ACADÉMIE D'AIX-MARSEILLE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

THÈSE

présentée à l'Université d'Avignon et des Pays de Vaucluse
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : Informatique

École Doctorale 536 « Sciences et Agronomie »
Laboratoire d'Informatique (EA 4128)

Extraction automatique de segments textuels, détection de rôles, de sujets et de polarités

par
Rémi Lavalley

Soutenue publiquement le 31 février 20 devant un jury composé de :

Laboratoire d'Informatique d'Avignon - CERI
Centre d'Enseignement et de Recherche en Informatique

Résumé

Dans cette thèse, nous présentons de nouvelles méthodes permettant l'extraction de chaînes de mots (segments textuels) relatives à des catégories (thématiques, rôles des locuteurs, opinions). Nous proposons, dans un premier temps, une méthode basée sur une métrique de recherche de collocations, que nous appliquons de manière distincte sur les documents liés à la même catégorie et qui, par itérations, nous permet d'obtenir des chaînes caractéristiques de cette catégorie. Ces chaînes sont alors employées pour améliorer les performances de systèmes de catégorisation de textes ou dans un but d'extraction de connaissances (faire ressortir des éléments textuels tels que des expressions employées par un type de locuteurs, des sous-thématiques liées à la catégorie, ...). Nous proposons ensuite une seconde méthode permettant de rechercher, dans un corpus d'opinions, des n-grammes exprimant des jugements sur des sujets prédéfinis. Nous pouvons alors extraire des segments textuels représentant l'expression d'une opinion sur un des sujets cibles.

Ces méthodes sont validées par un certain nombre d'expériences effectuées dans des contextes différents : écrits de blogs, transcriptions manuelles de parole spontanée, critiques de produits culturels, enquêtes de satisfaction EDF, en français ou en anglais, ...

Table des matières

Chapitre 1

Introduction

L'Homme a, de manière innée, toujours cherché à économiser son énergie (son labeur, son temps) pour arriver à ses objectifs. C'est ce comportement que George Kingsley Zipf a théorisé en tant que principe du moindre effort. En réalité, cela ne consiste pas à éviter le "travail" mais à trouver un moyen plus facile d'arriver à un résultat équivalent : par exemple, en français courant, employer abondamment le verbe "faire", qui possède le double avantage d'être court et de disposer d'une multitude de définitions (moindre effort), à la place d'un de ses nombreux synonymes disponibles, n'empêche généralement pas l'interlocuteur de comprendre la phrase (objectif atteint). Il en va ainsi de la recherche dans un certain nombre de domaines. Si je prends quelques minutes de mon temps pour inventer le métro, je peux m'éviter de marcher quotidiennement (la balance reposant sur la mesure de l'effort demandé par chacune des options envisageables, qui est extrêmement subjective). Dans une certaine mesure, la recherche en Traitement Automatique de la Langue Naturelle Écrite s'inscrit dans ce précepte : bien sûr que l'on pourrait manuellement tirer les mêmes informations statistiques sur des corpus, mais pourquoi passer des semaines à parcourir 10 000 documents pour y compter les occurrences des mots les composant quand on peut automatiquement créer un index ? Pourquoi à nouveau lire 10 000 autres documents, pour recalculer les occurrences des mots et comparer ces nouvelles fréquences à celles obtenues sur l'ensemble précédent afin de rechercher les mots les plus caractéristiques de chacun des ensembles, alors que la procédure écrite pour l'étape précédente pourrait resservir ? Et puis finalement, pourquoi lire tous les textes disponibles sur Internet, y comparer les mots présents afin de les regrouper par catégorie, rechercher les phrases les plus pertinentes (à refaire selon l'objectif poursuivi), les traduire toutes dans la même langue et écrire un résumé, quand on peut apprendre à l'ordinateur à le faire ?

D'autre part, l'accroissement de la quantité de données disponibles va de pair avec les progrès en Traitement Automatique de la Langue Naturelle (TALN). La quantité de données augmente par le fait des progrès de la technologie et de la recherche, qui permettent par ailleurs de mieux les traiter (ou est-ce l'inverse : si on ne pouvait pas traiter toutes ces données, chercherait-on à en obtenir sans cesse toujours plus ? Pourquoi une entreprise collecterait-elle des milliers d'heures d'entretiens si elle n'avait aucun moyen

de les explorer automatiquement ? Que serait Internet sans moteur de recherche ?).

Enfin, des lois statistiques élémentaires, on aura retenu que plus grande est la taille de l'échantillon considéré/sondé, plus grande est sa représentativité et plus les informations qu'il fournit sont proches de la réalité (qui prévaut dans l'ensemble de la population).

Les postulats énoncés dans les trois paragraphes précédents nous amènent naturellement au cadre des travaux abordés dans cette thèse : supposons que l'on souhaite connaître les avis des personnes concernées par un produit ou un service (des opinions exprimées, des thématiques abordées, ...), que ces personnes soient potentiellement très nombreuses (des centaines, des milliers, des dizaines de milliers, bien plus ?), que l'on souhaite avoir une bonne représentation de l'ensemble de ces personnes, que les progrès technologiques nous permettent d'accéder à des masses importantes d'informations et que l'on cherche un moyen nous demandant moins d'effort (ou tout simplement plus plausible) que d'explorer manuellement ces informations ?

Ainsi, EDF, qui possède plus de 40 millions d'abonnés, s'intéresse depuis plusieurs années à l'emploi de méthodes de TALN pour l'amélioration de la gestion de la relation clients, que ce soit au travers de l'étude automatique d'enquêtes (Kuznick et al., 2010) qui permettent alors de poser des questions ouvertes (retournant potentiellement plus d'informations que des QCM) à un grand nombre de personnes, ou de conversations en centres d'appels (Cailliau et Giraudel, 2008; Bozzi et al., 2009; Danesi et Clavel, 2010a).

Les travaux présentés dans cette thèse s'inscrivent dans cette continuité, en proposant d'aborder plusieurs tâches à différents niveaux de granularité. Nous traiterons donc du problème de la catégorisation thématique de textes, de la reconnaissance du rôle du locuteur et de la détection de l'opinion globalement exprimée dans un document. Mais, nous considérons aussi que, pour ces problèmes, une information plus ciblée peut être intéressante. Sachant qu'un mot ne possède pas forcément le même sens selon le contexte dans lequel il est employé et que, comme l'énonce (Cusin-Berche, 2003) "le même avocat ne peut, sauf circonstances tout à fait exceptionnelles méritant un article dans la rubrique faits divers, être à la vinaigrette et à la Cour", ou que, dans un cadre de détection d'opinion, un *goût amer* n'offre pas le même jugement s'il est laissé par des vacances ou par un pamplemousse, nous mettrons en correspondance les unités lexicales avec les informations dont nous disposons à l'échelle du document ainsi que leur entourage proche.

Nous proposons ainsi une méthode permettant, dans le cas où nous disposons d'ensembles de documents étiquetés à l'échelle globale (à chacun d'entre eux est rattachée une ou plusieurs catégories), d'extraire automatiquement des chaînes de mots (segments textuels) relatifs aux différentes étiquettes. La méthode, probabiliste, possède l'avantage de pouvoir être appliquée à différentes sortes de problèmes. Nous montrerons ici son intérêt dans le cas où les étiquettes des documents correspondent à des thématiques, à des rôles de locuteur, ou encore à des opinions (positive, négative ou neutre). L'intérêt est en réalité double : les segments ainsi extraits permettent de visualiser, selon le problème considéré, des points fréquemment commentés dans une thématique, des expressions typiquement employées par un certain type de locuteur,

des critiques récurrentes, ... et, qui plus est, ces segments peuvent permettre d'améliorer les tâches de catégorisation automatique de textes associées à ces cas (catégoriser un document selon les thématiques qui y sont abordées, reconnaître le rôle associé à un tour de parole, détecter l'opinion globale d'une critique). Nous présenterons son apport dans ces différents cas, sur des corpus de nature différente (transcription de parole spontanée, blogs), relatifs à divers domaines (EDF, des films, des jeux vidéos) et dans deux langues (française et anglaise).

Nous nous intéressons ensuite au ciblage précis de jugements exprimés dans des corpus d'opinions. Nous proposons une méthode permettant de trouver des n-grammes porteurs d'opinions exprimées sur certains sujets prédéfinis, ce qui nous permet par la suite d'extraire des documents les segments textuels correspondants à ces jugements ciblés. Là encore, la méthode est probabiliste et sera testée sur deux corpus différents : des transcriptions manuelles d'enquêtes téléphoniques de satisfaction effectuées auprès de clients EDF et des critiques de films en langue anglaise écrites par des internautes.

1.1 Organisation du document

Le chapitre 2 présente une méthode, basée sur une métrique de recherche de collocations, permettant d'extraire automatiquement des chaînes de mots caractéristiques des différentes catégories présentes dans un corpus. Nous y expliquons ensuite comment combiner ces chaînes avec un système de catégorisation automatique de textes. Des résultats sont présentés sur une expérience de catégorisation de segments conversationnels selon les thématiques qui y sont exprimées, puis dans une tâche de reconnaissance du rôle du locuteur.

Le chapitre 3 propose une mise à l'épreuve de la méthode présentée au chapitre précédent, cette fois-ci dans un contexte de catégorisation de textes selon l'opinion qui y est globalement exprimée (positive, négative ou neutre). Nous testons la méthode sur des corpus dont les documents sont rattachés à une de ces catégories, bien qu'il s'agisse de contextes assez différents (par les thématiques abordées, les modes d'expressions, les langues) : un corpus de critiques de jeux vidéos produites par des internautes en français, une transcription manuelle d'enquêtes téléphoniques de satisfaction effectuées auprès de clients EDF et un corpus de critiques de films en langue anglaise. En plus de l'impact (positif) sur les performances des systèmes de catégorisation de textes, nous présentons de nombreux exemples des chaînes retournées par la méthode.

Au chapitre 4, nous présentons une nouvelle méthode servant à rechercher, dans des corpus d'opinions, des n-grammes porteurs d'opinions exprimées sur des sujets prédéfinis. Nous pouvons ensuite extraire dans des documents, les segments représentant l'expression d'une opinion ciblée, permettant un résumé des points critiqués, de quelle manière, avec quelle polarité.

Enfin, nous concluons les travaux exposés dans thèse et présentons quelques perspectives.

Chapitre 2

Des collocations, des chaînes caractéristiques, des catégories thématiques et des locuteurs

Sommaire

Résumé

Ce chapitre présente une méthode d'extraction de chaînes caractéristiques de catégories, basée sur une métrique de recherche de collocations. Le but est de fournir un aperçu des chaînes liées à chacune des catégories dans un souci d'exploration de corpus, tout en améliorant les résultats d'une catégorisation automatique de textes. Dans un premier temps, nous décrivons la méthode permettant d'extraire les chaînes caractéristiques des différentes catégories. Puis, nous indiquons comment celle-ci peut être introduite dans un système de catégorisation automatique. Nous montrons ensuite son application dans un contexte de catégorisation thématique de conversations en centre d'appels et pour la reconnaissance du rôle du locuteur. Dans ce second cas, nous combinons ce système avec une modélisation du dialogue.

2.1 Introduction

Les premiers travaux réalisés au cours de cette thèse concernent la catégorisation automatique de textes. La catégorisation automatique de textes consiste à rattacher automatiquement un texte à une ou plusieurs catégories prédéfinies (Sebastiani, 2002). Nous disposons pour cela d'un ensemble de textes pour lesquels les catégories sont connues (corpus d'apprentissage), servant à entraîner des modèles qui, par comparaison des mots composant les documents, permettront d'attribuer automatiquement les catégories à de nouveaux textes (corpus de test). De nombreux modèles ont été présentés dans la littérature. On trouvera par exemple une étude comparative pour quelques-uns d'entre eux dans (Yang et Liu, 1999). Les applications possibles sont multiples, on peut notamment citer : la gestion automatique des CV et lettres de motivation (Kessler et al., 2008), les filtres anti-spams utilisés par les messageries électroniques (Androutsopoulos et al., 2000), la recherche de la paternité d'une oeuvre littéraire (Savoy, 2011), la classification d'informations (celle-ci ayant par exemple donné lieu à la campagne d'évaluation DEFT08 (Grouin et al., 2008), dont le but était de catégoriser des articles de journaux selon la thématique qu'ils traitaient : sport, société, télévision, politique, ...), etc.

Notre approche propose d'aller au-delà d'une simple catégorisation thématique et d'introduire dans le cadre de cette tâche, une extraction de chaînes caractéristiques des différentes catégories connues. Ces chaînes ont un double intérêt : elles permettent à la fois d'améliorer les performances d'un système de catégorisation de textes par la prise en considération d'éléments potentiellement plus discriminants que des mots pris isolément et peuvent être visualisées afin d'offrir un aperçu des sous-thématiques ou sujets fréquemment abordés, des expressions récurrentes de chacune des catégories, ... Nous appliquons ensuite la méthode pour deux tâches différentes de catégorisation automatique de textes : l'identification des thématiques abordées en centre d'appels EDF et l'identification du rôle du locuteur par tour de parole (la phrase a-t-elle été prononcée par un client ou un agent ?), nous mixons, dans ce deuxième cas, la méthode présentée avec une modélisation du dialogue.

2.2 De la collocation à la recherche de chaînes caractéristiques des catégories

Dans cette section, nous positionnons et présentons la méthode permettant d'extraire des chaînes de mots caractéristiques des différentes catégories.

La méthode que nous proposons pour extraire des chaînes de mots relatives à chacune des catégories 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 co-occurrant plus souvent que par le fruit du hasard (Smadja, 1993). Nous nous sommes intéressé aux méthodes statistiques de recherche de collocations (la plus simple étant celle qui retourne les bigrammes les

plus fréquents, bien que celle-ci ait une tendance à plutôt retourner les combinaisons de mots-outils (Manning et Schütze, 2000)), dont un panel est présenté dans (Yu et al., 2003; Smadja et McKeown, 1990). 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 constituer 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 catégorie, grâce à une méthode largement indépendante de la langue, puis de les exploiter pour une tâche de catégorisation. 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 forcément thématiques. Nous appliquerons ainsi la méthode dans différents contextes : catégorisation thématique, détection d'opinion, identification du rôle du locuteur. Dans un but de constitution de terminologie, et non de catégorisation 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 et peut constituer, pour nous, une perspective intéressante. Par la suite, (Dias et Vintar, 2005) utilisent une fenêtre de 11 mots pour extraire des multimots limités en taille (2 ou 3 mots). Notons toutefois qu'ils utilisent les catégories morpho-syntaxiques 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 à différents problèmes de catégorisation. Notre méthode consiste à extraire des collocations caractéristiques de chacune des catégories par un calcul du logarithme du rapport de vraisemblance (LRV) appliqué sur les documents rattachés à une des catégories. Par itérations (calcul de collocations sur les collocations déjà trouvées), on rajoute petit à petit des mots afin d'obtenir des chaînes d'une taille plus étendue que celles

proposées par les approches citées précédemment.

Le LRV, présenté sous sa forme développée en formule 2.1 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.*

$$\begin{aligned} LRV = 2 \times ([C_{12} * \log \frac{C_{12}}{C_1} + (C_1 - C_{12}) \times \log (1 - \frac{C_{12}}{C_1})] \\ +[(C_2 - C_{12}) * \log \frac{C_2 - C_{12}}{N - C_1} + ((N - C_1) - (C_2 - C_1)) \times \log (1 - \frac{C_2 - C_{12}}{N - C_1})] \\ -[C_{12} * \log \frac{C_2}{N} + (C_1 - C_{12}) \times \log (1 - \frac{C_2}{N})] \\ -[(C_2 - C_{12}) * \log \frac{C_2}{N} + ((N - C_1) - (C_2 - C_{12})) \times \log (1 - \frac{C_2}{N})]) \end{aligned} \tag{2.1}$$

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 ("corpus" faisant référence ici à l'ensemble de tous les textes rattachés à une même catégorie).

On considère tous les documents du corpus d'apprentissage étiquetés avec une certaine catégorie et on y applique un calcul de LRV. On obtient ainsi une liste de collocations par catégorie, 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'étiquette 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 mot 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 la collocation « *m1–m2* » et le mot *m3* ont un LRV et un nombre d'occurrences suffisamment élevé pour être considérés comme une collocation et ainsi 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 maximale, puis celles de la taille directement inférieure, etc. Par exemple, à l'itération 2, on va agglutiner toutes les chaînes de taille 4 que l'on trouve dans le corpus, puis celles de taille 3, puis celles de taille 2. 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 de nombre d'occurrences. Nous obtenons ainsi au final une liste de chaînes par catégorie et un corpus sur lequel ces chaînes ont été agglutinées. Notons qu'il pourrait être 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 [1] $t1\ t2$ 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. $i++$

À titre de comparaison, nous avons tenté, en expériences préliminaires, d'autres méthodes d'extraction de chaînes. Celles-ci suivent le même principe d'agglutinations de collocations, mais la métrique employée pour la recherche de collocations est différente : LRV sans distinction de classe, information mutuelle, ... Ces méthodes sont présentées succinctement en annexe C. Nous ne les avons pas retenues car les résultats produits étaient moins intéressants.

