinusBasique : classifieur fondé sur la mesure de similarité cosinus entre un vecteur Wd du document d à classer et un vecteur Wc représentant la catégorie c (centroïde de la classe) dont les dimensions sont les $TF \times IDF$ (*Term Frequency - Inverse Document Frequency*) (Salton et Buckley, 1988) des mots ou chaînes i les composant. Dans nos travaux, les TF ont été remplacés par une mesure binaire de la présence ou non d'un terme dans le document, ceci afin de tenir compte des spécifités des données (par exemple pour le corpus issu de l'oral, cela nous évite de compter plusieurs occurrences d'un même mot s'il s'agit d'une répétition involontaire). Les IDF sont calculés en utilisant l'opposé du logarithme du nombre de documents contenant i divisé par le nombre total de documents ;

– cosinusGini : la mesure de similarité entre les vecteurs est la même que dans le cas précédent, mais les valeurs attribuées à chaque mot (ou chaîne) incluent en plus une variable $pGini$, nommée critère de pureté de Gini et introduite par Torres-Moreno *et al.* (2007). Sa valeur est calculée pour chaque mot en étudiant sa répartition dans les textes des différentes catégories (corpus d'apprentissage). Son rôle consiste à réduire l'influence des mots ayant un faible pouvoir discriminant (ceux qui apparaissent de manière équitablement répartie entre les différentes catégories). Sa valeur est ainsi maximale quand le mot n'apparaît que dans une seule catégorie et minimale quand il apparaît équitablement dans toutes les catégories. Dans ce cas, les composantes des vecteurs Wd et Wc se calculent comme présenté en formule 1.

$$W_{id} = TF_d(i) \times IDF(i) \times pGini(i); W_{ic} = TF_c(i) \times IDF(i) \times pGini(i)$$
$$avec : pGini(i) = \sum_c P(c|i)^2 \qquad [1]$$

Bien que les classifieurs de type mesure de similarité cosinus ne soient pas réputés pour être les plus performants, ils nous permettent d'effectuer rapidement un certain nombre de tests, notamment pour rechercher les valeurs optimales du filtre permettant de sélectionner les collocations. Ces systèmes ont en effet une complexité faible et de plus ils ne nécessitent pas d'ajustement de paramètres lorsqu'on ajoute ou retire des dimensions (ce à quoi correspond la prise en compte de nouvelles agglutinations). Pour chaque expérience, nous avons testé une vingtaine de filtres différents, d'après l'observation des propositions retournées par le système.

3.2. *Évaluation avec des machines à vecteurs supports*

Dans un second temps, nous testons l'apport de ces chaînes pour d'autres systèmes de catégorisation. Nous avons choisi d'effectuer des tests avec des machines à vecteurs supports (SVM) (Joachims, 1998). Un choix important pour la catégorisation par SVM est le type de fonction noyau à utiliser. Nous avons choisi d'utiliser des noyaux linéaires, car d'après Nallapati (2004) ils seraient les plus appropriés à la recherche d'information dans les documents textuels. Nous avons utilisé l'implémentation fournie par Liblinear (Fan *et al.*, 2008).

Pour chaque corpus, nous avons effectué plusieurs tests avec des SVM, en changeant les vecteurs d'entrée :

– sansChaînes-binaire : les dimensions des vecteurs sont tous les mots du vocabulaire (ensemble des mots présents dans le corpus considéré) – pour chaque texte, ces dimensions prennent la valeur 1 si le mot est présent dans le texte, 0 sinon ;

– sansChaînes-TF : les dimensions des vecteurs sont les mêmes que dans le cas du sansChaînes-binaire, mais les valeurs correspondent à la fréquence normalisée du mot dans le texte (la somme des fréquences pour un texte est égale à 1) ;

– chaînesMaxBasique[TF | binaire] : les dimensions des vecteurs sont tous les mots du vocabulaire que l'on retrouve dans le corpus lorsque toutes les chaînes ont été agglutinées (chaînes extraites d'après le corpus d'apprentissage) – il s'agit ici des chaînes qui nous ont permis d'obtenir les meilleurs résultats globalement sur l'ensemble des itérations du cosinusBasique ;

– meilleuresChaînesBasique[TF | binaire] : les dimensions des vecteurs sont tous les mots du vocabulaire que l'on retrouve dans le corpus lorsqu'on a agglutiné les chaînes qui nous ont permis d'obtenir ponctuellement le meilleur résultat avec la catégorisation par cosinusBasique à une certaine itération du système d'extraction de chaînes (par exemple, certains paramètres de filtre de collocations ont permis d'obtenir à l'itération 3 un très bon score de catégorisation, mais ces mêmes paramètres ont été moins efficaces sur les autres itérations) ;

– chaînesMaxGini[TF | binaire] : idem SVM-chaînesMaxBasique, mais avec les chaînes qui ont été les meilleures pour la catégorisation avec le cosinusGini ;

– meilleuresChaînesGini[TF | binaire] : idem SVM-meilleuresChaînesBasique, mais avec les chaînes qui ont été les meilleures pour la catégorisation avec le cosi-

nusGini.

Pour chacune de ces expériences, les paramètres des SVM (le coût C et le critère d'arrêt ϵ) ont été optimisés en maximisant le score de catégorisation sur le corpus de développement par une recherche en grille : $C = 2^{-5}, 2^{-4}, ..., 2^{15}$ et $\epsilon = 2^{-15}, 2^{-14}, ..., 2^{3}$ (Hsu *et al.*, 2003). Ceci nous permet d'avoir une idée de ce que peuvent apporter nos chaînes à un système de classification état de l'art, en testant certains jeux de chaînes qui se sont révélés efficaces pour la classification par cosinus. Bien entendu, cela n'est pas aussi complet que d'évaluer une multitude de paramètres de filtres de collocations sur les SVM (comme nous le faisons pour les cosinus), mais la complexité de ceux-ci ne permet pas d'effectuer autant de tests qu'avec le cosinus.

3.3. *Une méthode de repli*

La recherche de chaînes longues relatives aux classes est intéressante pour repérer certaines particularités comme nous l'avons expliqué au début de cette section. Cependant, aller chercher des chaînes de plus en plus longues pose petit à petit des problèmes de couverture : la chaîne très longue et peu présente dans le corpus cache peut-être des courtes chaînes qui auraient plus de poids dans le classifieur (dans l'exemple « *ce produit est merveilleusement génial* » imaginons que « *merveilleusement* » et « *génial* » aient déjà toutes deux une forte propension à faire pencher le classifieur vers l'attribution de l'étiquette positive, peut-être alors que les remplacer par un seul mot « *merveilleusement-génial* » est moins intéressant). Pour tenir compte de ce fait, nous avons tenté une méthode de repli « basique » consistant à prendre en compte à la fois la chaîne comme un mot unique ainsi que tous les mots isolés qui la composent. Pour ces derniers, on a appliqué différents coefficients, permettant de les considérer (s'ils n'existaient pas déjà à l'état de mot isolé, hors chaîne) avec la même importance que la chaîne qu'ils composent ou comme une fraction d'occurrence du mot (comme s'ils occurraient une fraction de fois) : avec une valeur de 1, ou avec des valeurs plus petites (0,1 0,2 0,25 0,3 ou 0,5).

L'impact sur la classification est évalué par F-score moyen, selon la formule employée lors du défi Deft 07, c'est-à-dire en utilisant les macro-moyennes des scores de précision et rappel obtenus. 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 (favorisant ainsi un système qui attribuerait toujours la classe majoritaire).

Nous avons choisi une répartition des corpus en apprentissage, développement, test plutôt que d'effectuer une validation croisée pour les raisons suivantes :

– ne pas biaiser les résultats de la catégorisation, les chaînes, dépendant des catégories des documents, sont apprises sur le corpus d'apprentissage uniquement. Effectuer une validation croisée en n sous-parties obligerait à conserver n versions différentes du corpus, chacune s'étant vu agglutiner des chaînes apprises sur $n - 1$ sous-parties ; ce qui impliquerait d'effectuer n adaptations des paramètres des SVM pour chaque corpus, soit un temps de calcul très élevé ;

	classe 0	classe 1	classe 2	Total
Apprentissage	395	905	729	2 029
Développement	102	261	145	508
Test	332	779	583	1 694
Total	829	1 945	1 457	4 231

Tableau 1. *Répartition du nombre de critiques pour le corpus Deft07-jeuxvidéo*

Nombre de mots total	5 957 578
Nombre de mots uniques	67 104
Nombre de lemmes uniques	45 407
Nombre moyen de mots par document	1 408
Nombre de mots du plus petit document	16
Nombre de mots du plus grand document	4 847

Tableau 2. *Statistiques sur le corpus Deft07, les comptes des mots incluent les symboles et signes de ponctuation*

– respecter la chronologie des corpus (notamment celui d'enquêtes téléphoniques) : il ne nous semble pas pertinent d'utiliser par exemple des modèles appris sur des documents de l'année 2009 pour catégoriser des documents de l'année 2007.