2.3 Combiner catégorisation et recherche de chaînes

Dans cette section, nous expliquons comment et pourquoi nous réalisons une catégorisation automatique de textes en utilisant des chaînes calculées par catégorie, bien qu'appliquées de manière indifférenciée sur l'ensemble du corpus.

Nous nous proposons de tester l'impact de ces chaînes sur des problèmes de catégorisation automatique de textes. L'idée sous-jacente est que considérer des chaînes plutôt

1. À la première itération $t1$ et $t2$ seront des mots isolés, au-delà ils pourront aussi chacun être des agglutinations de plusieurs mots

que les mots isolés peut permettre de prendre en compte des expressions, des composants potentiellement 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 le cas où les catégories correspondraient à des appréciations, 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 ». Dans le cas d'articles de nouvelles à ranger selon la thématique, si nous considérons la phrase "*Hier soir, Antenne 2 diffusait le match, pendant que les concurrents nous offraient le débat présidentiel.*", le fait que le système parvienne à reconnaître comme une même entité *Antenne 2 diffusait le match* (ou d'un côté *Antenne 2* et de l'autre *diffusait le match*, Antenne 2 étant dans ce cas sans nul doute le nom de la chaîne) peut permettre d'attribuer la catégorie Télévision, alors qu'avec les mots pris isolément *match* aurait créé une confusion en augmentant la similarité avec la catégorie Sport.

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'empan, 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 catégorisation de textes. Cependant, nous avons choisi 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 est peu 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 (par exemple, les articles de nouvelles), mais à une des catégories (par exemple, Sport, Société, Politique) : il est possible qu'elles soient donc plus discriminantes dans un contexte de catégorisation automatique de textes ; dans certaines de nos expériences, ce phénomène est accentué par l'emploi d'une pondération renforçant

l'influence des termes les plus discriminants (formules 2.3 et 2.4) ;
- les multimots font généralement référence à des expressions de 2 ou 3 mots ; les chaînes que notre système retourne peuvent être plus grandes (dans les expériences présentées dans cette thèse, certaines seront composées d'une dizaine de mots), couvrant ainsi des expressions, des portions de phrases potentiellement plus intéressantes.

Nous avons développé un système qui extrait les chaînes, les applique et effectue une catégorisation, en fonctionnant par itérations destinées à rechercher des chaînes de plus en plus longues. Ceci correspond à l'ajout d'une phase d'apprentissage des modèles, de catégorisation et d'évaluation des résultats entre les instructions 9 et 10 du pseudo-code présenté en section 2.2. Ainsi, à l'itération 1, on effectue une première catégorisation en n'utilisant aucune chaîne (mots isolés seulement) puis on calcule 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 obtient ainsi des chaînes de tailles 2, 3 et 4, qui seront utilisées à l'itération 3. On peut alors voir l'évolution des résultats de la catégorisation en fonction de la longueur et la quantité des chaînes proposées. Le fonctionnement général du système correspondant est présenté en figure 2.1.

2.4 Application dans le cadre de catégorisation thématique d'enregistrements de conversations en centre d'appels EDF

Après avoir présenté le contexte applicatif, nous détaillons la méthodologie retenue et présentons les résultats obtenus.

2.4.1 Description de la tâche - Le corpus CallSurf ManTransTopics-570 (CM570)

Dans cette section, nous nous intéressons au problème de la catégorisation thématique de segments de conversations téléphoniques. (Clemens et al., 2009) se sont confrontés à un problème proche du nôtre, dans un contexte de centre d'appels d'une compagnie téléphonique : afin d'éviter les temps d'attente au téléphone pour les clients, l'entreprise a mis en place un système de répondeur avec analyse automatique de la requête du client. Ainsi, quand le client appelle, il est redirigé vers le répondeur. Il y enregistre sa demande, qui est transcrite par un module de reconnaissance automatique de la parole. Un système d'analyse textuelle attribue ensuite une ou plusieurs catégories à ce document, représentant les types de problèmes qui sont exprimées par le client (panne de connexion, facture, ...) et surligne les mots-clés qui y sont reconnus, ceux-ci étant un mélange de mots statistiquement liés à une catégorie et de certains termes manuellement définis par des experts. La demande peut alors être automatiquement

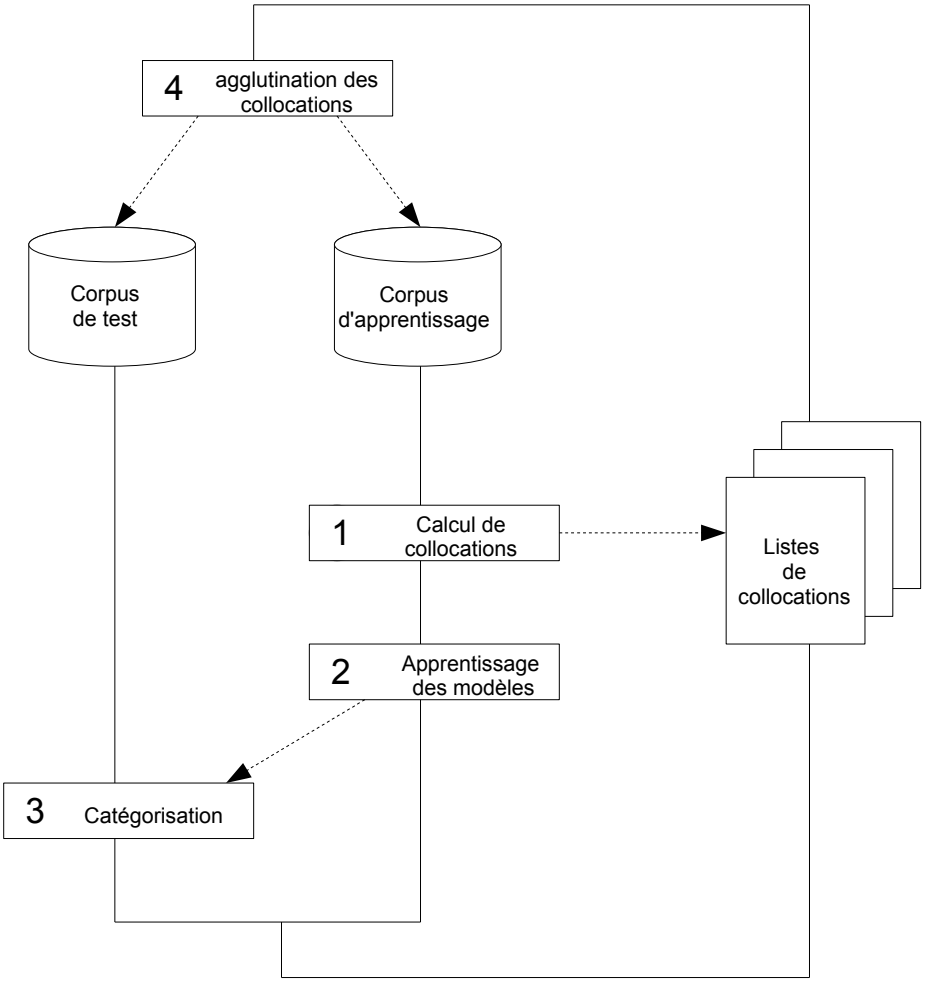

FIGURE 2.1 – *Fonctionnement du système de catégorisation avec prise en compte de chaînes. Dans l'ordre, à chaque itération : 1) on extrait du corpus d'apprentissage une liste de collocations par catégorie ; 2) on entraîne le classifieur sur le corpus d'apprentissage ; 3) on effectue la catégorisation du corpus d'apprentissage (+ calcul du score de catégorisation) ; 4) on utilise l'ensemble des listes de collocations réunies en une seule pour agglutiner dans l'ensemble des corpus les termes correspondants. À l'itération suivante, les nouvelles chaînes sont ainsi considérées.*

dirigée vers l'employé compétent pour la traiter tout en lui mettant en valeur les éléments importants de la requête. Ainsi, quand l'employé est disponible, il peut rappeler le client en ayant préparé l'entretien.

Dans notre cas, nous traitons l'intégralité d'un entretien entre un client et un employé dans un centre d'appels. Une conversation est découpée en plusieurs sous-parties (segments). À chaque segment sont attribuées une ou plusieurs catégories en fonction du contenu de l'échange.

Ce type de corpus provenant de parole spontanée possède des spécificités engen-

drant un certain nombre de difficultés pour son traitement automatique : il s'agit de transcriptions de données orales de données disfluentes, on y retrouvera des hésitations, des débuts de mots coupés, des phrases non-terminées, des répétitions, des retours en arrière pour reprendre le début d'une phrase, etc. comme cela a été soulevé dans (Danesi et Clavel, 2010b). Ceci peut entraîner des difficultés pour la lemmatisation, la recherche et l'agglutination des chaînes, ... De plus, ces données conversationnelles sont, par nature, peu structurées sur le plan thématique (les frontières entre deux thématiques peuvent être floues, les thèmes explicités de manière décousue et peu claire), ce qui peut rendre délicate l'identification de catégories uniques sur des segments bien délimités.

Le corpus *CallSurf ManTransTopics-570* utilisé ici est un sous-ensemble du corpus CallSurf (Garnier-Rizet et al., 2008) d'EDF. Le corpus CallSurf est constitué d'enregistrements de conversations téléphoniques effectués au cours de l'été 2006 dans un centre d'appel EDF Pro (les clients sont des entreprises). Les enregistrements ont été effectués sur 2 mois sur les postes de quelques dizaines d'agents, menant à un corpus total d'environ 5800 appels, soit 620 heures. Une conversation correspond à un appel complet d'un client à un agent pour soulever un problème, poser une question, demander un renseignement. Dans certains cas, il s'agit d'un appel de l'agent vers le client. Au cours de la conversation, l'agent peut appeler un autre agent, par exemple pour obtenir des informations supplémentaires auprès d'un autre service. Ce corpus a déjà été utilisé dans le cadre de travaux portant sur la recherche et navigation dans les textes (Cailliau et Giraudel, 2008).

Le corpus *CallSurf ManTransTopics-570* (CM570) consiste en 65 heures extraites du corpus CallSurf qui ont été manuellement transcrites et anonymisées (les noms, numéros de téléphones, références clients, informations bancaires ont été remplacés par des labels génériques). Ces transcriptions ont ensuite été manuellement segmentées et annotées. Les annotateurs devaient dans un premier temps découper la conversation en parties thématiquement cohérentes (segments), puis choisir pour chaque segment une ou plusieurs catégories parmi la liste présentée dans le tableau 2.1. Cette liste a été définie par des experts EDF après une analyse des résultats d'une classification non-supervisée effectuée sur l'ensemble du corpus CallSurf. On peut observer deux grands types de catégories : certaines sont purement thématiques (problème avec la Facture, prise de rendez-vous, ...) alors que d'autres sont plutôt structurelles (Ouverture, Fermeture, Conversation Agent-Agent, ...).

Le corpus a été lemmatisé avec LiaTagg [2], un étiqueteur morpho-syntaxique basé sur ECSTA (Spriet et El-Bèze, 1999). Nous avons choisi, afin de préserver la généricité de la méthode [3], de conserver tous les lemmes (pas d'utilisation d'anti-dictionnaire ou de filtre sur les catégories morpho-syntaxiques des mots). Ce corpus a été découpé de la manière suivante :

– corpus d'apprentissage : 55 heures, correspondant à 480 conversations, que nous avons réparties en 10 sous-ensembles afin d'adapter nos modèles en validation croisée ;

2. http ://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/bechet/download_fred.html
3. Des expériences ont été réalisées sans lemmatisation, celle-ci apporte un léger gain

– corpus de test : 10 heures, correspondant à 90 conversations, chacune ayant été annotée par 3 annotateurs différents ; nous considérons par conséquent 3 corpus de tests ;

Le fait d'avoir trois annotations sur les mêmes conversations permet de mettre en exergue le problème de l'accord inter-annotateur. Nous pouvons tout d'abord voir grâce au tableau 2.2 qu'il y a un désaccord sur la tâche de segmentation thématique : le premier annotateur (colonne "Test1") a trouvé 1001 segments, alors que le deuxième ("Test2") en a créé 686 sur les 90 mêmes conversations et le troisième 941. De plus, ils ont des avis différents sur les thématiques abordées dans les entretiens : nous pouvons voir dans le tableau 2.1 le nombre de fois où chaque catégorie a été assignée. On observe que :

– la répartition générale des catégories est assez déséquilibrée : dans le corpus d'apprentissage par exemple, la catégorie "Ouverture" est présente 473 fois, alors qu'on ne trouve que 12 segments étiquetés "Projet de Travaux". Certains pourraient donc penser que nos modèles seront meilleurs pour la première, sous prétexte qu'elle est plus peuplée, mais cela peut être plus complexe, comme nous le discuterons ultérieurement.
– il y a des différences non-négligeables entre les trois annotateurs, par exemple sur le nombre de segments étiquetés comme "Discussion Commerciale" : une vingtaine de fois pour deux des annotateurs, 4,5 fois (c'est-à-dire 4 segments, plus un segment pour lequel cette étiquette n'était pas la seule à être attribuée) pour le troisième.

L'accord inter-annotateur a été évalué de deux manières. Premièrement, nous avons utilisé WindowDiff (Pevzner et Hearst, 2002) pour mesurer les écarts sur la segmentation. Les annotateurs 1 et 2 sont ceux qui présentent le moins de différences (WindowDiff = 0,28), ensuite les annotateurs 1 et 3 (WindowDiff = 0,33) et les annotateurs 2 et 3 (WindowDiff = 0,35). Deuxièmement, nous avons évalué les différences sur l'étiquetage, en comparant les catégories assignées phrase à phrase. Les annotateurs 1 et 2 ont tendance à être relativement d'accord : ils ont 66% de phrases étiquetées pareillement. À l'opposé, les annotateurs 2 et 3 ont seulement un tiers d'étiquettes communes (34%) et les annotateurs 1 et 3, 37%. Toutefois, nous ne savons pas lesquelles de ces différences peuvent être considérées comme importantes (une phrase réellement informative) et lesquelles le sont moins (les phrases situées à la frontière entre deux segments thématiques par exemple). Cette mesure sur l'étiquetage reflète par ailleurs les erreurs sur la segmentation.

2.4.2 Expériences / Protocole

Nous avons appliqué sur ce corpus la méthode présentée en Section 2.3 pour extraire des chaînes caractéristiques tout en catégorisant les segments thématiques. La catégorisation est ici effectuée avec une mesure de similarité de type cosinus pondéré, utilisant la représentation TFxIDF ("Term Frequency - Inverse Document Frequency") (Salton et Buckley, 1988), ajustée avec le critère de pureté de Gini, tel que présenté dans (Torres-Moreno et al., 2007). Cette variable réduit l'influence des éléments

possédant un faible pouvoir discriminant (ceux qui apparaissent dans plusieurs catégories sans trop de disparités). Sa valeur est ainsi maximale quand le mot n'apparaît que dans une seule des catégories et minimale quand il est présent de manière équitable dans toutes les catégories possibles. Les éléments peuvent ici être des mots ou des multi-mots (les chaînes ayant été agglutinées seront considérées de la même manière qu'un mot par le système). La mesure de similarité entre un document d et une catégorie c est calculée comme présenté en formules 2.2 et 2.3. $Cos(d,c)$ est le cosinus pondéré (les coefficients $\lambda_{j=1...5}$ sont des nombres réels positifs) entre le vecteur du document Wd et le vecteur représentant la catégorie Wc, dont les dimensions sont les mots (ou multi-mots) i :

$$Cos(d,c) = \frac{\sum_i (W_{id} \times W_{ic})}{(\sqrt{\sum_i W_{id}^2 \times \sum_i W_{ic}^2})^{\lambda_1}} \tag{2.2}$$

où :

$$\begin{aligned} W_{id} &= TF_d(i)^{\lambda_3} \times IDF(i)^{\lambda_4} \times pGini(i)^{\lambda_5} \\ W_{ic} &= TF_c(i)^{\lambda_2} \times IDF(i)^{\lambda_4} \times pGini(i)^{\lambda_5} \end{aligned} \tag{2.3}$$

$pGini(i)$ est le critère de pureté de Gini, c'est-à-dire le pouvoir discriminant du mot i, selon sa répartition dans les catégories k (formule 2.4).

$$pGini(i) = \sum_k P(k|i)^2 \tag{2.4}$$

Il a empiriquement été décidé que :

- pour calculer les fréquences des mots dans les modèles (TFc), on ne compte qu'une seule occurrence maximum par segment ; cela évite d'accorder trop d'importance aux disfluences (dans la mesure où il s'agit d'un corpus issu de parole spontanée, des mots sont répétés de manière incorrecte : "*Je ne trouve plus le le le euh le dossier*").
- pour calculer les fréquences des mots (TFd) dans le document d que l'on tente de catégoriser, nous comptons en revanche toutes ses occurrences. Les résultats obtenus lors des expériences préliminaires ont été légèrement meilleurs dans ce cas. Toutefois, l'importance des répétitions incorrectes peut être réduite en ajustant la valeur du coefficient λ_3. Dans les expériences présentées ici, cette valeur sera d'ailleurs assez faible.

Pour éviter les problèmes de faible couverture de certains éléments, le système considère à la fois les chaînes agglutinées (multi-mots) et leurs composants. Pour le calcul des TF, alors que les chaînes comptent pour 1, leurs composants sont ajoutés aux comptes des mots isolés avec une valeur plus faible (baptisée $AggCoef$).

Pour le calcul des IDF (formule 2.5), les mots isolés, les chaînes agglutinées et les mots qui les composent comptent pour 1 (par exemple, si nous avons 4 documents : le

premier contenant "$w_1 - w_2$", le second "w_1", le troisième à la fois "w_1" et '$w_1 - w_2$", et le dernier aucune occurrence de w_1, alors $IDF(w_1) = -\log(\frac{1+1+1+0}{4})$).

$$IDF(i) = -\log\left(\frac{\#docs\ contenant\ i}{\#docs}\right) \tag{2.5}$$

Nous avons par ailleurs testé l'apport des chaînes dans un cas de catégorisation avec une mesure cosinus standard (sans pondérations), cela correspond à fixer à 1 la valeur de λ_1 dans la formule 2.2 et $\lambda_5 = 0$ ainsi que $\lambda_2 = \lambda_3 = \lambda_4 = 1$ dans la formule 2.3, les mots sont alors représentés par leur TFxIDF uniquement. Les résultats de catégorisation ont aussi été comparés à ceux obtenus par un algorithme de boosting. Nous avons pour cela utilisé Boostexter (Schapire et Singer, 2000), en utilisant comme données d'entrées soit des trigrammes, soit des bigrammes, soit des unigrammes (unimots), soit des unigrammes obtenus par la méthode des recherche des chaînes caractéristiques (les unigrammes sont ainsi des unimots ou des multimots agglutinés).

Tous les paramètres de ces différentes expériences (les lambdas servant à pondérer le cosinus, le nombre d'itérations de l'algorithme de boosting) ont été fixés par la recherche d'un score optimal de catégorisation sur le corpus d'apprentissage en validation croisée.

Les résultats sont évalués avec la formule 2.6, donnant, pour chaque segment à catégoriser, un score compris entre 0 et 1.

$$SegScore = \frac{\sum_{i=1}^{|Ref|} \frac{1}{i} \times \sigma_i}{\sum_{i=1}^{|Ref|} \frac{1}{i}} \tag{2.6}$$

où :

- Ref est l'ensemble non-ordonné des catégories qui auraient dû être trouvées (les références étiquetées par les annotateurs) pour le segment courant,
- $|Ref|$ est le cardinal de cet ensemble, c'est-à-dire le nombre de catégories attribuées par l'annotateur,
- $\sigma_i = 1$ si $Prop(i) \in Ref$; $\sigma_i = 0$ si $Prop(i) \notin Ref$
- $Prop$ est l'ensemble ordonné des étiquettes proposées par notre système.

Sachant que l'on ne connaît pas le nombre de catégories à trouver (et que celles-ci ont été attribuées par l'annotateur sans valeur indiquant leur importance relative) et que notre système propose un score de similarité avec chacune des catégories possibles, cette formule nous permet d'évaluer si les premières catégories que l'on propose (celles avec la plus forte similarité) sont exactes, en tenant compte de leur ordre (une erreur sur la première catégorie que l'on retourne est plus pénalisante qu'une erreur sur la seconde). Par exemple, pour un segment donné, trois catégories ont été proposées par les annotateurs. On va donc évaluer la pertinence des trois catégories que notre système retourne avec la plus forte probabilité. Supposons que l'on ait raison pour la première et la troisième (c'est-à-dire que celles-ci font bien partie des trois

étiquettes que l'annotateur a attribuées), notre score pour ce segment sera donc de : $segScore = \frac{(1/1) \times 1 + (1/2) \times 0 + (1/3) \times 1}{(1/1) + (1/2) + (1/3)} = 0,73.$

On additionne ensuite les scores obtenus pour chacun des segments, que l'on divise par le nombre de segments, afin d'obtenir la réussite moyenne sur l'ensemble du corpus. Dans les résultats présentés ici, on a multiplié cette moyenne par 100, afin d'indiquer le pourcentage de réussite à la catégorisation.

Les paramètres optimaux (pondérations du cosinus) trouvés sont :

$$\lambda_1 = 1.6, \lambda_2 = 0.7, \lambda_3 = 0.2, \lambda_4 = 0.6, \lambda_5 = 0.1, AggCoef = 0.1$$

Les propositions de chaînes que l'on conserve dans les expériences suivantes sont celles ayant un LRV supérieur ou égal à 75 et présentes au moins 7 fois dans le corpus d'apprentissage.

2.4.3 Résultats

2.4.3.1 Scores de catégorisation / Résultats numériques

Les résultats de la catégorisation en validation croisée sur le corpus de développement sont présentés en figure 2.2.

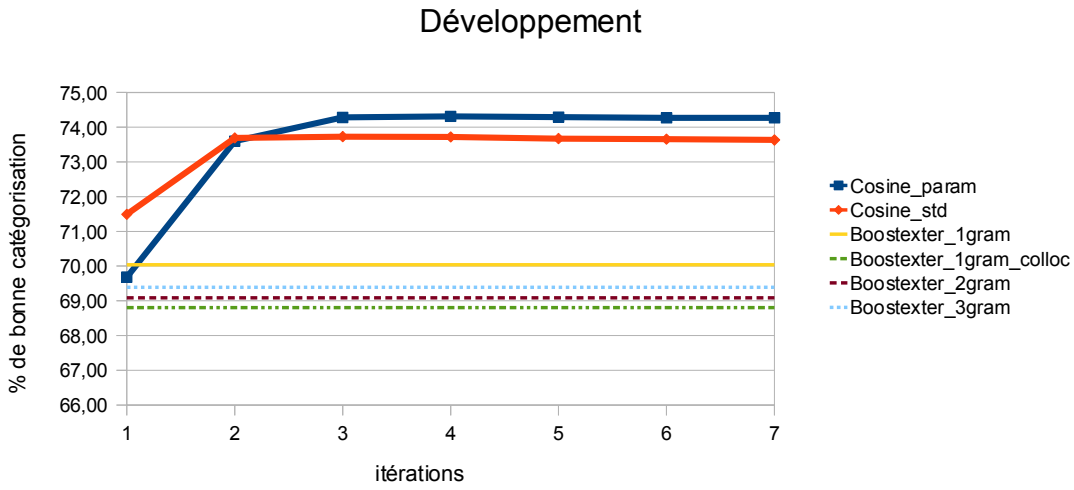

FIGURE 2.2 – *CM570 - Pourcentage de bonne catégorisation sur le corpus de développement.*

Les itérations correspondent à l'application du processus tel que présenté en section 2.3. C'est-à-dire qu'à l'itération 1, on catégorise sans avoir recherché de mots à

agglutiner. Le système de catégorisation prend donc en entrée les mots isolés contenus dans le corpus. À la fin de l'itération 1, on applique la méthode présentée en section 2.2, afin d'obtenir des propositions de paires de mots que l'on va agglutiner dans le corpus d'apprentissage comme dans le corpus de test (ici il s'agit d'un sous-ensemble tournant du corpus d'apprentissage). À l'itération 2, on ré-apprend les modèles des différentes catégories à partir de ce nouveau corpus (certains éléments sont donc des bimots agglutinés pour ne former qu'un seul, d'autres sont toujours des unimots isolés). Puis, on catégorise et calcule le nouveau score (celui qui est présenté dans la colonne "itération 2" en abscisse). À l'itération suivante, suite aux nouvelles propositions d'agglutinations, le corpus contiendra des multimots de longueurs 1, 2, 3 et 4... Ainsi de suite jusqu'à ce qu'il n'y ait plus aucune proposition d'agglutinations, c'est-à-dire qu'il n'y a plus aucun candidat qui obtienne un LRV et un nombre d'occurrences supérieurs aux seuils fixés pour le filtre. Ces itérations sont celles représentées sur les graphiques, mais sont inutiles dans les expériences basées sur Boostexter : pour celles-ci on a en effet choisi soit de catégoriser sans utilisation d'agglutinations (les centaines d'"itérations" évoquées dans le paragraphe suivant pour ces systèmes correspondent à celles propres à la catégorisation par boosting et non à des recherches de chaînes), soit avec les agglutinations maximales que l'on a obtenues sur ce corpus.

Les systèmes présentés sont :

- Cosine_param : le cosinus pondéré, tel que représenté par les formules 2.2 et 2.3, utilisant les agglutinations,
- Cosine_std : le cosinus standard (sans pondération), utilisant la représentation TFxIDF et les mêmes agglutinations,
- Boostexter_1gram : Boostexter utilisant un modèle unigramme - entraîné avec 750 itérations,
- Boostexter_2gram : Boostexter utilisant un modèle bigramme - entraîné avec 600 itérations,
- Boostexter_3gram : Boostexter utilisant un modèle trigramme - entraîné avec 850 itérations,
- Boostexter_1gram_colloc : Boostexter utilisant un modèle unigramme, quand les agglutinations obtenues après 7 itérations ont été appliquées sur le corpus (sans mesure de repli pour considérer les composantes des agglutinations de manière séparée) - entraîné avec 820 itérations.

Sur ce corpus d'apprentissage, on observe que :

- le cosinus pondéré fournit de meilleurs résultats que le cosinus standard : sans agglutinations (itération 1), le cosinus standard atteint 71,5% de bonne catégorisation, contre 69,7% pour le cosinus pondéré, mais à partir du moment où on utilise les agglutinations, le cosinus pondéré devient aussi performant ou meilleur que le cosinus standard (74,3% contre 73,6%). Le fait que les pondérations du cosinus aient été optimisées sur la tâche de catégorisation quand les agglutinations de la dernière itération ont été appliquées dans le corpus, explique pourquoi le cosinus pondéré n'offre pas de très bons résultats à l'itération 1.
- dans le cas du cosinus standard, les agglutinations sont utiles pour améliorer le score de catégorisation (+2,1%), ce système ayant l'avantage d'être plus facile à adapter que le cosinus pondéré qui nécessite la recherche des paramètres opti-

maux.

Les systèmes ont par la suite été mis à l'épreuve sur le triple corpus de test (les mêmes 90 conversations étiquetées par 3 annotateurs différents). Les résultats de ces expériences sont présentés pour les trois corpus de test : la segmentation et l'étiquetage proposés par l'annotateur 1 en figure 2.3, pour l'annotateur 2 en figure 2.4 et pour le troisième en figure 2.5.

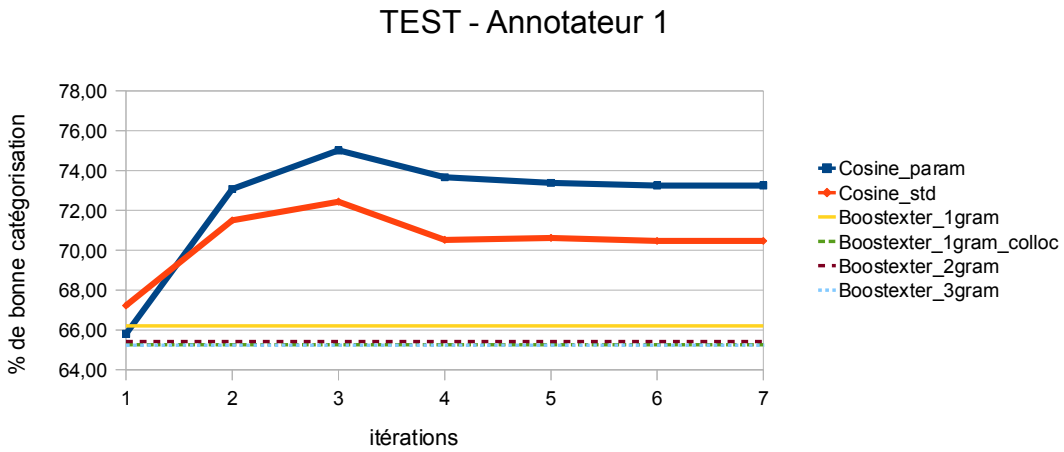

FIGURE 2.3 – *CM570 - Pourcentage de bonne catégorisation sur le corpus de test 1.*

Sur les corpus de test, on observe que :

- le types de classifieurs basés sur la similarité cosinus, offrent, sur cette tâche, de meilleures performances que ceux basés sur Boostexter.
- sur deux des trois jeux de test, le cosinus pondéré est meilleur que le cosinus standard. Comme précédemment, les gains importants observés lorsqu'on utilise les agglutinations avec le cosinus pondéré (+7,5% sur le test 1, +5% sur le test 2, +4% sur le test 3) sont à relativiser par le fait que l'on a adapté, lors de l'apprentissage, les paramètres sur les corpus dans lesquels les chaînes avaient été agglutinées.
- les agglutinations permettent d'améliorer la catégorisation effectuée avec le cosinus standard (+3,2% sur le test 1, +2,3% sur le test 2, +3% sur le test 3).
- ces agglutinations sont en revanche inutiles dans le cas de la catégorisation effectuée avec Boostexter prenant en compte des unigrammes : le fait que ces unigrammes puissent être des multimots agglutinés décroît les performances de 0,1 à 0,9% selon le corpus.
- d'un point de vue global, tous les systèmes sont meilleurs sur le corpus étiqueté par l'annotateur 2 : c'est celui qui a proposé l'annotation la plus proche de l'ensemble des personnes sollicitées pour l'étiquetage du corpus d'apprentissage.