4. Expériences

4.1. *Corpus Deft07–jeuxvidéo*

Le premier corpus que nous avons choisi pour tester nos méthodes provient de la campagne d'évaluation Deft07. Parmi les quatre corpus proposés au défi (Grouin *et al.*, 2007), nous avons choisi les critiques de jeux vidéo : un internaute attribue une note (sur une échelle de 0 à 20) à un jeu et laisse un commentaire. Le but du défi est de retrouver l'appréciation globale (positive, négative ou neutre) laissée par l'internaute à partir du texte de sa critique. Le passage d'une échelle de 0 à 20 à une échelle restreinte de 0 (appréciation négative – notes de 0 à 9) à 2 (appréciation positive – notes de 15 à 20) a été défini par les organisateurs du défi à partir de l'étude d'un coefficient Kappa entre juges humains et vis-à-vis de la référence. Nous avons utilisé comme corpus de test les 1 694 documents qui servaient à l'évaluation lors du défi. 508 des 2 537 documents (tirés aléatoirement) de l'apprentissage nous ont servi de corpus de développement. L'avantage de ce corpus est que nous disposons, grâce à la campagne d'évaluation, de références pour ces données. La répartition des documents dans les classes pour chacun des sous-corpus est présentée dans le tableau 1, tandis que le tableau 2 présente quelques statistiques sur la dimension du corpus.

Nous avons utilisé LiaTagg[1], un étiqueteur morphosyntaxique basé sur l'étiqueteur ECSTA (Spriet et El-Bèze, 1999), afin d'obtenir les formes lemmatisées. Nous avons choisi de conserver tous les mots, quelle que soit leur catégorie grammaticale, afin

1. http ://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/bechet/download_fred.html

Itération	Filtre [75;5]	Filtre [50;5]
1	0	0
2	6 034	9 335
3	10 427	16 513
4	11 808	18 590
5	12 061	18 938
6	12 114	18 980
7	12 133	18 982
8	12 135	18 982

Tableau 3. *Nombre de modèles de chaînes selon les itérations – corpus Deft07-jeuxvidéo-développement*

que les chaînes que nous extrayons soient cohérentes et compréhensibles. De plus, les mots-outils peuvent être porteurs d'opinion. De même, nous avons choisi de conserver les symboles et signes de ponctuation, ceux-ci pouvant avoir une certaine signification pour du texte écrit par des internautes (*smileys*), ils permettent aussi d'avoir des chaînes cohérentes (par exemple, on utilise la ponctuation dans le calcul des chaînes afin de ne pas obtenir une chaîne à cheval sur deux phrases).

4.1.1. *Développement*

Pour le cas du cosinusBasique, les meilleurs paramètres de filtre de sélection des collocations trouvés [LRV minimal ; nombre d'occurrences minimal] sont [75 ;5]. La figure 1 montre les variations du F-score en fonction des itérations. Les meilleurs résultats sont obtenus à l'itération 3 (chaînes de 4 mots au maximum), où l'on parvient à étiqueter correctement 386 critiques sur les 508 du corpus de développement (F-score = 0,753). Le gain observé de 2,9 % correspond à 2,3 % de documents mieux catégorisés dans l'absolu, c'est-à-dire que les documents mieux classés appartiennent aux classes moins peuplées, qui sont les classes 0 et 2 : les chaînes ont donc été utiles pour aider à retrouver des avis positifs et négatifs exprimés sur les produits. Au-delà, les chaînes n'aident plus à améliorer la catégorisation. Le tableau 3 indique le nombre de modèles de chaînes que l'on connaît à chaque itération (colonne [75 ;5]).

Pour le cas du cosinusGini, le meilleur filtre de sélection des collocations trouvé est [50 ;5], c'est-à-dire que l'on est plus laxiste dans la sélection de ce qui pourra former une chaîne intéressante. On introduit plus de modèles de chaînes, on en propose donc plus de discriminantes et le critère de Gini renforce leur pouvoir lors de la prise de décision. Sans utilisation de chaînes, le classifieur utilisant Gini est légèrement meilleur que le basique. Il devient particulièrement intéressant à partir du moment où l'on utilise des chaînes de plus de 2 mots : le pic à la troisième itération est supérieur et surtout les résultats se stabilisent à un F-score plus élevé.

Des SVM ont ensuite été entraînés sur le corpus d'apprentissage dans lequel ont été agglutinées les chaînes extraites à certaines étapes clés des précédentes expériences. Nous avons ainsi réalisé une catégorisation avec le corpus sans utilisation de chaînes, puis en utilisant les chaînes extraites aux itérations 3 de chacun des deux filtres [75 ;5]

et [50 ;5], puis avec les corpus correspondant aux itérations finales des expériences précédentes. Les résultats obtenus sont présentés dans le tableau 4.

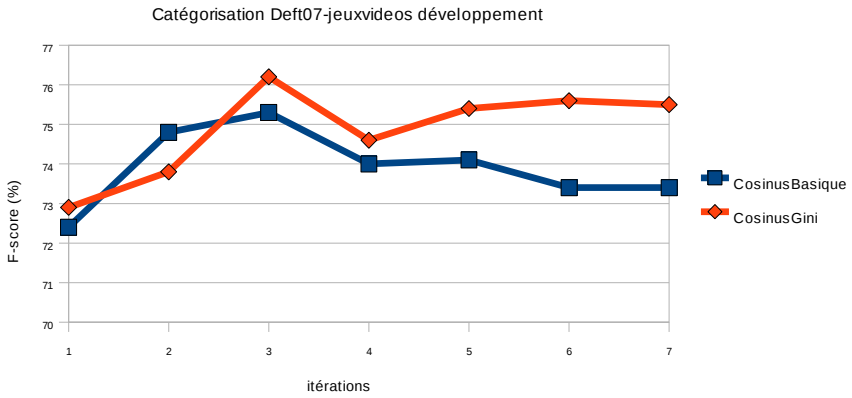

Figure 1. *Résultats des catégorisations par cosinus sur le corpus de développement de Deft07-jeuxvideo*

On peut constater que les résultats obtenus par les SVM sur les mots isolés sont supérieurs à ceux des cosinus dans la même configuration et au même niveau que cosinusGini utilisant les chaînes. Par ailleurs, les chaînes n'aident pas à mieux classifier avec les systèmes à base de SVM, à l'exception du cas où on a utilisé les chaînes de 4 mots au maximum avec le filtre [75 ;5]. Il faut cependant noter que l'utilisation de ces chaînes permet de significativement réduire le temps nécessaire pour effectuer la catégorisation avec SVM. Ainsi, pour des résultats assez proches, la recherche des paramètre optimaux (C et ϵ), consistant à entraîner et tester 399 modèles de SVM, s'effectue en environ 14 heures avec le texte original et 3 heures dans le cas *meilleuresChaînesBasique-binaire*, sur une machine de type 2×3 Ghz Quad Core Intel Xeon (chaque expérience n'utilisant qu'un seul processeur) avec 6 Go de RAM. Ce gain de temps s'explique par le fait que les chaînes agissent comme un prétraitement implicite aux SVM, en leur indiquant directement des rassemblements de dimensions discriminantes.

Les expériences avec repli ont donné des résultats identiques ou inférieurs à l'expérience correspondante sans repli, à de rares exceptions près. L'inefficacité du repli peut s'expliquer par le fait que l'on effectue le travail inverse de celui réalisé en agglutinant des chaînes : on avait auparavant indiqué aux SVM des dimensions à rapprocher et on lui rajoute maintenant les composantes détachées.

4.1.2. *Test*

Nous avons confirmé certains de nos résultats sur le corpus de test : les cosinusGini et cosinusBasique avec respectivement les deux filtres de sélection de collocations [50 ;5] et [75 ;5], les SVM chaînesMaxBasique-binaire, ainsi que les SVM sans utili-

Classifieur	Comptage	Chaînes	ϵ	C	F-score
SVM	TF	sansChaînes	2^{0}	2^{10}	0,759
SVM	TF	chaînesMaxBasique	2^{-15}	2^{9}	0,723
SVM	TF	chaînesMaxGini	2^{-2}	2^{8}	0,713
SVM	TF	meilleuresChaînesBasique	2^{-15}	2^{9}	0,716
SVM	TF	meilleuresChaînesGini	2^{-4}	2^{9}	0,725
SVM	binaire	sansChaînes-binaire	2^{0}	2^{10}	0,759
SVM	binaire	chaînesMaxBasique	2^{-15}	2^{9}	0,723
SVM	binaire	chaînesMaxGini	2^{-2}	2^{8}	0,713
SVM	binaire	meilleuresChaînesBasique	2^{0}	2^{-4}	0,767
SVM	binaire	meilleuresChaînesGini	2^{-4}	2^{9}	0,725

Tableau 4. *F-scores obtenus sur la catégorisation du corpus de développement de Deft07-jeuxvidéo par des SVM, ainsi que les paramètres optimaux C et ϵ trouvés*

sation de chaînes pour les valeurs binaires et TF. Pour les SVM, nous avons conservé pour chaque type les paramètres optimaux (C et ϵ) appris sur le développement.