Aussi bien sur le corpus d'apprentissage que sur ceux de test, on peut noter que le gain principal en terme de catégorisation s'effectue à l'itération 2 (par exemple en

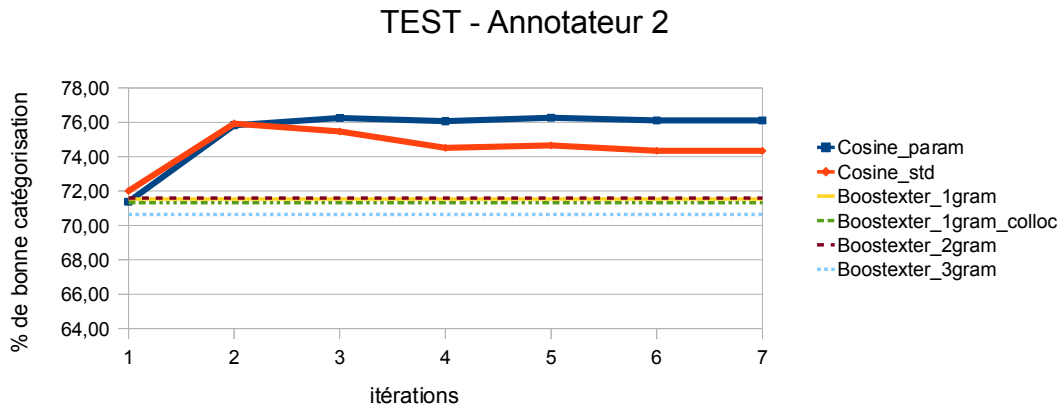

FIGURE 2.4 – *CM570 - Pourcentage de bonne catégorisation sur le corpus de test 2.*

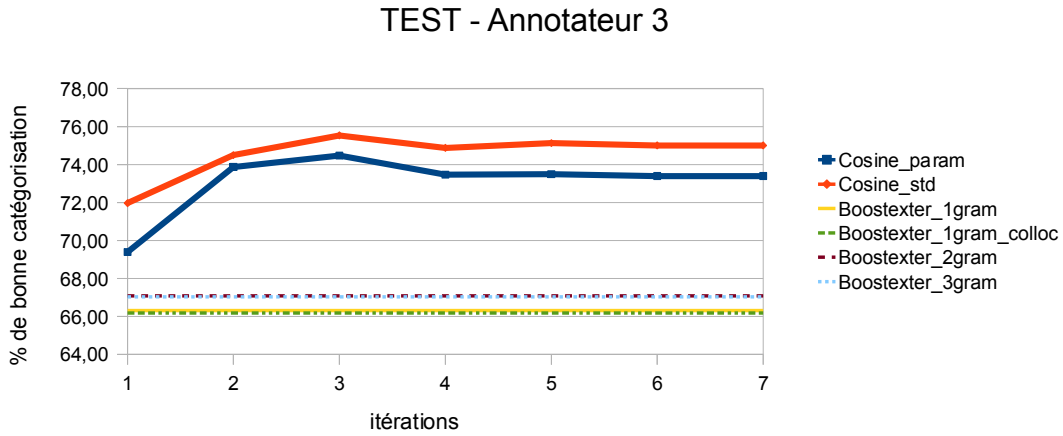

FIGURE 2.5 – *CM570 - Pourcentage de bonne catégorisation sur le corpus de test 3.*

regardant la courbe du cosinus standard, étant donné que le cosinus pondéré est optimisé pour la prise en compte des chaînes maximales, les gains aux différentes étapes sont plus difficilement interprétables pour ce classifieur, bien que les deux courbes suivent la même forme), c'est-à-dire quand on considère soit des unigrammes, soit des bigrammes. Un gain supplémentaire est généralement observé à l'itération 3 (on uti-

lise des chaînes de longueur 4, des trigrammes, des bigrammes ou des unigrammes pour les mots qui n'appartiennent à aucune chaîne connue), puis les résultats se stabilisent (les nouvelles chaînes que l'on trouve, qui peuvent à l'itération suivante contenir jusqu'à 8 mots, deviennent moins présentes dans les corpus et ont donc moins d'influence) à un niveau légèrement inférieur. Cette légère baisse provient des problèmes de couverture : les grandes chaînes sont peu présentes dans le corpus, elles sont donc peu souvent appliquées, quand c'est le cas, cela peut être au détriment de deux sous-chaînes qui auraient peut-être pu être plus discriminantes à elles deux réunies (par exemple, à *pGini* identiques, les TF des deux sous-chaînes - par définition au moins aussi fréquentes que la grande chaîne - auraient eu plus d'impact que le TF de la chaîne englobante). Qui plus est, ces grandes chaînes sont probablement plus discriminantes (les chaînes les plus grandes sont moins ambiguës et peuvent donc se retrouver plus fortement dans une des catégories par rapport aux autres) et ont ainsi un *pGini* élevé, ce qui peut pousser le système à prendre une décision plus tranchée pour une catégorie (celle dans laquelle la chaîne est le plus apparue, donc le TF dans le modèle sera élevé et l'importance de cette chaîne est accrue par son *pGini* fort) aux dépens des autres (qui pouvaient être correctes aussi, puisque l'on est dans un cas où plusieurs catégories peuvent être rattachées à un même segment).

Il est possible que ce critère ne soit ainsi pas parfaitement adapté aux problèmes multi-catégories (quand plusieurs catégories peuvent être rattachées à un même document) : il peut être trop discriminant et favoriser à outrance une catégorie, pénalisant ainsi trop fortement les autres (la grande chaîne a un *pGini* très élevé car elle n'est présente quasiment que dans les segments rattachés à la catégorie A, ses deux sous-chaînes avaient des *pGini* plus faibles, l'une penchant vers la catégorie A, l'autre plus mitigée penchant aussi vers la catégorie B : on a ainsi perdu de l'information quant à la catégorie B). Cela soulève des pistes d'améliorations possibles : trouver un critère moins discriminant qui serait plus adapté aux cas multi-labels, par exemple une variante de ce *pGini* qui serait plus souple, en prenant en considération qu'il peut être intéressant de favoriser plusieurs catégories contre toutes les autres (on attribue de l'importance à un élément qui se rencontre dans deux des n catégories, même s'il s'y retrouve équitablement réparti (un tel élément dispose d'un $pGini = 0,5^2 + 0,5^2 = 0,5$ ce qui peut être vu comme faible au regard d'un élément ne se retrouvant que dans une seule catégorie qui lui aura un $pGini = 1$)). Ce problème est ici représenté par le fait que le meilleur paramètre que l'on ait trouvé pour pondérer la valeur de la variable *pGini* est proche de zéro ($\lambda_5 = 0,1$) ce qui signifie que son impact est minime. Le nombre de chaînes différentes utilisables par le système pour chaque itération est indiqué dans le tableau 2.3. Ainsi, à l'itération 2, on connaît 702 bimots qui peuvent être agglutinés dans le corpus avant d'effectuer la catégorisation. À l'itération 3, il existe 1152 modèles d'agglutinations (du bimot au quadrimot). La quantité de nouvelles propositions ayant des scores/valeurs supérieur(e)s aux seuils décroît ensuite rapidement au-delà et il n'y en a plus aucune à la septième. Le tableau 2.4 indique le nombre de modèles de chaînes extraits sur le corpus d'apprentissage, en fonction de leur taille, pour la dernière itération : le système connaît 776 bimots, 334 trimots, ... Des exemples de ces chaînes sont présentés en section suivante.

2.4.3.2 Chaînes

Nous présentons ici quelques exemples de chaînes (extraites du corpus **lemmatisé**) proposées par le système. "*jérôme-xx-à-votre-service-bonjour*" est typique de la catégorie "Ouverture", il s'agit de la phrase d'introduction de l'agent auprès du client. Cette formulation est légèrement différente de ce que l'on peut observer dans le cas d'une conversation entre deux agents : la phrase est moins formelle (on trouve des expressions telles que "*ouais-ouais*", qui ne sont pas utilisées dans les segments des autres catégories, c'est-à-dire quand l'agent s'adresse à un client) et on y retrouve souvent une introduction plus complète, par exemple indiquant le la localisation du centre d'appels "*oui-bonjour-xx-edf-pro-montpellier*" ou encore une précision sur le service "*edf-gaz-de-france-distribution*". Pour la catégorie "Fermeture", on extrait bien entendu un certain nombre de formules de politesse "*ce-être-moi-qui-vous-remercier*", "*bon-fin-de-journée*". Concernant les catégories non-structurelles, on observe du vocabulaire spécifique aux différentes thématiques, par exemple pour les segments étiquetés comme "Informations Client", on trouve des chaînes relatives au numéro de téléphone du client "*coordonnée-téléphonique*", "*un-numéro-de-téléphone-où-on-pouvoir-vous-joindre*", "*votre-numéro-de-téléphone*".

Un segment de conversation où l'agent convient d'un rendez-vous avec le client contient bien entendu souvent "*avoir-un-rendre-vous*", ainsi que des informations temporelles "*d'après-midi*", "*seize-heure*". Dans le cas d'une discussion portant sur un contrat, nous extrayons des expressions liées au compteur, au voltage ou au courant : "*le-numéro-du-compteur*", "*lieu-de-consommation*", "*le-même-puissance-que*", une référence fréquente à l'occupant précédent des locaux "*de-votre-prédécesseur*", ainsi que l'expression courante "*sous-le-oeil*" (pouvant provenir d'une phrase telle que "J'ai mon/votre contrat sous les yeux"). Dans la catégorie "Facture-Paiement", on trouve des extraits des différentes étapes de cette thématique : du commencement "*relever-le-compteur*", au paiement et ses modalités ("*carte-bleu*", "*le-prélèvement-automatique*"), en passant par les délais et la régularité ("*tout-le-deux-mois*", "*délai-de-paiement*") et les problèmes éventuels "*lettre-de-relance*". Les chaînes extraites d'après la catégorie "Intervention Technique" permettent d'avoir un aperçu des raisons fréquentes d'interventions : "*branchement-provisoire*", "*mise-en-service*".

Un des intérêts de l'extraction des chaînes dans un tel contexte peut être l'analyse des sous-thématiques abordées dans le temps. On peut alors observer l'évolution des occurrences des chaînes ainsi extraites sur le même type de corpus à des périodes différentes et mettre en évidence des nouvelles problématiques (des chaînes apparaissent ou deviennent plus fréquentes), des sujets qui ne sont plus discutés. Ceci peut par exemple permettre de créer de nouvelles catégories (on observe l'apparition d'un certain nombre de chaînes jusqu'ici inconnues portant sur un thème commun). Des expériences préliminaires ont été menées en ce sens et sont présentées en annexe A.

Au-delà du contenu thématique (analyse des motifs d'appels, analyse de la réclamation), les chaînes offrent aussi une information sur la forme (le vocabulaire employé). Cela permet de se placer non plus seulement au niveau des thématiques abordées, mais aussi la manière dont elles le sont : est-ce que le client emploie des termes métier, est-ce que l'on retrouve la présence d'expressions d'humeur (agacement, énervement,

courtoisie). Ainsi, l'analyse des chaînes présentes la catégorie "Clôture" peut nous permettre d'obtenir des informations sur la satisfaction du client à l'issue de l'entretien (est-ce que l'on repère des formules traduisant le mécontentement du client ?), que l'on peut mettre en comparaison avec les chaînes trouvées dans la catégorie "Ouverture" (la conversation commence mal mais se termine par des remerciements par exemple). Ce genre d'informations sur la bonne ou mauvaise gestion d'une demande, le vocabulaire (expressions) employé peut notamment être utile dans un cas d'application tel que la formation des agents avec l'étude de stratégies de dialogue.

2.4.4 Conclusion

Dans cette section, nous avons présenté l'application de notre méthode d'extraction de chaînes caractéristiques des catégories dans un contexte de catégorisation thématique de segments conversationnels extraits d'enregistrements téléphoniques effectués en centre d'appels EDF.

Nous avons ainsi démontré que notre approche permettait de traiter des données complexes, tant sur le plan linguistique (discours spontané), que sur le plan structurel (conversations). L'avantage de cette méthode est sa robustesse par rapport aux données, ce qui n'est pas forcément le cas des méthodes symboliques d'extraction d'informations, pour l'introduction de chaînes porteuses de sens, qui permettent d'accéder à un certain niveau sémantique, contrairement aux méthodes classiques de catégorisation thématique.

Nous avons choisi ici de nous concentrer sur l'aspect catégorisation uniquement, en utilisant pour le corpus de test la segmentation fournie par les annotateurs. Bien entendu, dans une chaîne de traitement complète, ce problème de segmentation thématique serait aussi à gérer. Ceci fait partie des perspectives et nous envisageons de tester une adaptation des systèmes WLL (Sitbon et Bellot, 2007) qui prendrait en compte les mots et les chaînes afin de détecter les ruptures thématiques.

Les données que nous avons utilisées correspondent à des cas réels (corpus d'entreprise), avec tout ce que cela comporte de difficultés (hétérogénéité, répartition). Ainsi, étant donné que le corpus a été annoté par différentes personnes (car on a ici l'avantage de pouvoir obtenir de grandes quantités de données), les modèles ne peuvent pas être complètement discriminants : du bruit a été introduit par des mots ou expressions assignés à des catégories différentes par les divers annotateurs. C'est particulièrement le cas pour les frontières entre segments qui ne sont pas clairement définies : selon l'annotateur, une même phrase pourrait être rattachée à un segment ou au suivant. Il en va donc de même pour les mots composant cette phrase qui figureront dans le modèle de l'une ou l'autre des catégories ; imaginons (dans un cas extrême) que nous disposons de deux conversations assez similaires et qu'on retrouve dans les deux une même phrase située à la frontière entre deux catégories (cela peut être le cas par exemple si la discussion porte sur une modification de contrat - catégorie "Contrat" - qui serait demandée par le client car il souhaiterait disposer de plus de puissance : la conversation va donc se poursuivre par un segment "Aspects Techniques"), il est possible que

les deux annotateurs qui étiquetteront chacun une des conversations décident d'attribuer cette phrase frontière l'un à la première, l'autre à la seconde de ces catégories : au moment où l'on créera les modèles pour la catégorisation, le contenu de cette phrase se retrouvera simultanément dans les deux catégories. Sur ce même exemple d'enchaînement de thèmes dans la conversation, même s'il n'y avait pas plusieurs annotateurs pour l'ensemble du corpus, la phrase frontière peut tout simplement être à cheval sur deux catégories car des concepts appartenant aux deux sont mentionnés ("Mon activité ayant évolué, j'ai besoin d'une augmentation de puissance, je souhaiterais donc faire modifier mon contrat en conséquence"). Il faut donc décider d'attribuer cette phrase à l'une ou à l'autre, et des mots relatifs aux concepts de la première catégorie se retrouveront aussi dans le modèle de la seconde (et vice-versa).

Une variante de ce problème provient de la compréhension différente de certaines consignes par les annotateurs (ce qui ne se retrouve pas dans le cas d'un petit corpus qui serait étiqueté par la même personne tout le long), notamment sur les segments au cours desquels la conversation met en jeu deux agents. Certains des annotateurs ont décidé de doublement étiqueter ces segments, à la fois avec la catégorie "Conversation Agent-Agent" et avec la thématique dont ils discutent, alors que d'autres ont uniquement attribué la catégorie "Conversation Agent-Agent". Par ailleurs, on peut observer une répartition des segments dans les classes fortement déséquilibrées (contrairement à certains corpus académiques, où l'on peut retrouver une répartition relativement équilibrée des documents dans les catégories (ce sera le cas du corpus présenté au chapitre suivant)) : quasiment toutes les conversations commencent par un échange attribué à la catégorie "Ouverture" (473 segments pour les 480 appels du corpus d'apprentissage), alors qu'à l'opposé la catégorie "Projet de travaux" n'est présente que 12 fois. Hormis les problèmes que cela pose au niveau des classifieurs en tant que tels (les catégories très peu peuplées risquent d'être représentées par des modèles incomplets (exception faite des cas où la thématique possède un vocabulaire restreint et très spécifique et donc où peu d'exemples sont nécessaires pour la modéliser)), cela peut poser des questionnements sur la pertinence d'avoir un filtre de sélection des agglutinations qui est commun à toutes les catégories, celui-ci se basant sur les nombres d'occurrences des mots. Une idée intuitive pourrait être de régler le filtre proportionnellement à la taille de chacune des catégories. Mais là encore la réalité est toute autre : la taille du vocabulaire n'est pas forcément corrélée à la qualité des modèles, ainsi sur ce corpus nous avons besoin de peu de données pour modéliser correctement la catégorie "Ouverture" aussi bien pour l'extraction des chaînes que pour effectuer une catégorisation, alors que la même quantité de données serait insuffisante pour la catégorie "Aspects techniques" qui recouvre un ensemble de sujets plus vaste.

Du point de vue catégorisation, un cosinus pondéré a ici été proposé afin d'obtenir de meilleurs résultats sur ce corpus d'entreprise et de fournir un classifieur robuste bien que coûteux en temps d'estimation des paramètres, tant qu'on n'a pas trouvé une méthode permettant d'automatiquement définir un jeu optimal (nous avons pour les expériences précédentes testé des centaines de jeux différents, par dichotomie, en gravitant autour de certaines valeurs donnant des résultats intéressants, sans pour autant être sûr de ne pas être tombé sur un optimum local). Pour la suite des travaux, nous

ne conserverons pas ces pondérations et utiliserons des classifieurs plus légers à paramétrer. Ceci nous permettra d'évaluer tout de même l'apport de notre méthode sur des systèmes de catégorisation standards, sans pour autant chercher à obtenir le résultat de catégorisation optimal à quelques dixièmes de pourcents près. Néanmoins, il est à noter que ce type de classifieur peut permettre d'obtenir des résultats intéressants et qu'il pourrait être ré-utilisé pour d'autres tâches/dans d'autres contextes.

Par ailleurs, nos avons deux jeux de paramètres à définir : le filtre de sélection des agglutinations intéressantes et les pondérations du classifieur. Or, nous avons traité ces deux aspects de manière successive : dans un premier temps nous avons cherché un filtre intéressant pour les agglutinations, puis nous avons cherché un jeu de pondérations du cosinus qui maximise les scores de catégorisation à partir des chaînes ainsi extraites. Il eût peut-être été plus optimal de tenter de définir conjointement ces deux ensembles.

Sur le même thème, selon la tâche envisagée (si le but est d'obtenir les meilleurs résultats de catégorisation et que l'intérêt visuel des chaînes est secondaire) : étant donné que les paramètres du cosinus pondéré, qui ont été optimisés pour les chaînes de longueur maximale, donnent des meilleurs résultats à l'itération 3 (donc quand les chaînes ne sont pas maximales), il existe potentiellement un autre jeu de paramètres qui améliore encore le score obtenu à l'itération 3.

Nous avons aussi vu, lors de la présentation des résultats, que le critère *pGini* que nous utilisons, bien qu'ayant déjà montré son intérêt lors de travaux précédents, pourrait gagner à être adapté à ce contexte. Sa tendance à favoriser très fortement les éléments n'apparaissant que dans une catégorie est potentiellement perturbante dans un cas mutli-classes. C'est pourquoi nous réfléchissons à une variante un peu plus souple qui ne défavoriserait pas trop un élément présent par exemple dans deux classes uniquement, ce qui de plus pourrait aussi être pertinent dans un cas mono-classe dans le sens où cet élément, à défaut d'être discriminant, peut être considéré comme réfutant puisqu'il permet d'écarter les autres catégories.

Quelques perspectives spécifiques au type de corpus considéré sont envisagées afin d'améliorer la catégorisation. La première concerne le texte en lui-même : les agglutinations ne peuvent pas prendre en compte les disfluences. Le système recherche le patron exact, mais si la même chaîne est présente avec certains mots répétés, elle ne pourra pas être considérée. Autoriser la prise en compte de mots non forcément consécutifs dans les chaînes pourrait être une piste à explorer (une répétition ou une hésitation du type "euh" ne gênerait pas la reconnaissance des chaînes), à condition de réussir à ne prendre en compte que les disfluences et pas d'autres choses, telles que des négations qui inverseraient le sens de la phrase. La seconde perspective est liée à la nature des catégories proposées pour ce corpus. Nous avons pu voir qu'il y avait différents types de catégories (les thématiques et les structurelles). Il pourrait être intéressant de développer une stratégie permettant de traiter différemment ces deux types de catégories. Nous savons aussi que certaines catégories sont très souvent attribuées en même temps que d'autres, alors que certaines apparaissent toujours seules (c'est le cas de la catégorie "Ouverture"). On pourrait donc imaginer un système en plusieurs passes recherchant

en premier les catégories structurelles : si on a attribué "Ouverture" avec une forte probabilité on sait que ce n'est pas la peine de chercher une autre catégorie, alors qu'à l'inverse si on a attribué "Motif" on effectuera une deuxième passe pour ajouter une ou plusieurs catégories thématiques (la catégorie "Motif" correspondant au cas où le client exprime de lui-même le motif de son appel, elle est généralement liée à au moins une catégorie résumant ce motif). Il existe aussi un certain nombre de règles que l'on peut déduire de la description des classes, ainsi par nature, la catégorie "Motif" ne peut être attribuée en même temps que "Conversation Agent-Agent". Enfin, nous traitons ici les segments indépendamment les uns des autres, sans regard pour le déroulement de la conversation. Une amélioration possible serait de prendre en compte l'enchaînement des catégories tel qu'observé sur l'apprentissage : probabilités de transition d'une catégorie vers une autre, mais aussi dépendances à long terme (en fin de discussion, on revient sur le sujet initial).

2.5 Application dans une tâche d'identification du rôle du locuteur

Dans cette section, nous abordons le problème de l'identification du rôle du locuteur dans des enregistrements effectués en centre d'appel. Nous le traitons comme une tâche de catégorisation automatique de texte, en utilisant des classifieurs état de l'art prenant en compte ou non des chaînes caractéristiques des différents rôles, auxquels nous ajoutons une couche de modélisation du dialogue (enchaînement des rôles).

2.5.1 Description de la tâche - Le corpus CallSurf ManTransLoc-910

2.5.1.1 Introduction

L'identification du rôle du locuteur est une problématique de recherche récente, née au début des années 2000, qui a de fait été peu abordée. Elle est toutefois utile comme prémisse à un certain nombre d'autres tâches, telles que la recherche d'informations dans les documents audio, le résumé automatique (permettant par exemple d'indiquer à quel titre la personne s'est exprimée), l'analyse des interactions, du dialogue, ... La plupart des travaux s'intéressent à des données de type informations télévisées ou radiophoniques (les rôles sont à reconnaître parmi : présentateur, invité, journaliste, annonceur, annonceur météo, ...). Dans un cadre applicatif différent, on trouve les travaux de (Banerjee et Rudnicky, 2006), en contexte de réunions professionnelles, où les auteurs se basent sur l'extraction de mots-clés statistiquement corrélés à un des rôles (responsable de réunion, experts), pour atteindre 83% de précision de reconnaissance des rôles. L'analyse du contenu textuel est une des deux grandes approches suivies pour mener à bien cette tâche. Elle a notamment été explorée par (Barzilay et al., 2000) dans le contexte de journaux télévisés. Ce type de corpus mène à des méthodes différentes de celles que nous imaginons dans le cas de centres d'appels. Ainsi, dans (Barzilay

et al., 2000), les auteurs ont un traitement particulier pour les phrases situées aux frontières des tours de parole, dans la mesure où celles-ci fournissent potentiellement des informations discriminantes (le présentateur qui se... présente). Dans les conversations en centre d'appel, il y a généralement moins de locuteurs par enregistrement (dans la plupart des cas, un opérateur et un client, parfois un deuxième opérateur), mais les interactions sont beaucoup plus nombreuses (au maximum quelques phrases d'affilée pour un des protagonistes, souvent des questions-réponses) et les tours de paroles sont plus courts. Dans la même veine, (Liu, 2006) propose aussi une méthode basée sur les phrases frontières des tours de parole (un classifieur à maximum d'entropie utilisant les bigrammes et trigrammes de la première et la dernière phrase des tours) afin de reconnaître un des trois rôles présentateur, journaliste ou autre dans un corpus de journaux d'informations audio. Il combine cette approche avec une modélisation des successions de tours de parole sous forme de Modèles de Markov Cachés décodés par un algorithme de type forward-backward. Cette fusion permet d'atteindre 80% de précision.

La deuxième voie possible consiste en l'utilisation des signaux de parole uniquement : on n'utilise pas de transcription du contenu des échanges, ce qui permet de s'affranchir des erreurs commises par le système de reconnaissance automatique de la parole. Le précurseur dans ce domaine est (Vinciarelli, 2007) qui a testé deux méthodes. En se basant sur le signal acoustique, la première étape consiste à segmenter les conversations en tours de paroles et à "regrouper" ceux énoncés par le même interlocuteur (on assigne un même identifiant à ces tours). Partant de là, la première méthode utilise un réseau social modélisant les interactions entre les divers protagonistes [4] uniquement et aucune information sur le contenu de l'échange (que ce soit les mots prononcés, la prosodie, ...) - cette approche sera par la suite étendue dans (Salamin et al., 2009), par un réseau d'affiliations sociales. La deuxième méthode se base sur les durées des temps de parole de chacun des locuteurs. Enfin, ces deux méthodes reposant sur des "sources d'informations différentes", les auteurs émettent l'hypothèse qu'elles sont complémentaires (elles commettent des erreurs différentes) et qu'il est intéressant de les fusionner. Cette combinaison leur permet de correctement reconnaître le rôle de 85% du corpus (en termes de durée), sur un total de 96 enregistrements de 12 minutes pour 6 rôles possibles. Des travaux récents (Bigot et al., 2010) ont étudié la pertinence d'indices temporels (pour un locuteur : nombre de segments, leurs tailles ainsi que celles des inter-segments), prosodiques (fréquence fondamentale, positions et durées des instants parlés et des silences dont on estime le nombre par seconde) et acoustiques (force du signal) de bas niveau pour cette tâche. Les auteurs rapportent 85% de bonne reconnaissance sur 5 rôles (13 heures, soit 46 documents représentant différentes émissions de radio), les indices pertinents pour chacun des rôles étant différents.

Dans cette section, nous nous intéressons au problème de la reconnaissance du rôle (Client ou Agent) dans les enregistrements de conversations en centre d'appels EDF. Ce travail est motivé par le fait que ces enregistrements ont été effectués sur des appareils monocanaux (il n'y a pas de canal différent pour les clients et les agents, donc des moyens limités pour physiquement différencier la voix de l'un ou de l'autre). Retrouver

4. La personne A parle fréquemment après la B, qui elle-même interagit beaucoup avec la C ...

le rôle de la personne ayant prononcé chaque phrase est intéressant à plus d'un titre. Premièrement, cela permet d'étudier les interactions entre clients et agents, définir des profils de conversations, qui peuvent être utilisés pour de l'analyse marketing, l'amélioration de l'efficacité des centres d'appels, ... De plus, nous pensons que disposer de l'information sur le rôle du locuteur pourrait permettre d'améliorer la catégorisation des thématiques abordées en centres d'appels, par exemple par la création de modèles de langue spécifique à chaque couple (rôle ; thématique). Les clients et les agents n'utilisent en effet pas forcément les mêmes moyens d'expressions (vocabulaire, tournures) pour parler d'un même thème. L'utilisation de la méthode de recherche de chaînes caractéristiques est intéressante à ce titre puisqu'elle permet d'extraire des expressions couramment employées par l'un des deux types de protagonistes pour aborder un même sujet.

Nous avons choisi dans un premier temps de considérer cette tâche de reconnaissance du rôle comme un classique problème de catégorisation automatique de textes, en nous reposant sur l'idée que les indices linguistiques doivent être discriminants dans ce contexte. En effet, les agents employés en centres d'appels ont été formés sur la manière de répondre aux clients et emploient ainsi des termes et tournures de phrases que l'on ne retrouve pas forcément chez les clients. Ainsi, une précédente étude menée sur ce corpus (Cailliau et Poudat, 2008) a mis en valeur les différences au niveau du vocabulaire associé à chacun des deux rôles. Nous appliquons bien évidemment notre méthode de recherche de chaînes, afin d'extraire des expressions caractéristiques du client ou de l'agent. Puis, inspirés par les travaux de (Liu, 2006), nous ajustons les résultats de cette première catégorisation par la prise en compte de l'observation statistique des enchaînements de rôles dans les conversations. Cette modélisation du dialogue est testée pour un contexte local (au regard des deux tours de parole précédents) ou global (en décodant par un algorithme de Viterbi la séquence probable de rôles sur l'ensemble de la conversation).

2.5.1.2 CallSurf ManTransLoc-910

Le corpus *CallSurf ManTransLoc-910* utilisé ici est un sous-ensemble (différent de celui présenté en section précédente) du corpus CallSurf. Il s'agit de la transcription manuelle de 910 conversations, d'une durée moyenne de 6 minutes, enregistrées en centre d'appels EDF Pro. Les entretiens ont subi le même processus d'anonymisation et ont été manuellement segmentés en tours de paroles puis étiquetés selon le locuteur. On peut observer des recouvrements entre les locuteurs, provenant du fait que les conversations sont enregistrées sur un seul canal, il est impossible de distinguer les personnes qui parlent. Bien que cela ne soit pas systématique, un entretien débute généralement par l'appel d'un client (le premier locuteur de la conversation est alors un agent, puisque c'est lui qui décroche et dit bonjour) qui a une question à poser ou un problème à soulever. En cours de conversation, l'agent peut en appeler un autre (dans un autre service), mettant alors le client en attente, puis la conversation peut reprendre avec le client (une fois que l'information désirée a été obtenue auprès du second agent). Plus rarement, la conversation peut débuter par un appel d'un agent vers un client (le

client est alors le premier locuteur). Les rôles annotés dans une conversation peuvent ainsi être : Agent, Client, Client-Agent (correspondant aux cas de recouvrements - ces tours de parole sont généralement vides, car le contenu était inaudible et n'a donc pas pu être transcrit), Agent2 (dans le cas où l'agent en appelle un second, celui-ci est étiqueté de manière distincte), Client2 (lorsque le client transfère l'appel vers une autre personne de l'entreprise, par exemple au secrétariat qui possède le contrat, à la personne qui va prendre un rendez-vous, ...), Agent3, ... Un tour de parole ne correspond pas forcément à une phrase : il peut y en avoir plusieurs ou à l'inverse le locuteur peut être interrompu en milieu de phrase. De même, il n'y a pas obligatoirement une alternance parfaite des rôles (Agent/Client/Agent/Client) : les tours de parole sont limités en durée (une tirade trop longue pourra être représentée par plusieurs tours de parole), ou encore si une pause trop longue est observée par un locuteur son intervention sera segmentée en plusieurs tours. Une séquence de tours de parole peut par exemple être :
1 : Agent
2 : Client
3 : Agent
4 : Client
5 : Client
6 : Agent
7 : Client-Agent
8 : Client
9 : Agent
10 : Agent2
11 : Agent
12 : Agent2
13 : Agent
14 : Client
15 : Agent
16 : Client
17 : Agent

Nous avons, pour nos travaux, choisi de considérer trois rôles : Client, Agent et Client-Agent. Tous les sous-ensembles ont été regroupés (Agent, Agent2, Agent3, ... ont été changés en Agent). Les statistiques générales sur ce corpus sont présentés en tableau 2.5.

Nous avons choisi de ne pas lemmatiser ce corpus. En effet, lors des expériences préliminaires, nous avons obtenu des résultats de catégorisation supérieurs de 1 à 2% sans lemmatisation. Cela provient majoritairement du fait que les formes fléchies délivrent des informations quant au degré de formalité de la conversation : le tutoiement change la forme du pronom ainsi que la conjugaison du verbe, ce qui permet de reconnaître plus facilement une phrase prononcée par un agent lorsqu'il s'adresse à un autre agent. Les modèles sont adaptés par validation croisée en 10 sous-ensembles.

2.5.2 Expériences

2.5.2.1 Catégorisation par similarité cosinus

Notre première approche consiste à considérer l'identification du rôle du locuteur comme un problème classique de catégorisation automatique de textes. Les documents sont ici les tours de parole. Nous construisons un modèle par rôle (Client, Agent, Client-Agent). Puis, chaque tour de parole doit être rattaché à une de ces trois catégories, pour l'instant sans prise en compte du contexte (les tours de parole précédents ou suivants).

La catégorisation est effectuée à l'aide d'une normalisation cosinus des $TF \times IDF \times pGini$ (voir section 2.4.2, sans utilisation de pondérations. La similarité entre un tour de parole s et une catégorie (un rôle) r est calculée comme présentée en formule 2.7. $Cos(s,r)$ est la normalisation cosinus du vecteur document (tour de parole) W_s et du vecteur rôle W_r, où les dimensions sont les mots i :

$$Cos(s,r) = \frac{\sum_i (W_{is} \times W_{ir})}{\sqrt{\sum_i W_{is}^2 \times \sum_i W_{ir}^2}} \tag{2.7}$$

où :

$$\begin{aligned} W_{is} &= TF_s(i) \times IDF(i) \times pGini(i) \\ W_{ir} &= TF_r(i) \times IDF(i) \times pGini(i) \end{aligned} \tag{2.8}$$

$$pGini(i) = \sum_k P(k|i)^2 \tag{2.9}$$

$$IDF(i) = -\log\left(\frac{nombre\ de\ tours\ de\ parole\ contenant\ i}{nombre\ total\ de\ tours\ de\ parole}\right) \tag{2.10}$$

Cette première approche parvient à correctement reconnaître le rôle de 66% des tours de parole.

2.5.2.2 Modélisation du dialogue

Notre seconde approche consiste à ajuster les résultats fournis par la catégorisation à l'étape précédente par la prise en compte des successions de rôles dans les séquences de tours de parole observables sur l'apprentissage. Nous ajoutons donc une étape qui combine les sorties de la catégorisation par la similarité cosinus (c'est-à-dire les probabilités d'appartenance de chaque tour de parole aux différentes catégories représentant les rôles, les scores de similarité ayant été normalisés pour sommer à 1) avec la probabilité d'avoir chaque rôle sachant les rôles attribués aux tours de parole précédents. Nous testons pour cela deux méthodes : soit en considérant un contexte local, soit un contexte global.

Contexte local

Dans un premier temps, nous utilisons le corpus d'apprentissage pour estimer les probabilité des n-tours dans les séquences de successions de rôles (par exemple, la probabilité qu'un tour de parole soit étiqueté Agent, sachant que les deux précédents étaient déjà attribués à Agent - cela arrive par exemple dans le cas d'une conversation entre deux agents). Puis, pour assigner un rôle à un tour de parole, nous combinons cette information avec le score de similarité cosinus, comme présenté en formule 2.11.

$$\begin{aligned} Sim(s,r) &= \lambda_c \times Cos(s,r) \\ &+ \lambda_d \times (\lambda_u \times U_s + \lambda_b \times B_s + \lambda_t \times T_s) \end{aligned} \tag{2.11}$$

où :
$U_s = P(r(s)), B_s = P(r(s)|r(s-1))$
$T_s = P(r(s)|r(s-1), r(s-2)), \lambda_c + \lambda_d = 1, \lambda_u + \lambda_b + \lambda_t = 1$
$P(r(s)|r(s-1))$ est la probabilité d'avoir le rôle r pour le tour de parole s, sachant le rôle attribué au tour de parole précédent

Nous comparons les résultats obtenus avec un modèle de dialogue 2-tours (bitours) et un modèle 3-tours (tritours). Les bitours correspondent au cas où $\lambda_t = 0$. Nous avons, de manière empirique, fixé $\lambda_c = \lambda_d = 0,5$. De cette façon, la modélisation du dialogue a la même importance dans la décision que la similarité cosinus. Dans le cas des bitours, le rôle que le système propose pour un tour de parole est choisi en fonction des scores de similarité cosinus, la probabilité d'avoir ce rôle (sa fréquence dans l'absolu) et la probabilité d'avoir ce rôle connaissant le rôle du tour de parole précédent. Les paramètres optimaux que nous avons trouvés pour cette configuration - donc quand $\lambda_t = 0$ - sont $\lambda_u = 0,1; \lambda_b = 0,9$. On peut observer que la probabilité des bitours est bien plus influente dans la décision que celle des unitours (qui ne représente que la distribution des tours de parole dans les différents rôles). Dans le cas tritours, nous prenons aussi en compte la probabilité d'avoir un rôle connaissant ceux que l'on a déjà assignés aux deux tours de parole précédents. Les paramètres optimaux trouvés empiriquement sont alors : $\lambda_u = 0,1; \lambda_b = 0,45; \lambda_t = 0,45$.