Nous avons effectué les catégorisations avec et sans le corpus de développement intégré dans l'apprentissage (respectivement « *avec DEV* » et « *sans DEV* »). Dans les cas des cosinus, les résultats sont meilleurs de 1,5 % : le corpus est plus grand, il contient donc plus de données d'apprentissage qui permettent d'avoir des modèles plus représentatifs et de meilleurs exemples de chaînes relatives aux classes. Les expériences équivalentes ont été menées avec les SVM, améliorant faiblement les F-scores.

Le tableau 5 récapitule les résultats que nous avons obtenus sur le corpus de test ainsi que certaines références issues de la campagne d'évaluation Deft07 (Paroubek *et al.*, 2007). Les quatre résultats des cosinus correspondent aux résultats obtenus à la dernière itération (de meilleurs résultats sont obtenus lors des itérations précédentes mais cela n'a pu être détecté à la volée).

Le cosinusGini (avec le corpus de développement inclus dans l'apprentissage) utilisant les chaînes s'avère donner le meilleur score de classification de tous les systèmes testés, avec un F-score supérieur de 1,2 % au meilleur des SVM. Il possède de plus l'avantage d'être très rapide, par opposition aux SVM qui sont longs à entraîner. Enfin, on peut considérer qu'il est relativement robuste, alors qu'avec les SVM ce ne sont pas toujours les mêmes configurations qui donnent les meilleurs résultats. On observe tout de même que la configuration utilisant les chaînes maximales trouvées avec le filtre [75;5] et prenant en compte un repli à 1 donne des résultats proches du meilleur et est plus rapide à l'entraînement que des SVM sans chaînes. On note que dans le cas de notre meilleur système (cosinusGini - avec DEV), l'apport des chaînes pour la catégorisation est assez restreint (+ 1 % entre la première et la dernière itération). Il faut toutefois prendre en compte le fait qu'on obtient ici d'assez bons résultats pour ce seul système : meilleur que tous les tests avec SVM et si on le compare aux participants à Deft, on est à 10 % au-dessus de la moyenne des participants (ligne « moyenne Deft ») et 0,8 % en dessous du système vainqueur (« vainqueur Deft »), celui-ci consistant en réalité en une fusion d'une dizaine de classifieurs, qui donne des résultats supérieurs

Classifieur	Comptage	Chaînes	F-score
SVM	TF	sansChaînes, sans DEV	0,763
SVM	binaire	sansChaînes-binaire, sans DEV	0,739
SVM	binaire	chaînesMaxBasique, sans DEV	0,755
SVM	binaire	chaînesMaxBasique, repli=1, sans DEV	0,757
SVM	TF	sansChaînes, avec DEV	0,764
SVM	binaire	sansChaînes, avec DEV	0,750
SVM	binaire	chaînesMaxBasique, avec DEV	0,758
SVM	binaire	chaînesMaxBasique, repli=1, avec DEV	0,763
CosinusGini		sans chaînes	0,765
CosinusBasique		sans DEV	0,756
CosinusGini		sans DEV	0,761
CosinusBasique		avec DEV	0,772
CosinusGini		avec DEV	**0,776**
Vainqueur Deft			0,784
Moyenne Deft			0,664

Tableau 5. *Résultats de la catégorisation du corpus de test de Deft07-jeuxvidéo*

à chacun des systèmes pris individuellement (Torres-Moreno *et al.*, 2007). Par conséquent, il devient difficile d'améliorer encore significativement les résultats.

4.1.3. *Exemples de chaînes*

D'une manière générale, toutes classes confondues, on trouve en grande partie des expressions typiques du domaine « *afin-de-gagner* », « *tirer-sur-tout-ce-qui-bouger* » ainsi que d'autres servant à exprimer une opinion, une comparaison « *force-être-de-constater-que* » et des expressions se rapportant à des points précis qui portent à discussion, par exemple le ressenti dès la prise en main du jeu « *dès-le-premier-minute-de-jeu* », qui peut être positif (apparaît 12 fois dans les textes de la classe positif) ou non (présent 1 fois dans les textes exprimant une opinion négative et 16 fois dans les critiques mitigées), ou sur le déroulement du jeu « *au-fur-et-à-mesure-de-votre-progression* ». Certaines de ces expressions, évoquant par exemple un type de jeu « *jeu-de-hockey* », peuvent être quantifiées (on calcule leur nombre d'occurrences dans les différentes classes) et permettent de tirer des conclusions sur les types de jeu que les internautes préfèrent. Ainsi, la chaîne « *jeu-de-rôle* » a été retournée pour la classe positive et neutre, mais n'apparaît pas pour la classe négative.

Parmi les remarques positives, on trouve des expressions se référant à l'appréciation globale : « *absolument-superbe* », « *aller-être-aux-ange* », « *ce-qui-se-faire-de-mieux* », « *l'un-des-meilleur-jeu* » ; des expressions générales portant sur des points précis : « *le-intérêt-du-jeu* », « *le-point-fort-de* » ; des opinions exprimées sur des points précis, comme le genre du jeu : « *aimer-le-genre* », « *aimer-le-style* » ; des points techniques particuliers : « *graphisme-particulièrement-soigner* », « *jouabilité-exemplaire* » ; avec juste des accroches permettant de repérer les avis : « *ambiance-sonore-être* », « *angle-de-caméra-être* », « *au-niveau-de-son-gameplay* » ; le scénario

du jeu ainsi que sa durée sont beaucoup discutés : « *tenir-le-joueur-en-haleine* », « *le-richesse-des-possibilité* », « *un-durée-de-vie-conséquent* ». Il y a tout de même dans la classe positive des expressions quelque peu négatives, des nuances, provenant du fait que cette classe englobe les appréciations les plus positives (notes de 15 à 20) : « *assez-décevant* », « *parfois-pénible* ».

Parmi les expressions rattachées à la classe négative, on trouve les pendants des expressions précédentes, c'est-à-dire des appréciations générales : « *avoir-déjà-vu-beaucoup-mieux* », « *l'un-des-plus-mauvais* » ; des points précis : « *beaucoup-trop-répétitif* » ; des accroches (que l'on peut utiliser pour extraire des corpus les mots précédents ou suivants) : « *le-pire-,-ce-être-que* », « *ne-avoir-aucun-intérêt* » ; et un certain nombre d'expressions temporelles (pour exprimer le fait que l'on se lasse du jeu au bout de dix minutes par exemple) : « *au-bout-de-dix-minute* », « *au-bout-de-un-heure* » ; on note que les internautes sont assez imaginatifs lorsqu'il s'agit de dire du mal et qu'ils ne mâchent pas leurs mots : « *bouillie-de-pixels* », « *complètement-idiot* », « *complètement-raté* », « *grand-n'importe-quoi* », « *foncièrement-mauvais* », « *être-un-calamité* », « *gros-daube* » ; même si l'on trouve tout de même des nuances, permettant de ne pas être totalement négatif : « *le-seul-point-positif* », « *quelque-bon-idée* ». Le système extrait par ailleurs certaines expressions que l'on n'aurait peut-être pas pensé à chercher s'il avait fallu extraire manuellement ce type de chaînes : « *rose-bonbon* » est considéré comme péjoratif et n'apparaît pas dans la classe positive.

Parmi les chaînes extraites à partir des critiques neutres, on trouve un certain nombre d'expressions vues dans une des deux autres classes, des expressions positives, soit très positives : « *bouder-pas-notre-plaisir* », « *ce-qui-faire-son-charme* », « *de-fort-beau-manière* », « *en-avoir-pour-votre-argent* » ; soit légèrement positives : « *tout-à-fait-honorable* », « *se-en-sortir-relativement-bien* » ; des expressions négatives : « *un-certain-lassitude* », « *à-la-limite-du-supportable* » ; des expressions qui peuvent à la fois être négatives ou positives selon le contexte : « *en-passer-et-des-meilleur* », « *ce-ne-être-pas-non-plus* » ; et énormément d'expressions de nuances : « *on-pouvoir-regretter* », « *ce-être-de-autant-plus-dommage-que* », « *certes-sympathique-,-mais* ».

On observe des ensembles d'expressions correspondant à certains patrons linguistiques, ainsi dans la classe positive, on voit apparaître un certain nombre de chaînes ayant les premiers mots en commun : « *aussi-beau* », « *aussi-convaincant* », « *aussi-fun* », « *aussi-grandiose* », « *aussi-réussir* »... ; ou encore « *aucun-problème* », « *aucun-reproche* », « *aucun-souci* »... ; les adverbes servant souvent à introduire une appréciation : « *parfaitement-crédible* », « *parfaitement-dans-le-ton* », « *parfaitement-doser* »... ; « *très-bon-facture* », « *très-bon-jeu* », « *très-bon-moment* »... . Le fait que ces extractions soient ici automatiques et qu'on ne cherche pas à partir de patrons prédéfinis, permet de faire ressortir des chaînes auxquelles on n'aurait pas forcément pensé, ainsi on en trouve un certain nombre commençant par le mot « *digne* » : « *digne-de-ce-nom* », « *digne-de-intérêt* », « *digne-des-plus-grand* », « *digne-successeur* ».