La méthode bitours permet de reconnaître correctement le rôle de 76,6% des tours de parole, les tritours portent ce score a 78,8%. Nous ne pensons pas que considérer un contexte plus large (4-tours) apporterait une amélioration significative. Les tritours aident le système à détecter les morceaux de discussions entre deux agents (les deux tours de parole précédents ont été attribués à la catégorie Agent) ou entre un agent et un client (alternance de Client et Agent). En réalité, les principales erreurs produites par cette méthode proviennent d'un effet boule de neige : quand la mesure de similarité attribue un mauvais rôle avec un score très élevé, cela entraîne une confusion du modèle de dialogue. Si cette erreur est commise sur le premier tour de parole de la conversation, cela peut mener à mal catégoriser l'ensemble de l'entretien (dans le cas

où aucun des tours suivants n'est assigné par le cosinus à sa bonne catégorie avec une probabilité suffisamment forte pour réussir à redresser la barre).

Contexte global

La seconde méthode que nous avons testée pour la modélisation du dialogue consiste en l'utilisation de l'algorithme de Viterbi (Viterbi, 1967) pour combiner la mesure de similarité avec les probabilités de successions de rôles. Le but de l'algorithme de Viterbi est de trouver la séquence d'états (rôles) la plus probable en considérant la conversation dans son intégralité (permettant ainsi d'éviter l'effet boule de neige décrit précédemment). Dans notre cas, un état réfère à un rôle, connaissant le précédent. Ainsi, pour chaque tour de parole, on calcule la probabilité d'avoir chaque rôle, pour chaque origine possible (formule 2.12).

$$\begin{aligned} &State\big(r(s), r(s-1)\big) = \\ &\quad \underset{r(s-2)}{\operatorname{argmax}} \Big(\lambda_d \times \log\Big(P\big(r(s)|r(s-1), r(s-2)\big)\Big) \\ &+\lambda_c \times \log\big(Cos(s,r)\big) + State\big(r(s-1), r(s-2)\big) \Big) \end{aligned} \tag{2.12}$$

où : $r(s)$ est le rôle du tour de parole courant
$s-1$ est le tour de parole précédent
$P(r(s)|r(s-1), r(s-2))$ est la probabilité d'avoir r dans l'hypothèse où les rôles $r(s-1)$ et $r(s-2)$ ont été attribués aux tours de parole précédents (fréquence du tritour calculée sur le corpus d'apprentissage)

Les coefficients lambdas ont été empiriquement fixés à $\lambda_d = 0,2$ et $\lambda_c = 0,8$. Cette configuration permet d'obtenir 91,3% de rôles correctement reconnus. Cette méthode surpasse la précédente (modélisation du dialogue dans un contexte local), notamment car l'attribution de rôle pour tous les tours de parole ne dépend pas du premier.

2.5.2.3 Chaînes caractéristiques des rôles

Nous avons appliqué la méthode présentée en section 2.2 pour extraire des chaînes de mots caractéristiques des différents rôles (les catégories). Nous avons choisi de conserver les propositions présentant un LRV > 75 et présentes plus de 10 fois en itérant la procédure jusqu'à ce que plus aucune nouvelle proposition ne soit retournée. Nous présentons ici quelques-unes des chaînes obtenues pour les deux principaux rôles (les segments étiquetés Client-Agent étant trop peu nombreux et souvent vides, aucune chaîne n'a été proposée par le système). Parmi celles liées aux Agents, on trouve les introductions typiques : *"sylvie-xx-edf-pro"*, *"xx-edf-pro-bonjour"*, *"edf-pro-jérôme-xx-à-votre-service-bonjour"* ; des formules de politesses de fin d'entretien : *"bonne-journée-merci-de-votre-appel-au-revoir"* ; entre autres expressions formelles : *"vous-remercie-de-patienter-un-*

petit-instant" ; des expressions techniques : "*pénalités-de-retard*", "*votre-historique-réel-de-consommation*", "*vous-envoie-autorisation-de-prélèvement*". D'après les tours de parole des Clients, on récupère plusieurs expressions utilisées pour expliquer les raisons de l'appel : "*vous-appelle-parce-que*", "*c'-est-pour-ca-que-je-vous-appelle*", "*le-problème-c'-est-que*", "*me-permets-de-vous-appeler*" ; des chaînes intéressantes pour la gestion de la relation client pourraient être celles liées aux problèmes non-résolus, par exemple : "*un-de-vos-collègues*", que l'on trouve généralement dans des phrases du type "*La dernière fois que j'ai appelé, un de vos collègues m'a dit que ...*", si on repère un certain nombre de fois ce genre de chaînes, on peut en déduire que l'on a un problème pour répondre de manière satisfaisante aux requêtes des clients (ou en étudiant leurs fréquences à différentes périodes, on peut analyser l'évolution de l'efficacité du centre d'appel). De la même façon, on peut voir quelles sont les questions récurrentes des clients à divers moments. Ces chaînes permettent par ailleurs de faire ressortir des thématiques sans que le corpus ne soit étiqueté en ce sens : une expression ou en ensemble d'expressions fréquemment retournées peut ainsi correspondre à un thème. De plus, elles peuvent mettre en valeur la différence de vocabulaire employé par les clients ou les agents : un service dont le nom est déformé par les clients, les expressions métier sont-elles utilisées par les clients ("point-de-livraison") ? ...

Nous avons évalué l'influence de ces chaînes sur l'efficacité de la catégorisation. L'usage des agglutinations dans le cas de la catégorisation avec cosinus a permis d'arriver à 68,1% de tours de parole correctement reconnus (soit un gain de 2,1%). Leur utilisation dans le cas de la combinaison entre le cosinus et la modélisation du dialogue avec Viterbi a mené à 92% de réussite (+0,7% par rapport à la même méthode ne prenant pas en compte les chaînes).

2.5.2.4 Algorithme de Viterbi pondéré selon la longueur des tours

D'après les expériences précédentes, nous observons que :

- l'étape de catégorisation classique est réellement efficace quand les tours de parole sont suffisamment longs (mais souvent peu fiable quand il n'y a que un ou deux mots) ;
- si les longs tours de parole sont correctement catégorisés avec une confiance élevée (la mesure de similarité cosinus penche fortement vers un des rôles relativement aux autres), l'algorithme de Viterbi pourra aisément déduire les rôles qui auraient dû être assignés aux plus courts.

Partant de ces observations, nous avons transformé la formule 2.12 afin qu'elle prenne en compte la taille des tours de parole. Le résultat de la catégorisation cosinus est maintenant pondéré par la taille du tour, comme présenté en formule 2.13.

$$
\begin{aligned}
&State\big(r(s), r(s-1)\big) = \\
&\quad \underset{r(s-2)}{\operatorname{argmax}} \Big(\lambda_d \times \log\Big(P\big(r(s)|r(s-1), r(s-2)\big)\Big) \\
&\quad + \mu \times \log\big(Cos(s,r)\big) + State\big(r(s-1), r(s-2)\big) \Big)
\end{aligned} \tag{2.13}
$$

où : $\mu = \lambda_c + \dfrac{length(s)}{W_s}$

$length(s)$ est le nombre de caractères du tour de parole s (sans espaces)

Après expériences, W_s a été fixé empiriquement à 12 et les lambdas à $\lambda_d = 0,88$ et $\lambda_c = 0,1$.

Ce système atteint 93% de bonne reconnaissance du rôle du locuteur.

2.5.2.5 Baselines

Nous avons comparé nos approches à trois systèmes :

- pour la première étape de catégorisation de textes, nous avons testé des Machines à Vecteurs de Supports avec un noyau linéaire, nous avons utilisé l'outil Liblinear (Fan et al., 2008), les paramètres optimaux (C=1 et $\varepsilon = 0,0625$) ont été trouvés par une recherche en grille, avec C variant de 2^{-5}, 2^{-4}, ... à 2^{15} et ε dans [2^{-15}, ..., 2^{3}]. Ces SVM catégorisent correctement 69,1% des tours de parole, c'est-à-dire 3,1% de mieux que le cosinus avec mots isolés et 1% de mieux que le cosinus avec agglutination de chaînes. Ce qui confirme qu'une simple catégorisation de textes appliquée à ce problème atteint des résultats aux alentours de 70%.
- pour l'étape de modélisation du dialogue, nous avons testé une règle de décision triviale assignant tour à tour Agent et Client aux tours de parole (un tour sur deux), en commençant par un Agent, étant donné que nous avons observé que c'était le cas de la majorité des conversations. Cette règle étiquette correctement 71,1% des tours de parole. Nous pouvons en déduire qu'une modélisation du dialogue plus sophistiquée est effectivement nécessaire et que l'étape de catégorisation de textes reste utile.
- enfin, nous avons testé la combinaison entre le classifieur SVM et l'algorithme de Viterbi. Cela amène à 90,1% de bon étiquetage. Ce qui est inférieur de 1,2% à la combinaison équivalente cosinus+Viterbi et de 2,9% à la combinaison du cosinus utilisant les mots agglutinés avec le Viterbi pondéré par la longueur des tours de parole. On note que le couple SVM+Viterbi est moins bon que Cosinus+Viterbi alors que sans modélisation du dialogue, les SVM donnaient des meilleurs résultats que le cosinus. Cela s'explique par le fait que les SVM prennent ici des décisions plus tranchées, y compris quand ils se trompent et que ces mauvaises décisions sont prises avec un score de confiance élevé qui ne peut pas forcément être rattrapé par la modélisation du dialogue.

2.5.3 Résumé des résultats

En tableau 2.6 nous résumons les résultats obtenus (pourcentage de bonne reconnaissance du rôle du locuteur) par les méthodes présentées dans les sections précédentes. Pour l'ensemble d'entre elles, il y a environ 0,5% d'erreurs dues à la mauvaise catégorisation des tours de parole "Client-Agent", qui sont sous-représentés dans le corpus et ne contiennent généralement aucun texte. Ils ont donc des mauvais modèles et sont assez difficiles à reconnaître.

2.5.4 Conclusion

Dans cette section, nous avons présenté plusieurs approches pour l'identification du rôle du locuteur. Dans un premier temps, nous avons traité ce problème comme une tâche de catégorisation automatique de textes, menant par diverses méthodes à des résultats compris entre 66 et 69,1% de bonne reconnaissance. Nous avons ensuite tenté différentes méthodes pour combiner ces premiers résultats avec une modélisation de la structure du dialogue (modèle de succession des locuteurs). L'utilisation d'un algorithme de Viterbi nous a ainsi permis de reconnaître plus de 90% des rôles. Nous avons aussi ajouté un module d'extraction et agglutination de chaînes de mots caractéristiques des différents locuteurs. Celles-ci permettent d'améliorer quelque peu les résultats de l'identification et fournissent des informations intéressantes pour l'analyse des conversations.

Notre méthode finale, qui intègre la prise en compte de la longueur des tours de parole afin de se reposer pour leur étiquetage plutôt sur la catégorisation de texte (quand le tour de parole est suffisamment long) ou plutôt sur la modélisation du dialogue (tour peu fourni) reconnaît correctement les rôles de 93% des tours de parole dans notre contexte de conversations en centre d'appels. Il sera intéressant par la suite de tester ce système dans le cas où il y a plus de rôles impliqués dans les entretiens (et donc où l'on peut s'attendre à ce que la modélisation du dialogue soit moins efficace) et dans le cas d'un corpus automatiquement transcrit par un système de reconnaissance automatique de la parole (où la première étape de catégorisation de texte pourrait se baser sur des modèles plus bruités). Par ailleurs, nous nous sommes ici reposé sur la segmentation manuelle en tours de locuteurs, qui n'est pas forcément identique à ce que produirait une segmentation automatique ; celle-ci, à moins d'être parfaite, commettrait quelques erreurs, qui conduiraient à intégrer des mots prononcés par un des locuteurs dans le modèle du rôle opposé.

Nous considérons que cette méthode atteint des résultats suffisamment élevés pour être réutilisés dans l'optique d'améliorer d'autres tâches. Ainsi, nous envisageons d'utiliser cette reconnaissance du rôle du locuteur dans un contexte de catégorisation thématique d'entretiens, en créant des modèles thématiques spécifiques à chaque rôle. Nous pensons que l'utilisation de modèles propres à la fois à un rôle et une thématique pourrait être plus efficace que des modèles uniquement thématiques, dans la mesure où il a été observé que les clients et les agents n'utilisaient pas les mêmes expressions pour

parler d'un même sujet.

2.6 Conclusion

Cette partie conclut les travaux présentés dans ce chapitre sur l'extraction et l'utilisation de chaînes caractéristiques à des catégories appliquées sur des transcriptions de conversations en centre d'appels EDF, à la fois dans un contexte de catégorisation thématique et de reconnaissance du rôle des locuteurs.

Dans ce chapitre, nous avons présenté une méthode d'extraction de chaînes de mots caractéristiques des différentes catégories présentes dans un corpus. Nous avons expliqué comment et pourquoi les intégrer dans des systèmes de catégorisation automatique de textes et avons démontré leur intérêt pour améliorer les résultats d'une catégorisation de segments de conversations selon les thématiques qui y sont abordées, ainsi que pour la reconnaissance du rôle du locuteur (client ou agent). Nous avons montré des exemples de ces chaînes, qui peuvent ainsi être employées pour extraire automatiquement des informations, telles que les remarques fréquentes, les points les plus discutés, les styles d'expression, ...

Bien que le corpus provienne d'enregistrements téléphoniques, nous avons travaillé à partir de transcriptions manuelles. Un cas plus réel (et efficace) serait d'appliquer nos méthodes sur les sorties d'un système de reconnaissance automatique de la parole. Dans ce cas, on peut imaginer que les modèles seraient plus bruités en raison des erreurs sur les mots : il sera plus difficile de reconnaître une chaîne, sous réserve que le même mot prononcé ne soit pas toujours transcrit par le même mot en sortie du système de reconnaissance (si "bancaire" est toujours reconnu comme "babar", les chaînes extraites seront de la forme "le numéro de votre carte babar", ce qui n'empêchera pas sa reconnaissance, bien que la compréhension n'en soit pas facilitée). Étant donné qu'il s'agit de parole spontanée et que les locuteurs sont en permanence différents, on pourrait supposer que les erreurs ne conduiront pas toujours à la même transcription. Cependant, des études (Danesi et Clavel, 2010b) effectuées sur une partie du corpus CallSurf, transcrit automatiquement par le système de reconnaissance de Vecsys, ont montré qu'on y retrouvait un certain nombre d'erreurs récurrentes : sur la reconnaissance des entités nommées ("Madame Maubert" devenant "Madame ampère"), sur certains homophones ("moi" vs "mois") ou sur des combinaisons de mots incluant des disfluences ("l'heure" pour "leur euh"). De même, "l'abonnement" pourra de manière fréquente devenir "la bonne non" et sera intégré comme tel dans les chaînes.

Nos travaux futurs porteront sur l'étude des cas de chevauchements. Notamment, comment choisir d'agglutiner une chaîne plutôt qu'une autre lorsqu'on en a le choix : dans la phrase "mot1 mot2 mot3 mot4 mot5", si on connaît les patrons "mot1 mot2 mot3" et "mot3 mot4 mot5", lequel est le plus intéressant ? Lequel présente un meilleur intérêt visuel ou en termes de catégorisation, à la fois à cette étape, mais aussi à la suivante, puisque si on choisit le second patron, on s'interdit potentiellement de créer ultérieurement "mot1 mot2 mot3 mot4".