Dans les textes de la classe négative, on retrouve des types de patrons communs à la classe positive, comme ceux ayant « *très* » pour base : « *très-banal* », « *très-*

classique », « *très-gênant* », « *très-laid* », « *très-mauvais* »... ; mais surtout des patrons propres à cette classe, comme ceux commençant par des adverbes que l'on ne retrouve pas dans la classe positive : « *totalement-absent* », « *totalement-creux* », « *totalement-inintéressant* », « *totalement-injouable* »... ; « *peu-convaincant* », « *peu-crédible* », « *peu-de-intérêt* », « *peu-de-plaisir* », « *peu-maniable* », « *peu-probant* »... ; et enfin des noms ou verbes exprimant des défauts : « *problème-de-caméra* », « *problème-de-collision* », « *problème-de-jouabilité* »... ; « *manque-cruel* », « *manque-flagrant* », « *manque-total* », « *manquer-cruellement-de-intérêt* », « *manquer-de-fluidité* »...

Les patrons que l'on retrouve fréquemment dans la classe neutre, peuvent soit être de même type que ceux que l'on a pu trouver dans les autres classes, soit s'appuyer sur un modèle typique de cette classe : « *loin-d'-être-au-top* », « *loin-d'-être-convaincant* », « *loin-d'-être-mauvais* », « *loin-d'-être-parfait* », « *loin-d'-être-à-la-hauteur* »... ; ou « *relativement* », que l'on ne retrouve pas dans les autres classes : « *relativement-bien* », « *relativement-convaincant* », « *relativement-correct* », « *relativement-pauvre* »... On observe que pour cette classe, un même patron peut correspondre à la fois à des expressions positives et négatives.

Finalement, on trouve quelques expressions typiques (du domaine et du mode d'expression), surprenantes : « *de-le-mort-qui-tuer* » (positif), « *réduire-à-la-portion-congrue* » (neutre) ou « *tirer-son-épingle-du-jeu* » (jeu de mots ?).

On observe qu'une même chaîne peut se retrouver dans plusieurs catégories, voire dans toutes. Il serait possible, par soustraction d'ensembles, de ne faire émerger que les chaînes qui n'apparaissent que dans les textes d'une seule des catégories, mais cela ne nous intéresse pas, car pour les utilisations que nous en faisons, il peut être utile de faire ressortir des expressions qui apparaissent dans plusieurs catégories :

– dans les cas où le corpus contient plus de deux opinions (par exemple positif, négatif et neutre), cela permet de visualiser des expressions qui apparaissent dans plusieurs catégories (une remarque que l'on retrouve fréquemment dans les critiques positives et neutres) ;

– cela offre la possibilité de quantifier les occurrences d'une expression dans les différentes catégories : si l'expression est le nom d'un produit, on peut connaître la tendance des avis sur ce produit, ou s'il s'agit d'une des caractéristiques de ce produit (par exemple « *le scénario du film* »), voir si celui-ci est plutôt discuté dans les bonnes ou les mauvaises critiques ;

– en cas d'utilisation de ces chaînes pour une tâche de catégorisation automatique de textes, une expression occurrant dans deux classes sur trois reste discriminante (elle permet d'écarter une des trois classes).

4.2. *NPS07-09*

L'enquête *Net Promoting Score* est effectuée auprès de clients du segment EDF Entreprise ayant eu un contact récent avec EDF, pour l'un des motifs suivants : récla-

	Détracteurs	neutres	Promoteurs	Total
Apprentissage	2 632	5 515	1 882	10 029
Développement	282	520	198	1 000
Test	244	617	234	1 095
Total	3 158	6 652	2 314	12 129

Tableau 6. *Répartition du nombre d'entretiens pour le corpus NPS07-09*

mations (réclamations techniques et réclamations commerciales), demande de mise en service suite à un raccordement, contacts sortants (ventes de nouvelles offres), autres demandes (hors raccordement). Des questions sont posées par des opérateurs à certains de ces clients pour mesurer leur intention de recommander EDF à des proches (note sur une échelle de 1 à 10) et évaluer les raisons de la note de satisfaction ainsi attribuée. Les réponses sont retranscrites par l'opérateur au cours de l'entretien. Les notes attribuées sont utilisées pour calculer un score de satisfaction des clients : l'entreprise considère comme *détracteurs* (respectivement *promoteurs*) les clients ayant donné une note de recommandation < 6 (respectivement > 8). Le score de satisfaction (dit « NPS ») est calculé en soustrayant le pourcentage de clients *détracteurs* au pourcentage de clients *promoteurs*. On peut ainsi suivre l'évolution de ce score dans le temps. Nous disposons des transcriptions de 12 124 réponses, recueillies au cours des années 2007 à 2009, réparties comme suit : 5 068 pour l'année 2007, 5 961 pour l'année 2008, 1 095 pour l'année 2009.

Nous avons associé chacun de ces entretiens à une des trois catégories *détracteur, promoteur, neutre* en fonction de la note attribuée par le client, en respectant la consigne métier permettant d'identifier les *détracteurs* et *promoteurs* et en rajoutant une classe *neutre* correspondant aux cas où la personne interrogée a donné une note comprise entre 6 et 8 inclus. Nous avons découpé ce corpus en apprentissage, développement et test selon la chronologie : les entretiens de l'année 2009 servent de test ; le corpus de développement est constitué de 1 000 entretiens de l'année 2008 ; l'apprentissage, des autres entretiens de 2008 (4 961 entretiens) et de ceux de 2007. Ce choix entraîne une sous-estimation des capacités des classifieurs par rapport à ce qui se pratique généralement dans le domaine : en effet, une répartition aléatoire des documents dans les corpus d'apprentissage et test permet d'intégrer implicitement dans l'apprentissage des éléments de nouveauté entrant en jeu dans l'évolution des opinions. La répartition des entretiens selon les catégories est présentée dans le tableau 6. Nous présentons dans le tableau 7 quelques statistiques supplémentaires sur ce corpus. Nous pouvons ainsi observer que, même si le corpus contient plus d'individus (12 000 au lieu de 4 000) que pour Deft07-jeuxvidéo, les interventions sont ici 30 fois plus courtes. La taille du vocabulaire est par ailleurs 7 fois plus réduite pour le NPS07-09.

Ce corpus possède d'autres particularités qui rendent son traitement automatique intéressant, la première étant le fait que la note attribuée par le client soit très subjective : chacun note la prestation selon sa propre échelle de valeur. Le fait que l'on ne recherche pas la note exacte mais une tranche compense en partie ce problème, mais cela ne suffit pas toujours. Ainsi on trouve des clients n'ayant rien à reprocher mais

Nombre de mots total	576 753
Nombre de mots uniques	10 903
Nombre de lemmes uniques	6 425
Nombre moyen de mots par document	47,6
Nombre de mots du plus petit document	1
Nombre de mots du plus grand document	134

Tableau 7. *Statistiques sur le corpus NPS07-09*

qui ne veulent pas mettre 10, par exemple : « *On ne peut pas mettre une note maximale. On n'est jamais satisfait à 100 % pour l'entreprise cela permet de se remettre en question.* » Ce client a attribué la note 7 et est ainsi entré dans la catégorie neutre, alors qu'il n'avait rien à reprocher. On note aussi un certain nombre de réponses ne correspondant pas exactement à l'enquête (clients jugeant EDF sur un point différent de la dernière prestation). D'autre part, bien qu'il s'agisse de conversations transcrites manuellement, ce corpus est issu de langage parlé (l'objectif étant, à terme, de travailler directement sur des sorties de système de reconnaissance automatique de la parole) : on ne peut pas prendre en compte la ponctuation alors que cela peut aider à la classification dans certains cas et les réponses sont synthétiques, à l'inverse des opinions exprimées dans les forums ou blogs, où les internautes peuvent insister en répétant plusieurs fois la même chose avec des formulations différentes, ce qui donne plus de chance au classifieur de trouver des mots discriminants les catégories. Nous avons utilisé TreeTagger (Schmid, 1994) pour effectuer la lemmatisation et l'analyse morphosyntaxique de ce corpus. Comme pour les expériences précédentes, nous avons jugé qu'il fallait conserver tous les mots composant les documents.

4.2.1. *Développement*

Dans le cas du cosinusBasique, les paramètres de filtre de collocations [50 ;4] sont les plus optimaux. Dans le cas du cosinusGini, on a retenu les paramètres :

– [30 ;5] qui apporte, à l'itération 2, le meilleur taux de classification que l'on ait réussi à avoir sur ce corpus ;

– [75 ;5] qui n'atteint pas la performance établie au point précédent, mais possède l'avantage de converger vers un F-score plus élevé que celui obtenu aux dernières itérations avec le filtre précédent.