Par ailleurs, au vu du corpus ici considéré, il nous semble pertinent de nous pencher sur une méthode autorisant une certaine tolérance dans l'application des chaînes, par la prise en compte d'insertions, substitutions, délétions, afin de tenir compte des disfluences. On pourrait par exemple imaginer la chaîne représentée par un automate dans lequel "euh" correspondrait à une boucle ne changeant pas d'état. Ceci devrait, dans l'idéal, être calculé automatiquement (sans qu'il n'y ait besoin d'ajouter des connaissances sur les différents types de disfluences possibles) tout en conservant la qualité des chaînes extraites.

Catégorie	Description	App	Test1	Test2	Test3
Ouverture	Ouverture de la conversation, présentations, formules de politesse	473	74,9	73,4	84,4
Clôture	Fin de la conversation, formules de politesse	415,5	59,9	68,3	71,9
Motif	Le client exprime de lui-même son problème	238,1	42,9	41,4	44,9
Conversation Agent-Agent	Conversation entre deux agents (le client est en attente)	219,5	46,5	55	88,5
Récapitulatif	L'agent récapitule ce qu'il va faire	300,3	42,2	37,3	44,1
Référence Client	Le client donne sa référence EDF, vérifications de l'agent	350,8	66,1	67,6	85,4
Coordonnées Client	Prise de coordonnées du client (nom, adresse, tel, email, société, etc.)	277,8	35,1	41,8	50,1
Rendez-vous	Prise de rendez-vous	123,9	36,6	18,5	35,1
Facture-Paiement	Problèmes liés à la facture (règlement, impayés, duplicata, contestation, compréhension, etc.)	302,7	51,3	35,6	44,2
Contrat	Contrat (question, ouverture, résiliation, modification, etc.)	673	134,4	121,8	154,2
Aspect Technique	Puissance, ampérage, qualité fourniture, coupure intempestive, etc.	102,8	15	10,5	26,5
Discussion Commerciale	L'agent explique les différentes offres, types de contrats, services, etc.	79	23,6	4,5	20,5
Intervention Technique d'EDF	Conversation traitant d'une intervention pour panne, mise en service, raccordement etc.	120,5	36	28,7	36
Relève de compteurs	Conversation traitant de la relève de compteur p. ex. en cas de résiliation)	74,7	19,1	5	18,6
Projets de Travaux	Informations techniques et discussions en vue de la réalisation de travaux	12	2	1	4
Facilité de contact	Problèmes relationnels, difficultés de contact, de traitement de la demande	243,8	84,9	29,2	81,7
Proposition de Service	Mensualisation, prélèvement, assistance-dépannage, etc. et réponses du client	93,5	16,6	8	12,1
Divers	On ne peut pas assigner d'autre thème (p. ex. le client explique son métier)	263,8	110,2	37,5	37,5
Vide	Tours de parole vides et autres	48,3	4	1	0

TABLE 2.1 – *Les 19 catégories du corpus CM570 et le nombre de segments par catégorie. Un segment étiqueté avec plusieurs catégories compte pour* $1/(\textit{nombre de catgories})$ *pour chacune d'elles.*

	App	Test1	Test2	Test3
nombre de segments	4413	1001	686	941
nombre total de mots	567 361	109 715	109 715	109 715
nombre de lemmes uniques	5953	2887	2887	2887
nombre de mots dans le segment le plus court	2	2	2	2
nombre de mots dans le segment le plus long	2187	1573	1799	1315
nombre moyen de mots par segment	128,6	109,6	159,9	116,6
nombre moyen de segments par conversation	9,2	11,5	7,9	10,8

TABLE 2.2 – *Statistiques générales sur le corpus CM570.*

Itération	Nombre de modèles
1	0
2	702
3	1152
4	1299
5	1333
6	1335
7	1335

TABLE 2.3 – *CM570 - Nombre de modèles de chaînes utilisés sur le corpus de test.*

Nombre de mots	Nombre de modèles
2	776
3	334
4	149
5	40
6	16
7	12
8	4
9	3
11	1

TABLE 2.4 – *CM570 - Nombre de modèles de chaînes selon leur taille (en nombre de mots), à la dernière itération.*

nombre de caractères	5 127 380
nombre total de mots	995 893
nombre de mots uniques	12 716
nombre moyen de mots par conversation	1 094,4
nombre moyen de mots par tour de parole	9,8
nombre de tours de parole par rôle :	
- Agent	55 848
- Client	45 386
- Client-Agent	513

TABLE 2.5 – *Statistiques générales sur le corpus CallSurf ManTransLoc-910.*

Méthode	**%**
Catégorisation par cosinus (TF x IDF x pGini)	66
Cosinus + agglutinations	68,1
Catégorisation par SVM	69,1
Cosinus + modélisation du dialogue bitour	76,6
Cosinus + modélisation du dialogue tritour	78,7
Cosinus + Viterbi	91,3
SVM + Viterbi	90,1
Cosinus + agglutinations + Viterbi	92
Cosinus + agglutinations + Viterbi pondéré par la taille	**93**
assignation triviale : alternance Agent/Client	71,1

TABLE 2.6 – *Résultats en pourcentage de reconnaissance du rôle du locuteur sur le corpus CallSurf ManTransLoc-910.*

Chapitre 3

Des collocations, des chaînes caractéristiques et des opinions

Sommaire

Résumé

Nous appliquons ici la méthode présentée au chapitre précédent, mais cette fois-ci dans un contexte de détection d'opinion. Nous recherchons donc des chaînes caractéristiques des différentes opinions exprimées dans un corpus. Nous testons l'impact de ces chaînes sur un système de catégorisation de textes selon l'opinion globale et en proposons des exemples. Dans la mesure où notre méthode est relativement généralisable à différents langages et types de corpus, nous l'expérimentons tour à tour sur un corpus de critiques de jeux vidéos en français, des critiques de films en anglais et une enquête de satisfaction effectuée auprès de clients EDF.

3.1 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 (e-reputation). 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.

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 le plus souvent la tendance de l'avis exprimé dans un texte, est-elle positive, négative, positive et négative, neutre (selon le contexte, "neutre" peut signifier soit qu'aucun avis n'est exprimé dans le texte, soit que l'on y trouve à la fois des avis positifs et des avis négatifs) ? Les travaux au niveau du document ont été 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 afin de connaître les opinions exprimées 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). De nombreux travaux abordent ce problème comme une tâche de catégorisation automatique de textes : il s'agit donc, à partir de l'ensemble des mots composant le document de le comparer à des modèles représentatifs des différentes catégories connues, qui seront dans ce cas les opinions globales des documents (positif, négatif, neutre). À ce titre, (Ng et al., 2006) présentent des travaux portant sur deux tâches au niveau du document, effectués sur un corpus de critiques de films sur lequel nous appliquerons aussi nos méthodes :
  - l'identification de critiques, c'est-à-dire reconnaître si un document est une critique ou non : dans ce cas, le corpus a été enrichi de 2000 documents portant sur le thème du cinéma mais qui ne sont pas des critiques de films (résumés, publicités, ...) ; sur cette tâche un classifieur SVM (Machines à Vecteurs de Supports) utilisant les unigrammes offre une précision de 99,8%.
  - la détection de la polarité du document, vue comme une catégorisation automatique des critiques en tant que "Positives" ou "Négatives", effectuée là aussi

avec un classifieur SVM, pour lequel plusieurs types de vecteurs ont été testés : en premier lieu, les auteurs ont considéré les 10 000 unigrammes les plus représentatifs de chaque opinion (d'après un calcul de logarithme de rapport de vraisemblance pondéré) qui permet de correctement catégoriser 87,1% des 2000 critiques ; le fait de prendre en considération 5000 bigrammes et 5000 trigrammes (sélectionnés selon la même méthode) en plus des unigrammes, permet de gagner 2,1% de précision ; l'ajout (manuel) de connaissances sur la polarité de certains adjectifs permet de porter ce score à 90,4% ; en revanche, le fait de retirer des vecteurs les n-grammes considérés comme objectifs ou la prise en compte de relations de dépendances - des couples du style (like ; movie) - ne permettent pas d'améliorer significativement les résultats.

- au niveau de la phrase, deux problématiques ont été fréquemment abordées dans la littérature : la détection de subjectivité, consistant à décider si une phrase est porteuse d'opinion ou non - on peut citer (Riloff et Wiebe, 2003) qui utilisent des patrons syntaxiques aidés de classifieurs afin de retrouver des expressions subjectives qui serviront à décider si on peut considérer que la phrase l'est ; l'autre problématique concerne la catégorisation de phrases selon la polarité, comme cela a par exemple été fait avec des SVM entraînés sur des phrases manuellement étiquetées dans (Bossard et al., 2008). Ces deux problématiques sont bien entendu complémentaires : il peut s'agir de décider si une phrase est subjective, puis de reconnaître l'opinion qui y est exprimée le cas échéant.
- les approches plus détaillées (à l'échelle du mot 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), (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.
- le ciblage précis d'une expression d'opinion (quelle est l'opinion exprimée sur quoi en général et sur quel point précis par qui et quand ?) est abordé au chapitre suivant.

Les différents niveaux de granularité se retrouvent souvent mêlés : la détection de syntagmes porteurs d'opinion pour catégoriser les phrases, l'utilisation de sous-chaînes pour la polarité des documents, ... On peut ainsi citer les travaux de (Saggion et Funk, 2009), où les auteurs testent plusieurs indices (mots, lemmes, POS, polarité des mots obtenue en se basant sur SentiWordNet (Esuli et Sebastiani, 2006), ...) pour attribuer une étiquette positive ou négative à un document (ou, sur un deuxième corpus, une catégorisation plus précise selon une échelle de 5 niveaux d'opinions) tout en offrant la possibilité d'extraire des syntagmes exprimant une opinion, tels que "a very efficient management", "the interesting thing" ou "the most shockingly service".

Pour notre part, nous ne traitons pas le problème au niveau de la phrase, qui est une construction syntaxique mais qui peut correspondre à plusieurs vues sémantiques. Ainsi, dans la phrase "Dans les haricots verts, j'aime bien le vert, mais pas trop les haricots", deux opinions opposées sont exprimées. Dans ce cas, la détection de subjectivité peut être intéressante dans un souci d'extraire les phrases porteuses d'opinions, mais

nous pouvons aussi constater qu'attribuer une polarité globale à cette phrase paraît difficile. Nous travaillons dans ce chapitre à deux niveaux : au niveau du document pour détecter l'opinion exprimée globalement et nous appliquons la méthode de recherche de chaînes présentée au chapitre précédent, avec les deux mêmes objectifs :

1) étudier l'impact d'éléments potentiellement plus discriminants que des mots isolés sur les résultats de la catégorisation du document. Les travaux de (Dave et al., 2003) sont à ce titre intéressants : dans un contexte de catégorisation de commentaires concernant des produits électroniques et informatiques, les auteurs ont testé différentes données d'entrée du classifieur. Partant des unigrammes, ils ont tenté les modifications suivantes :

- des substitutions de certains termes par un label plus générique (basées sur des meta-données ou des observations statistiques) : les nombres, les noms de produits, des mots à faible occurrence, apparaissant dans un contexte similaire. Ces substitutions n'ont pas amélioré les performances du classifieur, voire dans certains cas les ont dégradées.
- des substitutions basées sur des approches linguistiques : la recherche des POS (catégories morpho-syntaxiques) des mots et de relations de dépendances entre les mots d'une même phrase (que les auteurs jugent "computationally-expensive"), l'utilisation de WordNet pour trouver des similarités sémantiques (mais dont l'utilisation est restreinte en raison des problèmes d'ambiguïté proposant plusieurs synsets par mot, ce qui rajoute du bruit - de plus, cela a conduit à utiliser des vecteurs d'entrée de trop grande taille), l'utilisation de collocations de type adjectif-nom non-forcément contiguës. Ces expériences n'ont pas non plus été concluantes.
- la racinisation des mots, qui a donné des résultats mitigés (amélioration sur un jeu de test, dégradation sur l'autre). La raison avancée étant la sur-généralisation, notamment car les critiques négatives seraient plus souvent écrites au passé, une fois que le client a rapporté le produit.
- remplacement des mots suivant des négations par des mots négationnés ("not good or useful" devient ainsi "NOTgood NOTor NOTuseful"), basée sur de précédents travaux (Pang et al., 2002) rapportant un léger gain par cette méthode. Mais dans ce cas, ceci n'a pas été couronné de succès.
- l'utilisation de n-grammes, qui a apporté des améliorations significatives (les trigrammes pour un des corpus, plutôt les bigrammes pour le second). Cependant, cela n'a fonctionné que dans le cas où ces n-grammes étaient utilisés seuls, sans inclure des n-grammes de degré inférieur (par exemple la prise en compte des unigrammes en même temps que les bigrammes dégrade les résultats).
- la proximité : rassemblement dans une même dimension des mots rencontrés dans un intervalle proche (opérateur NEAR). Les performances sont améliorées mais cependant moins qu'avec les trigrammes.
- la prise en compte de sous-chaînes de taille non-fixe (sélectionnées selon des critères tels que le gain d'information, le nombre de documents dans lesquels elles apparaissent, ...), une des principales difficultés évoquée provenant des problèmes de couverture (plus les chaînes sont longues plus elles ont des chances d'être plus discriminantes, mais sont moins représentées). Cette approche amé-

liore les résultats de catégorisation, dans certains cas plus que les trigrammes.

Les observations consécutives à ces travaux, recoupant celles de (Ng et al., 2006), nous laissent à penser que notre approche pourrait être intéressante pour améliorer la catégorisation de critiques en opinions : notre méthode va dans le même sens que leurs conclusions positives (les chaînes que nous proposons effectueront un travail similaire à leur utilisation de n-grammes ou de sous-chaînes et permettent de prendre en compte en partie les négations sans créer d'heuristique spécifique), tout en évitant celles de leurs modifications qui n'ont pas aidé (méta-données, dictionnaires, dépendances), bien que nous ayons déjà décidé de les écarter pour d'autres raisons (conserver au maximum la généricité de la méthode).

2) pouvoir présenter les chaînes caractéristiques de chacune des opinions présentes dans les textes, qui permettent ainsi :

- soit d'un point de vue global, d'avoir un aperçu des éléments que l'on retrouve généralement dans les textes d'opinion ;
- soit dans un texte précis, de mettre en valeur les chaînes présentes que l'on a reconnues.

Les chaînes sont ici caractéristiques des textes porteurs d'une opinion et l'on pourra donc y retrouver des expressions d'une opinion ("vraiment-peu-ragoûtant") ou des éléments fréquemment énoncés dans les textes représentant cette opinion (un service beaucoup discuté par exemple). Comme pour le chapitre précédent, ces chaînes sont *caractéristiques* d'une opinion et pas obligatoirement *discriminantes*, dans le sens où une même chaîne peut être extraite pour plusieurs des catégories connues (elles sont calculées en considérant isolément les textes de chacune des opinions, il n'y a pas de comparaison par rapport aux textes rattachés aux autres opinions). Nous avons délibérément choisi cette approche car pour nous cela présente des intérêts (une chaîne trouvée pour deux opinions sur les trois possibles présente un apport pour la catégorisation en permettant de réfuter la troisième, de plus cela permet d'extraire aussi des expressions propres au domaine et visualiser leur représentation dans les différentes opinions). Cependant, la méthode peut être adaptée dans le cas où on souhaiterait obtenir les chaînes réellement discriminantes (un moyen simple à mettre en oeuvre étant la soustraction d'ensembles, pour rechercher les chaînes que l'on ne retrouve que dans une des catégories).

La méthode que nous suivons étant probabiliste, n'utilisant pas de ressources linguistiques (à l'exception de la lemmatisation, que nous pouvons considérer comme triviale dans un certain nombre de langues et qui de plus n'est pas indispensable), cela lui permet d'être appliquée sur des corpus présentant des caractéristiques différentes. Nous la testons ici tour à tour sur : des critiques de jeux vidéos en français, une enquête téléphonique de satisfaction de clients EDF et des commentaires en anglais laissés par des internautes à propos de films.

3.2 Protocole

Dans cette section, nous décrivons les expériences que nous avons réalisées pour l'utilisation de chaînes dans un contexte de catégorisation de textes selon l'opinion qui y est exprimée.

La méthode générale (recherche de chaînes et application dans un cadre de catégorisation automatique) est la même que celle présentée en section 2.3, à ceci près que les catégories considérées sont ici des opinions. Cependant, les classifieurs employés sont différents et nous avons par ailleurs testé une méthode de repli sur les chaînes.

3.2.1 Classifieurs

Dans un premier temps, 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 $TF \times IDF$ des mots ou chaînes i les composant. Les TF correspondent ici en réalité à une mesure binaire de la présence ou non d'un mot dans le document, ceci afin de tenir compte des spécificités 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 - par opposition au chapitre précédent où l'influence de ces disfluences pouvait être réduite par l'application d'une pondération). 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 le critère de pureté de Gini $pGini$ (voir section 2.4.2).