Les F-scores associés à ces expériences sont représentés graphiquement en figure 2. Le couple cosinusBasique + chaînes n'est pas vraiment adapté au corpus. Dans le cas du cosinusGini, les résultats sont plus intéressants : si le gain en F-score n'est pas forcément convaincant, il correspond à un gain plus important en nombre de documents correctement étiquetés (555 documents correctement étiquetés sur 1 000 à l'itération 1, 591 documents à l'itération 2 (soit + 3,6 %)) ; la courbe « Gini [75 ;5] » se stabilise à 584 documents correctement étiquetés (+ 2,9 %). On en déduit donc que les chaînes aident en réalité à étiqueter des documents appartenant à la classe la plus fréquente, c'est-à-dire la neutre. Ceci nous est confirmé par l'étude des matrices de

Réf | Hyp	DET	NEU	PRO
DET	220	51	11
NEU	142	207	171
PRO	19	51	128

Réf | Hyp	DET	NEU	PRO
DET	223	48	11
NEU	116	242	162
PRO	14	65	119

Tableau 8. *Matrices de confusion pour les itérations 1 (à gauche) et 8 (à droite) de la catégorisation par cosinusGini pour le filtre [75 ;5], pour les trois classes* détracteur *(DET),* neutre *(NEU) et* promoteur *(PRO)*

confusion issues de ces classifications (présentées en tableau 8). Cela est dû au fait que cette classe est bien plus présente dans le corpus que les autres. Le calcul de collocations fait donc ressortir bien plus de chaînes spécifiques à cette classe, qui aident ainsi à bien la reconnaître. On peut au premier abord penser qu'il y a peu d'intérêt à mieux reconnaître la classe neutre, sauf à considérer que cette catégorisation peut être une première étape d'une classification en deux temps : dans un premier temps, on chercherait à reconnaître tous les documents appartenant à la classe neutre (on range les documents dans deux catégories : neutre ou autre), puis on effectuerait une nouvelle catégorisation des documents autre en positif ou négatif. Ceci peut être efficace si notre système est performant dans la différenciation de la classe positive et de la classe négative. Pour le savoir, nous avons retiré du corpus tous les documents appartenant à la classe neutre et avons effectué une catégorisation des documents restants en positif ou négatif. On obtient en effet un taux de bonne classification de 90 %.

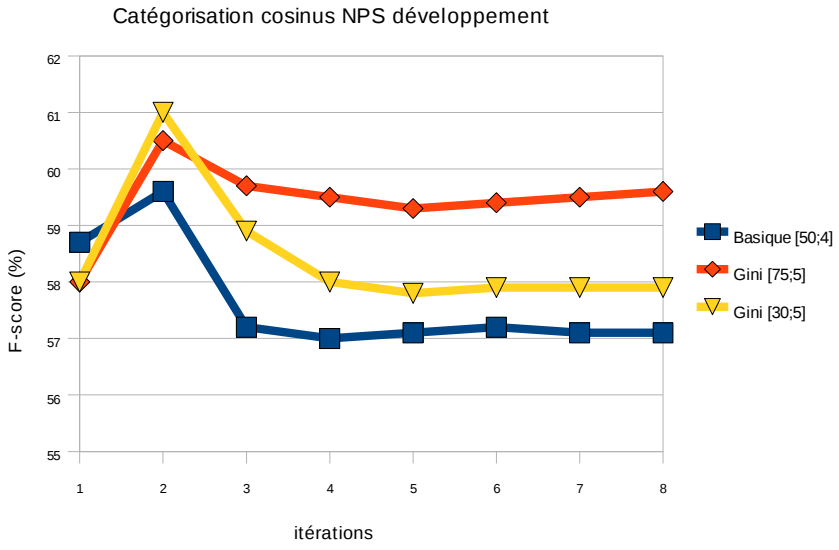

Figure 2. *F-scores des catégorisations par cosinus sur le corpus de développement de NPS07-09*

Classifieur	Comptage	Chaînes	ϵ	C	F-score
SVM	TF	sansChaînes	2^2	2^0	0,581
SVM	TF	chaînesMaxBasique	2^{-2}	2^5	0,554
SVM	TF	chaînesMaxGini	2^3	2^5	0,551
SVM	TF	meilleuresChaînesBasique	2^{-4}	2^3	0,557
SVM	TF	meilleuresChaînesGini	2^3	2^3	0,557
SVM	binaire	sansChaînes	2^3	2^{-2}	0,533
SVM	binaire	chaînesMaxBasique	2^{-3}	2^{-3}	0,524
SVM	binaire	chaînesMaxGini	2^{-15}	2^{-4}	0,541
SVM	binaire	meilleuresChaînesBasique	2^{-1}	2^{-4}	0,539
SVM	binaire	meilleuresChaînesGini	2^3	2^{-5}	0,540

Tableau 9. *F-scores obtenus sur la catégorisation du corpus de développement de NPS07-09 par des SVM, ainsi que les paramètres optimaux C et ϵ trouvés*

Les résultats de catégorisation par SVM pour trois classes sont présentés dans le tableau 9. Les résultats avec repli sont soit inférieurs à ceux sans repli, soit identiques. On observe que les SVM ne donnent pas de meilleurs résultats que les cosinus et que le meilleur F-score pour les SVM est obtenu sans chaînes.

4.2.2. *Test*

Nous avons étudié l'impact des chaînes sur les catégorisations par cosinus, en testant les trois filtres de sélection de collocations qui avaient donné les meilleurs résultats sur le développement. Nous avons testé chacun de ces filtres à la fois sur le cosinusBasique et le cosinusGini. Les F-scores ainsi obtenus sont présentés en figure 3. On remarque que la correspondance filtre optimal/classifieur n'est pas forcément la même entre le développement et le test. Globalement, l'ensemble de ces expériences converge vers le même résultat : F-score entre 61 et 62 %, soit entre 2 et 3 % de mieux que sans utilisation de chaînes, ce qui correspond par exemple pour le cosinusGini avec filtre à [30 ;5] à passer de 597 à 665 documents correctement étiquetés sur 1 095. Pour ce qui est des SVM, nous avons réalisé les tests présentés dans le tableau 10, avec et sans chaînes (les plus grandes obtenues), en TF et binaire, sans repli puisque les expériences avec sur le développement n'ont rien donné.

Les catégorisations avec cosinus sont ici meilleures que celles par SVM. Ceci s'explique par l'utilisation du corpus de développement pour apprendre les modèles, ce qui, comme nous l'avons vu pour le corpus Deft07-jeuxvideo, améliore de quelques pour-cent les résultats des cosinus mais pas vraiment ceux des SVM. Ces résultats sont encore quelque peu améliorés par la prise en compte des chaînes. L'utilisation des chaînes pour les SVM n'améliore pas les résultats de la catégorisation. Les expériences de catégorisation en *promoteur/détracteur* uniquement confirment les résultats observés sur le corpus de développement : on est capable, avec chaînes (filtre [50 ;4]) de classifier correctement 90 % des textes dans ces deux classes avec un cosinusGini (87 % sans chaînes).

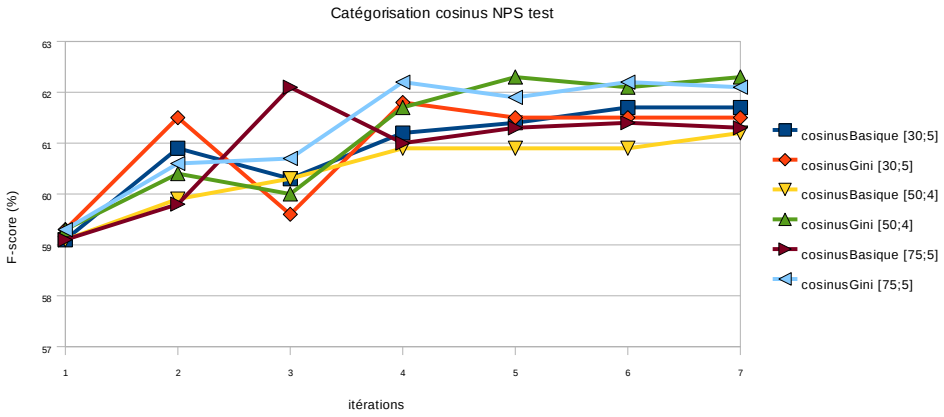

Figure 3. *F-scores des catégorisations par cosinus sur le corpus de test de NPS07-09*

Classifieur	Comptage	Chaînes	Filtre	F-score
SVM	TF	sansChaînes, avec DEV		0,579
SVM	binaire	sansChaînes, avec DEV		0,565
SVM	TF	chaînesMaxBasique, avec DEV	[50 ;4]	0,555
SVM	binaire	chaînesMaxBasique, avec DEV	[50 ;4]	0,561
SVM	TF	chaînesMaxGini, avec DEV	[75 ;5]	0,535
SVM	binaire	chaînesMaxGini, avec DEV	[75 ;5]	0,582
CosinusBasique		sansChaînes avec DEV		0,591
CosinusGini		sansChaînes avec DEV		0,593
CosinusBasique		avec DEV	[50 ;4]	0,612
CosinusGini		avec DEV	[50 ;4]	0,623

Tableau 10. *F-scores obtenus sur la catégorisation du corpus de test de NPS07-09 par les différentes méthodes présentées précédemment*

4.2.3. *Exemples de chaînes*

Nous présentons ici quelques-unes des 4304 chaînes uniques (6 661 chaînes en tout, certaines pouvant être communes à plusieurs classes) obtenues après 7 itérations de la méthode décrite en section 2.1, avec les paramètres de filtre [50 ;4]. Malgré le filtre plus laxiste que celui qui a mené aux exemples de la section 4.1.3, le système retourne beaucoup moins de chaînes car, en raison de la petite taille du corpus, il est plus rare de trouver des séquences qui se répètent. De plus, en raison de l'inégale répartition du nombre de documents par catégorie, les chaînes retournées ne sont pas également réparties entre les différentes classes (52 % des chaînes proposées sont apparentées (non exclusivement) à la classe neutre, qui représente 54 % des documents de l'apprentissage).

D'une manière générale, de par sa définition, la classe neutre recouvre un certain nombre de chaînes que l'on ne trouve que dans l'une ou l'autre des deux autres catégories : « *accueil-téléphonique* » dans la classe *promoteur*, « *appeler-plusieurs-fois-pour* » dans la catégorie des *détracteurs*. Nous ne présentons pas de détail sur la classe neutre, étant donné que cela est peu pertinent pour l'utilisation qui en est faite (on cherche ce que les clients ont à reprocher à l'entreprise ou ce qu'ils en disent de positif). Plus généralement, nous avons observé beaucoup de variantes sur un même sujet, car chaque client interrogé l'exprime à sa façon ; ainsi on trouve une vingtaine de chaînes exprimant le fait qu'il n'y ait pas de problème particulier avec l'entreprise ou l'intervention ayant conduit au rappel : « *avoir-pas-avoir-de-problème* », « *avoir-pas-de-souci* », « *je-n-avoir-jamais-avoir-de-problème* », « *tout-s-être-bien-dérouler* »... Un certain nombre de variantes sont purement linguistiques. « *résolution-du-problème* », « *résoudre-le-problème* », « *régler-le-problème* », ou encore la différence entre l'emploi du pronom « il » ou « elle » pour des chaînes identiques selon que l'employé d'EDF était un homme ou une femme, ou du pronom « je » ou « on » quand le client parle de son expérience : « *je-être-content* », « *je-être-satisfaire* », « *on-être-content* », « *on-être-satisfaire* ».

On note par ailleurs que notre méthode retourne des expressions qu'il aurait pu être possible de chercher à partir d'une méthode symbolique, avec des patrons de chaînes, comme c'est actuellement le cas dans l'entreprise ; par exemple les éléments suivants de la classe *promoteur* auraient pu être extraits à partir d'une règle de type « très + ADJECTIF [+ NOM] » ou « très + ADVERBE + VERBE » : « *très-accueillant* », « *très-agréable* », « *très-aimable* », « *très-bien-accueillir* », « *très-bien-comprendre* », « *très-bien-expliquer* », « *très-bien-passer* », « *très-bien-renseigner* », « *très-bon-contact* », « *très-compétent* », « *très-cordial* », « *très-courtois* », « *très-gentil* », « *très-professionnel* », « *très-sympathique* ». Notre méthode permet ainsi de s'affranchir de l'écriture manuelle et coûteuse de telles règles.

Bien évidemment, un même sujet peut se retrouver dans les deux catégories, avec des avis différents. Il en va ainsi des chaînes se rapportant à l'interlocuteur privilégié (expression métier désignant le fait de n'avoir qu'un unique interlocuteur dans l'entreprise) : « *interlocuteur-privilégier* », « *interlocuteur-unique* ». Les remarques temporelles sont aussi évidemment très subjectives, elles dépendent du problème et de la personne. Nous les retrouvons ainsi dans deux catégories : « *très-longtemps* », « *cela-avoir-être-très-long* », « *le-délai-d-intervention* », et à l'inverse : « *avoir-répondre-rapidement* », « *avoir-être-faire-rapidement* », « *cela-avoir-être-très-rapide* », « *dans-le-jour-qui-avoir-suivre* », « *dans-le-semaine* », « *en-temps-et-en-heure* », « *quelque-jour-après* », « *rapide-et-efficace* », « *le-rapidité-de-le-réponse* ». Certaines chaînes portent sur des actes précis « *pour-un-augmentation-de-puissance* », « *pour-un-duplicata* »... et peuvent être quantifiées et comparées entre les deux classes, permettant ainsi de savoir si cet acte se passe généralement bien ou pose souvent des problèmes aux clients.

On trouve aussi quelques expressions servant à nuancer les réponses, par exemple chez les *détracteurs* « *agréable-mais* », ou chez les *promoteurs* « *par-contre* ».

4.3. *Movies Polarity Dataset v2.0*

Dans la mesure où nos méthodes sont probabilistes et donc relativement indépendantes de la langue, aussi bien pour l'extraction de chaînes que pour la catégorisation, nous avons choisi de les tester sur un corpus en langue anglaise. Un corpus couramment utilisé pour effectuer de la classification de textes en opinions est le *Movies Polarity Dataset v2.0*, introduit dans (Pang *et al.*, 2002) et complété dans (Pang et Lee, 2005). Il est constitué de 1 000 critiques positives et 1 000 critiques négatives écrites par des internautes à propos de films. Les 600 premières critiques de chacune des deux classes sont utilisées comme corpus d'apprentissage, les 100 suivantes en tant que développement et les 300 dernières servent de corpus de test. Le fait que le corpus comporte peu de documents nous a poussés à n'utiliser que 200 critiques pour le développement, ce que nous avons considéré comme suffisant pour adapter seulement deux paramètres des SVM. Le corpus a été lemmatisé avec LiaTagg, nous avons conservé tous les mots, ponctuations et symboles. La petite taille des corpus de développement (200 critiques) et test (600 critiques) rend les écarts de résultats difficilement significatifs et l'application des chaînes de mots réduite. Nous avons donc tenté d'extraire des collocations avec des filtres beaucoup plus permissifs que dans les expériences précédentes, qui retournent plus de chaînes et de moins bonne qualité, mais offrent ainsi une plus grande probabilité qu'elles se retrouvent dans le test.

4.3.1. *Développement*

Nous avons retenu les filtres :

– [40 ;3] qui a donné les meilleurs résultats avec le cosinusBasique (85,5 % avec les mots isolés, 88 % dans le meilleur des cas (itération 4), 87,5 % à la dernière itération) ;

– [100 ;10] qui a donné les meilleurs résultats avec le cosinusGini (85 % sans chaîne, 89 % à partir de l'itération 4).

Pour les SVM, on observe des résultats allant de 85 à 90 %. Les scores sans chaînes sont déjà assez bons : 88 % en TF, 86 % en binaire. Le meilleur résultat (90 %) est obtenu dans le cas où on applique les chaînes apprises avec le filtre [100 ;10] avec les valeurs en binaire, aussi bien celles de l'itération 4 que les maximales : 90 % en binaire, 89,5 % en TF, ce qui correspond aussi à ce que l'on a obtenu avec le cosinusGini pour ce même filtre, c'est-à-dire des résultats stabilisés à leur meilleur niveau à partir des chaînes de l'itération 4.

4.3.2. *Test*

Sur le corpus de test, les chaînes n'aident pas à améliorer les résultats de catégorisation, le gain le plus important que l'on observe étant de 5 documents mieux classés sur 600 dans le cas du cosinusGini avec le filtre [40 ;30]. Les SVM offrent ici les meilleures performances, soit 87,8 % (527 critiques) en binaire sans application de chaînes (83,8 % en TF). Les quatre autres expériences avec des SVM (avec les chaînes maximales du filtre [40 ;3], celles du filtre [100 ;10], en TF et en binaire) donnent des

résultats allant de 82,4 % (494 critiques) à 85,7 % (514 critiques). Nous ne considérons pas que ces faibles écarts soient réellement significatifs.

Les résultats que nous obtenons sur ce corpus sont proches de l'état de l'art. En comparaison, dans (Pang et Lee, 2004), les auteurs ont obtenu 86,4 % avec un classifieur bayésien naïf aidé d'un système de détection de sujectivité entraîné sur un autre corpus et 87,2 % avec des SVM. Ils ont utilisé une validation croisée en 10 sous-parties sur l'ensemble du corpus, ce que nous avons choisi de ne pas faire pour les raisons évoquées en section 3. À notre connaissance, (Abbasi *et al.*, 2008) ont obtenu les meilleurs résultats sur ce corpus en utilisant des SVM aidés d'indices syntaxiques et stylistiques et un algorithme génétique basé sur l'entropie pour la sélection des dimensions. Ils ont obtenu 91,7 % en validation croisée (10 sous-parties) et 91,5 % en *bootstrapping* (le test est effectué sur 100 critiques tirées aléatoirement, 50 fois). L'ajout des indices stylistiques (indices lexicaux au niveau du mot ou du caractère, mesures de richesse du vocabulaire, distribution des mots selon leur taille, nombre moyen de mots par phrase, etc.) ayant entraîné un gain de 4 %, nous envisageons de les intégrer à notre méthode afin d'étudier leur impact combiné à celui des chaînes. D'une manière générale, pour avoir une meilleure idée de nos performances par rapport à celles présentées dans ces articles, il serait intéressant de connaître la manière exacte dont les corpus sont répartis entre apprentissage et test pour chacune d'entre elles.

4.3.3. *Exemples de chaînes*

Parmi les 8 660 chaînes uniques (5 571 négatives et 6 019 positives) extraites du corpus d'apprentissage avec le filtre [40 ;3], on trouve, comme pour les corpus précédents, un certain nombre d'expressions communes aux deux catégories, par exemples celles servant à exprimer une opinion « *in-my-mind* », « *in-my-opinion* », « *because-of-the-fact-that* », des expressions du domaine annonçant une opinion. « *first-twenty-minutes* », « *the-dialogue-be* », ou celles présentant des genres « *black-comedy* », « *romantic-comedy* », « *horror-movies* », qui peuvent être quantifiées. On trouve ensuite des expressions d'opinion sur l'ensemble du film négatives « *a-huge-disappointment* », « *a-pretty-bad-movie* », ou positives : « *a-great-film* », « *one-of-the-best-movies* », « *of-the-best-films* », « *best-films-of-the-year* », « *good-movie* ». Des critiques négatives sur des points précis « *difficult-to-follow* », « *bad-acting* », « *bad-dialogue* », et positives « *a-hilarious* », « *perfectly-cast* », « *plenty-of-laughs* ».

Parmi les autres expressions négatives « *attempts-at-humor* », « *be-a-shame* », « *be-even-worse* », « *easily-the-worst* », « *biggest-disappointment* », « *the-most-irritating* », « *problem-be-that* », « *try-to-convince* », « *waste-your-time* », « *with-the-exception-of* », « *bad-enough* », « *unintentionally-hilarious* ». Et parmi les positives « *a-good-time* », « *a-incredible-job* », « *absolutely-breathtaking* », « *as-one-of-the-best* », « *attention-to-detail* », « *be-one-of-the-most* », « *be-absolutely-brilliant* », « *be-excellent* », « *be-probably-the-best* », « *brilliant-performance* », « *definitely-recommend* », « *have-a-good-time* », « *not-disappoint* », « *one-of-the-greatest* »,

« *pleasantly-surprise* », « *pure-entertainment* », « *summer-blockbuster* », « *the-most-famous* » avec toutefois des nuances « *biggest-problem* », « *a-little-too-long* ».

On trouve quelques noms de films n'apparaissant que dans des critiques positives, laissant penser que peu de personnes en ont dit du mal « *nightmare-before-christmas* », « *forrest-gump* » ou d'autres qu'en négatif « *batman-and-robin* ».

Enfin, on trouve beaucoup de réalisateurs, scénaristes ou acteurs, dont la plupart sont mentionnés dans les deux catégories, mais il y en a cependant certains que l'on ne retrouve que dans les critiques négatives « *andrew-kevin-walker* » (qui a travaillé sur *batman*, confirmant ainsi l'exemple précédent), « *angelina-jolie* », « *dennis-hopper* », « *dennis-rodman* », « *donald-sutherland* », « *jean-reno* », « *jeff-daniel* », « *jeff-goldblum* », « *jenna-elfman* », « *jennifer-aniston* », « *jennifer-esposito* », « *jennifer-love* » et ceux dont on ne dit jamais de mal « *helena-bonham-carter* », « *harvey-keitel* », « *george-lucas* », « *alfred-hitchcock* », « *francis-ford-coppola* ».

5. Discussions

La méthode décrite en section 2.1 permet d'extraire d'un corpus des chaînes de mots relatives aux différentes opinions exprimées dans celui-ci. Ces chaînes offrent la possibilité de visualiser des expressions exprimant une opinion, soit directement (« *avoir-vraiment-rien-de-extraordinaire* »), soit sous forme d'accroche (« *force-être-de-constater-que* »). Ceci peut permettre d'extraire les remarques fréquemment exprimées ou de visualiser dans un document les passages exprimant des opinions (par exemple en les surlignant). Les questions soulevées par l'extraction de ces chaînes portent sur l'adaptation automatique aux corpus du filtre de sélection des collocations intéressantes : est-il possible de trouver automatiquement un filtre en fonction des paramètres du corpus (nombre de classes, de documents, taille des documents, type de corpus (issu de l'oral, écrit formaté, écriture de type blogs)...) ? Faut-il que ce filtre s'adapte aux changements de taille du corpus (quand on ajoute des documents à l'apprentissage par exemple) ? Faut-il un filtre différent pour les différentes classes si la répartition des documents dans les classes n'est pas équilibrée ?

Par ailleurs, un des objectifs de ces travaux étant d'être appliqué directement sur des corpus issus de reconnaissance automatique de la parole, nous travaillons à l'adaptation de la méthode à ce type de corpus : prise en compte des disfluences par la recherche de chaînes constituées de mots non exclusivement contigus, pouvant prendre en compte des ajouts, suppressions ou substitutions de mots.

Pour ce qui est du filtre, nous optimisons à l'heure actuelle ses paramètres en étudiant l'impact des chaînes qu'ils permettent d'extraire sur les sorties de systèmes de catégorisation automatique de textes. Les chaînes que nous proposons permettent de prendre en compte des phénomènes tels que l'ironie (en désambiguïsant un mot par son contexte), la négation, les nuances... Leur intérêt pour la classification a été montré sur les trois corpus présentés dans cet article. En particulier, le couple cosi-

nusGini + chaînes qui obtient généralement de meilleurs résultats que tous les autres systèmes comparés. Dans tous les cas, ces chaînes présentent aussi un intérêt pour la catégorisation à base de SVM : leur utilisation avec des SVM à noyau linéaire permet, si elle n'améliore pas forcément les résultats, de réduire significativement le temps nécessaire à l'estimation des paramètres du noyau. On pourrait penser que le gain de temps observé n'est pas intéressant s'il faut tester des dizaines de filtres différents, comme cela a été fait ici, à part si :

– on les calcule de toute façon pour une autre raison (visualiser les chaînes relatives à chacune des opinions) ;

– on considère que certaines valeurs de filtre sont passe-partout, même si non optimales, par exemple [75 ;5] qui donne souvent des résultats assez bons pour la catégorisation ou encore si on dispose d'une méthode permettant d'automatiquement estimer les valeurs optimales de ce filtre, comme évoqué précédemment.

Pour améliorer la catégorisation, il serait intéressant de trouver une méthode de repli performante : les chaînes trop grandes sont peu représentées et deviennent inutiles pour le classifieur, voire peuvent diminuer leur intérêt pour la catégorisation. Il faudrait ainsi soit améliorer la méthode de recherche des chaînes, pour qu'elle ne propose pas d'agglutiner deux chaînes déjà fortement discriminantes, bien que cela s'écarte du premier intérêt (visuel) des chaînes, soit mettre en place une meilleure méthode de repli au niveau de la catégorisation. Prendre en compte des sous-chaînes (de plusieurs mots) plutôt que tous les mots isolés pourrait permettre de continuer à améliorer la catégorisation quand les chaînes deviennent trop grandes. Enfin, l'intérêt de ces chaînes peut être discuté vis-à-vis de méthodes prenant déjà implicitement en compte ce genre de patrons, par exemple les SVM à noyaux de mots (Cancedda *et al.*, 2003). Ces méthodes présentent à nos yeux plusieurs inconvénients.

1) Ces méthodes sont très coûteuses en temps de calcul : nous avons tenté d'effectuer des classifications avec des SVM à noyau de type sous-séquences de mots (Lodhi *et al.*, 2002), en utilisant l'implémentation (Kleedorfer et Seewald, 2005) fournie dans le package Weka (Hall *et al.*, 2009)/SMO (Platt, 1999). Nous avons testé les configurations suivantes : avec les paramètres $subsequenceLength = 2$, $maxSubsequenceLength = 4$, avec pour tâche de catégoriser les 1 694 critiques du corpus de test de Deft07-jeuxvideo, et une expérience réduite avec les paramètres $subsequenceLength = 1$, $maxSubsequenceLength = 2$, avec pour tâche de catégoriser les 508 critiques du corpus de développement de Deft07-jeuxvideo ; dans les deux cas la taille du cache a été fixée à 100 000 007 et on a attribué à chacune de ces expériences un processeur à temps plein et 6 Go de RAM à la machine virtuelle Java. Au bout de 20 jours, aucune de ces deux expériences n'est arrivée à terme.

2) Les chaînes prises en compte par ces méthodes sont implicites : il n'est pas possible de les visualiser, ce qui est l'intérêt premier de notre méthode, qui offre de plus la possibilité de savoir lesquelles de ces chaînes sont discriminantes (celles qui ont un critère de pureté de Gini élevé) et pertinentes pour la classification (par exemple en les ordonnant par $TF \times IDF \times pGini$).

6. Conclusion

Dans cet article nous avons présenté une méthode permettant d'extraire des chaînes de mots relatives à des opinions diverses exprimées sur des produits et services. Au-delà des améliorations à court terme que nous proposons dans le paragraphe précédent, cette méthode a notamment été éprouvée dans un contexte applicatif précis, celui de la gestion de la relation avec le client par EDF. Nous avons ainsi répondu à trois besoins industriels précis :

– la robustesse de l'approche vis-à-vis du corpus étudié. La méthode a été testée sur différentes langues (français et anglais) et sur différents modes d'expressions spontanées (oral pour les enquêtes de satisfaction et écrit pour les critiques de films et de jeux vidéo) ;

– la robustesse de l'approche vis-à-vis des classes d'opinion considérées (*promoteur*/neutre/*détracteur* pour les enquêtes de satisfaction, positif/neutre/négatif pour les critiques). La méthode peut donc être ajustée en fonction du contexte applicatif et laisse la possibilité à l'utilisateur de circonscrire la notion d'opinion en fonction de l'étude qu'il souhaite faire et ainsi construire lui-même son ensemble d'apprentissage en fonction des classes d'opinion définies ;

– l'extraction des chaînes relatives à chacune de ces classes permet à la fois d'avoir un aperçu des points les plus abordés, des remarques formulées les plus fréquemment. Elles peuvent être utilisées pour améliorer les performances des systèmes de catégorisation automatique de textes en opinion car elles présentent l'avantage d'être plus discriminantes que des mots pris isolément.

Bien plus, certaines de ces chaînes peuvent correspondre à des thématiques sur lesquelles les clients expriment le plus fréquemment une opinion donnée permettant ainsi de fournir un premier aperçu des cibles ou des objets des opinions exprimées. Cette méthode a également été mise en œuvre dans un contexte de classification de thématique (Lavalley *et al.*, 2010). Le croisement des opinions et des thématiques auxquelles elle réfère est un sujet de recherche qui, malgré sa complexité, mériterait à terme d'être traité.

Par ailleurs, nous aimerions également nous attaquer au problème de la détection de nouveauté et mesurer l'impact de l'application de cette méthode dans ce contexte. Quelles sont les thématiques et les opinions relatives qui apparaissent au fil du temps ? Quelles sont celles qui disparaissent ?

7. Bibliographie

Abbasi A., Chen H.-H., Salem A., « Sentiment Analysis in Multiple Languages : Feature Selection for Opinion Classification in Web Forums », *ACM Transactions on Information Systems*, 2008.

Bozzi L., Suignard P., Waast-Richard C., « Segmentation et classification non supervisée de conversations téléphoniques automatiquement retranscrites », *TALN*, Senlis, France, 2009.

Cailliau F., Giraudel A., « Enhanced Search and Navigation on Conversational Speech », *Proceedings of SIGIR 2008 SSCS Workshop*, Singapore, 2008.

Camelin N., Damnati G., Béchet F., Mori R. D., « Opinion mining in a telephone survey corpus », *International Conference on Spoken Language Processing (ICSLP 06)*, Pittsburg, PA, 2006.

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

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.

Damerau F., « Generating and Evaluating Domain-Oriented Multi-Word Terms from Texts », *Information Processing and Management*, vol. 29, n° 4, p. 433-447, 1993.

Danesi C., Clavel C., « Impact of Spontaneous Speech Features on Business Concept Detection : a Study of Call-Centre Data », *Workshop Searching Spontaneous Conversational Speech ACM Multimedia 2010*, 2010.

Dave K., Lawrence S., Pennock D. M., « Mining the Peanut Gallery : Opinion Extraction and Semantic Classification of Product Reviews », *WWW-2003*, Budapest, 2003.

Dias G., Carapinha L., Trindade R., Mota S., Dias J., « Construire et Accéder à une Base de Données d'Expressions Figées à partir des Ressources de la Toile », *TIA-2003*, p. 92-101, 2003.

Dias G., Vintar S., « Unsupervised Learning of Multiword Units from Part-of-Speech Tagged Corpora : Does Quantity Mean Quality ? », *in* C. Bento, A. Cardoso, G. Dias (eds), *Progress in Artificial Intelligence*, vol. 3808 of *Lecture Notes in Computer Science*, Springer Berlin/Heidelberg, p. 669-679, 2005.

Drouin P., « Spécificités lexicales et acquisition de la terminologie », *JADT*, p. 345-352, 2004.

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.

Frantzi K., Ananiadou S., Mima H., « Automatic recognition of multi-word terms : the C-value/NC-value method », *International Journal on Digital Libraries*, vol. 3, p. 115-130, 2000.

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) », *DEFT*, Grenoble, France, p. 1-8, 2007.

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. H., « The WEKA data mining software : an update », *SIGKDD Explor. Newsl.*, vol. 11, n° 1, p. 10-18, 2009.

Hsu C.-W., Chang C.-C., Lin C.-J., A Practical Guide to Support Vector Classification, Technical report, Department Of Computer Science, 2003.

Joachims T., « Text categorization with Support Vector Machines : Learning with many relevant features », *Machine Learning : ECML-98*, vol. 1398/1998, p. 137-142, 1998.

Kleedorfer F., Seewald A., Implementation of a String Kernel for WEKA, Technical Report n° TR-2005-13, Oesterreichisches Forschungsinstitut fur AI, Wien, Austria, 2005.

Kuznick L., Guènet A.-L., Peradotto A., Clavel C., « L'apport des concepts métiers pour la classification des questions ouvertes d'enquête », *TALN*, 2010.

Lavalley R., Clavel C., El-Bèze M., Bellot P., « Finding topic-specific strings in text categorization and opinion mining contexts », *The 2010 International Conference on Data Mining (DMIN'10)*, 2010.

Lodhi H., Saunders C., Shawe-Taylor J., Cristianini N., Watkins C. J. C. H., « Text Classification using String Kernels », *Journal of Machine Learning Research*, vol. 2, p. 419-444, 2002.

Moschitti A., Basili R., « Complex Linguistic Features for Text Classification : A Comprehensive Study », *Lecture Notes in Computer Science*, vol. 2997, p. 181-196, 2004.

Nallapati R., « Discriminative models for information retrieval », *SIGIR*, p. 64-71, 2004.

Ounis I., de Rijke M., Macdonald C., Mishne G., Soboroff I., « Overview of the TREC 2006 Blog Track », *Proceedings of TREC 2006*, Gaithersburg, USA, 2007.

Pang B., Lee L., « A sentimental education : Sentiment analysis using subjectivity summarization based on minimum cuts », *Proceedings of the ACL*, p. 271-278, 2004.

Pang B., Lee L., « Seeing stars : Exploiting class relationships for sentiment categorization with respect to rating scales », *Proceedings of the ACL*, p. 115-124, 2005.

Pang B., Lee L., Vaithyanathan S., « Thumbs up ? Sentiment Classification using Machine Learning Techniques », *EMNLP*, p. 79-86, 2002.

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 », *DEFT*, Grenoble, France, p. 9-17, 2007.

Patin G., « Unsupervised Chinese Lexicon Extraction on a Domain Specific Corpus : Method and Evaluation », *The 23rd International Conference on Computational Linguistics (COLING 2010)*, 2010.

Pearce D., « A Comparative Evaluation of Collocation Extraction Techniques », *LREC*, Las Palmas, Canary Islands, p. 1530-1536, 2002.

Platt J. C., « Fast training of support vector machines using sequential minimal optimization », vol. 1, MIT Press, Cambridge, MA, USA, p. 185-208, 1999.

Popescu A.-M., Etzioni O., « Extracting Product Features and Opinions from Reviews », *HLT/EMNLP*, Vancouver, p. 339-346, 2005.

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

Schmid H., « Probabilistic part-of-speech tagging using decision trees », *International Conference on New Methods in Language Processing*, Manchester, p. 44-49, 1994.

Sebastiani F., « Machine learning in automated text categorization », *ACM Computing Surveys*, vol. 34(1), p. 1-47, 2002.

Seretan V., Nerima L., Wehrli E., « Using the Web as a corpus for the syntactic-based collocation identification », *LREC*, Lisbon, Portugal, p. 1871-1874, May, 2004.

Smadja F., « Retrieving collocations from text : Xtract », *Computational Linguistics*, vol. 19, n° 1, p. 143-178, 1993.

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.

Spriet T., El-Bèze M., « Introduction of Rules into a Stochastic Approach for Language Modelling », *NATO ASI Series F Computer and Systems Sciences*, vol. 169, p. 350-355, 1999.

Thanopoulos A., Fakotakis N., Kokkinakis G., « Comparative Evaluation of Collocation Extraction Metrics », *Language Resource and Evaluation (LREC)*, Las Palmas, Canary Islands, 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 ? », *DEFT*, Grenoble, France, p. 119-133, 2007.

Turney P. D., « Thumbs Up or Thumbs Down ? Semantic Orientation Applied to Unsupervised Classification of Reviews », *ACL*, Philadelphia, p. 417-424, 2002.

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

Wilson T., Wiebe J., Hoffmann P., « Recognizing contextual polarity in phrase-level sentiment analysis », *HLT/EMNLP*, Vancouver, p. 417-424, 2005.

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