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 des valeurs intéressantes 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.

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 cosinusGini.

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.

3.2.2 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 chaînes courtes 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 étaient présents 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).

3.2.3 Critère d'évaluation

L'impact sur la classification est évalué par F-score moyen (avec $\beta = 1$ dans la formule 3.1), c'est-à-dire en utilisant les macro-moyennes des scores de précision 3.2 et rappel 3.3 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).

$$F_{score}(\beta) = \frac{(\beta^2 + 1) \times Precision \times Rappel}{\beta^2 \times Precision + Rappel} \tag{3.1}$$

$$Precision = \frac{\sum_{i=1}^{n} Precision_i}{n} \tag{3.2}$$

$$Rappel = \frac{\sum_{i=1}^{n} Rappel_i}{n} \tag{3.3}$$

3.3 Expériences

Dans cette section, nous décrivons les trois corpus sur lesquels nous avons appliqué notre méthode, ainsi que les résultats obtenus.

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

- pour 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 ;
- 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.

3.3.1 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 3.1, tandis que le tableau 3.2 présente quelques statistiques sur la dimension du corpus.

	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

TABLE 3.1 – *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

TABLE 3.2 – *Statistiques sur le corpus Deft07, les comptes des mots incluent les symboles et signes de ponctuation*

Nous avons utilisé LiaTagg [1] afin d'obtenir les formes lemmatisées. Nous avons choisi de conserver tous les mots, quelle que soit leur catégorie morpho-syntaxique (POS), afin 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* indiquant l'ironie), 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). Dernière raison, non des moindres : si nous considérons que la lemmatisation est une

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

opération facilement réalisable pour beaucoup de langues [2], cela n'est pas forcément le cas de la recherche des POS.

3.3.1.1 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 3.1 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 3.3 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.

Dans les deux cas, on converge à l'itération 7 et le pic du F-score est atteint à la troisième.

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

TABLE 3.3 – *Nombre de modèles de chaînes selon les itérations – corpus Deft07-jeuxvidéo-développement*

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.

2. Par ailleurs son impact est assez limité, cette étape pourrait être ignorée sans grande influence ou remplacée par une racinisation

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] (“meilleuresChaînes”), puis avec les corpus correspondant aux itérations finales (“chaînesMax”) des expériences précédentes. Les résultats obtenus sont présentés dans le tableau 3.4.

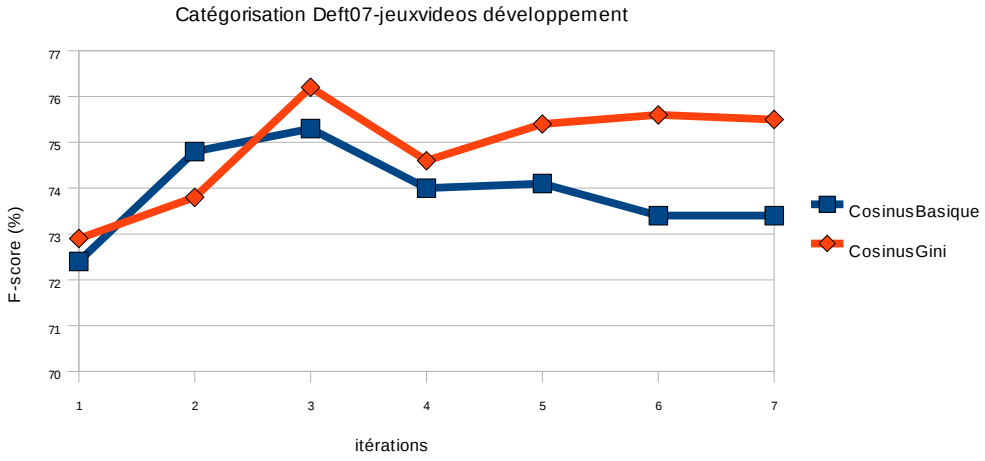

FIGURE 3.1 – *Résultats des catégorisations par cosinus sur le corpus de développement de Deft07-jeuxvideo.*

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] (chaînes qui avaient permis d’obtenir le meilleur résultat avec le cosineBasique).

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, pour toutes les valeurs testées (voir section 3.2.2), 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.

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

TABLE 3.4 – *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*

3.3.1.2 Test

Nous avons confirmé certains de nos résultats sur le corpus de test : les cosinus-Gini 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 utilisation 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* »), afin d'étudier l'impact de la quantité de données d'apprentissage disponible sur la catégorisation. Intuitivement, on peut s'attendre à obtenir de meilleurs résultats (on a plus de données, donc de meilleurs modèles de représentation des classes), mais ce n'est pas évident : cela peut aussi bruiter les modèles par l'ajout de mots moins discriminants qui feraient perdre de leur importance à d'autres, la perte de certains patrons de chaînes, une critique notée de manière incohérente (l'internaute a attribué une note qui n'est pas en rapport avec son commentaire) qui diminuerait le pouvoir d'éléments auparavant fortement liés à une opinion, ... 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 3.5 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.

Le cosinusGini (avec le corpus de développement inclus dans l'apprentissage) utilisant les chaînes s'avère donner le meilleur score de catégorisation 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 entre le développement et le test. 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 à chacun des systèmes pris individuellement (Torres-Moreno et al., 2007). Par conséquent, il devient difficile d'améliorer encore significativement les résultats.

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

TABLE 3.5 – *Résultats de la catégorisation du corpus de test de Deft07-jeuxvidéo*

3.3.1.3 Exemples de chaînes

D'une manière générale, toutes classes confondues, on trouve en grande partie des expressions typiques du domaine « *afin-de-gagner* », « *appuyer-sur-le-bouton* », « *avec-le-stick-droit* », « *le-immersion-dans-le-jeu* », « *le-ia-des-ennemi* », « *tirer-sur-tout-ce-qui-bouger* » ainsi que d'autres servant à exprimer une opinion, une comparaison « *et-le-moins-que-on-pouvoir-dire-ce-être-que* », « *force-être-de-constater-que* », « *que-il-me-avoir-être-donner-de-voir* » 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* », « *jeu-de-stratégie* », « *jeu-de-course-automobile* », 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. Ce type d'estimation peut évidemment aussi être appliqué directement à des noms de jeux, ainsi, les noms suivants ne se retrouvent pas dans la classe négative : « *crazy-taxi* », « *granturismo* », « *max-payne* » mais cela présente moins d'intérêt, dans la mesure où il suffit finalement d'aller directement regarder les critiques se rapportant à un jeu précis pour savoir si les gens en pensent du bien ou pas.

Parmi les remarques positives, on trouve des expressions se référant à l'appréciation globale : « *absolument-superbe* », « *aller-être-aux-ange* », « *autant-de-plaisir* », « *ce-qui-se-faire-de-mieux* », « *un-très-bon-jeu* », « *un-très-bon-titre* », « *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* », « *prise-en-main-être-immédiat* », « *rapide-et-fluide* », « *animation-être-fluide-et* », « *grâce-à-son-gameplay* », « *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* », « *assez-de-nouveauté* », « *de-nombreux-heure-de-jeu* », « *humour-omniprésent* », « *un-durée-de-vie-conséquent* », « *un-durée-de-vie-correct* », « *plonger-littéralement* », « *plein-de-rebondissement* », « *ambiance-prenant* », « *grand-originalité* », « *le-richesse-des-possibilité* », « *immersion-être-total* ». On note au passage des références au cinéma que l'on ne trouve pas dans la catégorie négative, ce qui pourrait laisser supposer que les jeux issus de films ont une tendance à plaire aux internautes, avec des vocables différents : « *adaptation-cinématographique* » dans la classe neutre, l'emploi de « *7ème-art* » pour les critiques dithyrambiques.

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) : « *absence-de-un-mode* », « *assez-limiter* », « *assez-décevant* », « *atteindre-pas-des-sommet* », « *atteindre-pas-le-niveau* », « *avare-en-nouveauté* », « *de-quelque-défaut* », « *dommage-de-constater-que* », « *laisser-un-peu-désirer* », « *un-peu-moins-convaincant* », « *parfois-agaçant* », « *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* », « *avoir-vraiment-rien-de-extraordinaire* » ; des points précis : « *aspect-cubique* », « *aucun-mode-multijoueur* », « *faible-durée-de-vie* », « *on-se-ennuyer-ferme* », « *réalisation-bâcler* », « *bugs-de-affichage* », « *beaucoup-trop-répétitif* », « *ennuyeux-à-mourir* » ; des accroches (que l'on peut utiliser pour extraire des corpus les mots précédents ou suivants) : « *et-comme-si-cela-ne-suffire-pas* », « *le-pire-,-ce-être-que* », « *mais-ce-ne-être-pas-tout* », « *mais-là-où-le-bât-blesser* », « *ne-avoir-aucun-intérêt* », « *pousser-le-vice-jusqu'à* », « *souffrir-de-un-manque-d'* » ; des remarques ciblant les concepteurs, développeurs ou éditeurs des jeux : « *développeurs-ne-avoir-pas* », « *développeurs-ne-se-être* » ; 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-quelque-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* ». « *plutôt-sympa* ».

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* », « *globalement-satisfaisant* », « *pas-de-gros-problème* » ; des expressions négatives : « *un-certain-lassitude* », « *autant-plus-frustrant* », « *atteindre-pas-des-sommet* », « *à-la-limite-du-supportable* », « *absence-de-renouvellement* », « *briller-par-leur-absence* », « *tout-ce-qu'-il-y-avoir-de-plus-classique* » ; 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* » « *le-niveau-de-difficulté-ne-être-pas-très-élevé* », « *ressembler-comme-deux-goutte-de-eau* » ; et énormément d'expressions de nuances : « *on-pouvoir-regretter* », « *ce-être-de-autant-plus-dommage-que* », « *avoir-gagner-à-être-un-peu-plus* », « *attendre-quelque-chose-de-plus* », « *certes-sympathique-,-mais* », « *correct-sans-plus* », « *sympathique-sans-plus* », « *tout-ne-être-pas-rose* », « *certes-,-il-y-avoir-bien* ».

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-passionnant* », « *aussi-soigner* », « *aussi-réussir* »... ; ou encore « *aucun-problème* », « *aucun-reproche* », « *aucun-souci* »... ; « *excellent-facture* », « *excellent-idée* », « *excellent-réalisation* », « *excellent-surprise* », ... les adverbes servant souvent à introduire une appréciation : « *extrêmement-complet* », « *extrêmement-détailler* », « *extrêmement-varié* », ... ; « *parfaitement-dans-le-ton* », « *parfaitement-maîtriser* », « *parfaitement-crédible* », « *parfaitement-doser* »... ; « *très-bon-facture* », « *très-bon-jeu* », « *très-simple-à-prendre-en-main* », « *très-bien-modélisés* », « *très-bien-penser* », « *très-bien-réaliser* », « *très-bon-moment* »... ; ou des patrons construits à partir du verbe « *être* » : « *être-toujours-de-le-partie* », « *être-très-réussir-et* », « *être-un-excellent-idée* », « *être-un-merveille* », « *être-un-petit-merveille* », « *être-un-régal* », « *être-un-réussite* »... 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-meilleur* », « *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 l'intensif « *très* » pour base : « *très-banal* », « *très-classique* », « *très-gênant* », « *très-laid* », « *très-sommaire* » , « *très-linéaire* », « *très-mauvais* »... des expressions en « *être* » : « *être-de-un-laideur* », « *être-de-un-pauvreté* », « *être-grossier* », « *être-ridicule* », « *être-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-inspiré* », « *peu-maniable* », « *peu-original* », « *peu-pratique* », « *peu-probant* »... « *quasi-inexistant* », « *quasi-nul* »... « *quasiment-absent* », « *quasiment-aucun* », « *quasiment-identique* »... ; des adjectifs péjoratifs : « *mauvais-goût* », « *mauvais-jeu* », « *mauvais-jouabilité* », « *mauvais-qualité* » ; des prépositions : « *sans-finesse* », « *sans-intérêt* » ; et enfin des noms ou verbes exprimant des défauts : « *problème-de-caméra* », « *problème-de-collision* », « *problème-de-jouabilité* »... ; « *manque-cruel* », « *manque-flagrant* », « *manque-total* », « *manque-d'-originalité* », « *manque-de-pêche* », « *manque-de-rythme* », « *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 : « *sans-grand-intérêt* », « *sans-grand-originalité* », « *sans-grand-prétention* », « *sans-grand-surprise* », « *sans-être-mauvais* » ; éventuellement avec des répartitions différentes dans les différentes classes, par exemple on trouve 90 expressions commençant par « *assez* » dans les critiques neutres (contre 50 en positif et 20 en négatif) : « *assez-agaçant* », « *assez-agréable* », « *assez-anecdotique* », « *assez-basique* », « *assez-bon-surprise* », « *assez-calamiteux* », « *assez-chaotique* », « *assez-drôle* », « *assez-décevant* », « *assez-inégal* », « *assez-marrant* », « *assez-plaisant* », « *assez-simpliste* », « *assez-sommaire* » ; 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 catégories (à l'exception de « *relativement-long* » dans les critiques positives) : « *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 : « *aucun-amélioration* », « *aucun-difficulté* », « *aucun-intérêt* », « *aucun-problème* », « *aucun-ralentissement* », « *aucun-sensation* », « *aucun-souci* », « *aucun-évolution* ».

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 ?). Par ailleurs, certains mots sont assez caractéristiques des corpus de critiques écrites par des internautes (néologismes, hyperlatifs ou déformations) et s'ils ne sont pas répandus ou ne possèdent pas une graphie standard empêchent d'extraire quelques chaînes. Il en va ainsi du mot « *hypeeeer* » que l'on trouve dans 2 critiques du corpus d'apprentissage (comme dans l'expression « *hypeeeer mignon* »), qui existe aussi avec un peu moins de « *e* » : « *hypeeer* » apparaît dans une critique, de même que « *hypeer* » (dans la même).

D'autres exemples pour les trois classes sont présentés en figure 3.2.

3.3.1.4 Equivalences

D'après les résultats extraits, nous pouvons remarquer qu'il y a un certain nombre de chaînes relativement proches à la fois dans la forme et par le sens : « *au-bout-de-dix-minute* » et « *au-bout-de-quelque-minute* », ou encore "*force-être-de-constater-que*" et "*force-être-de-reconnaître-que*". Nous avons donc expérimenté une première approche pour tenter d'unifier certaines de ces chaînes, aussi bien pour faciliter la présentation des extractions que pour étudier l'impact possible sur la catégorisation.

L'objectif de ce travail est de se baser sur la catégorisation des textes pour constituer des classes d'équivalence. Une classe d'équivalence contient plusieurs mots substituables pour un contexte donné (voir (Harris, 1969)). Cela s'apparente quelque peu à la notion de "synonymes", à la différence notable qu'ici des termes comme *lenteur* et *vitesse* peuvent être considérés comme équivalents (quand on parle de "lenteur d'exécution" ou de "vitesse d'exécution"), de même que *xbox* et *ps2*. Des travaux similaires ont déjà été menés, citons par exemple le logiciel Upery (Bourigault, 2002) pour l'extraction d'ontologies ou encore (Lin, 1998) pour l'extraction de mots similaires, tous deux utilisant des informations syntaxiques. Nous avons, pour notre part et pour rester dans la continuité du reste de nos travaux, choisi de nous attaquer au problème par des méthodes numériques, en partant du principe que l'idée portée par un mot est dépendante de son contexte et que la catégorie associée au texte dans lequel il se trouve peut permettre de mieux en cerner le sens. L'idée est donc d'utiliser des n-grammes (ici des pentagrammes) pour détecter des mots qui y sont substituables.

Pour constituer des classes d'équivalence, on cherche des mots substituables dans un contexte identique. Nous avons décidé de travailler au niveau des pentagrammes (compromis entre nombre de termes constituant le contexte qui doit être assez important pour définir un sens et le problème de couverture que poserait le fait de travailler sur des contextes trop grands). Nous cherchons donc des pentagrammes ayant un mot de différent avec des contextes communs. Par ailleurs, nous utilisons la répartition des textes en catégories pour essayer de garder un minimum de cohérence sémantique.

Ainsi, la démarche est la suivante :
Premièrement, on extrait du corpus d'apprentissage tous les pentagrammes (mots lemmatisés), avec pour chacun d'eux son nombre d'occurrences dans chacune des catégories où il est apparu. On ne conserve parmi ceux-ci que ceux qui ont un pouvoir discriminant suffisant, pouvoir déterminé selon leur ventilation dans les trois catégories par le critère de pureté de Gini.

On conserve donc les pentagrammes pour lesquels *pGini* est strictement supérieur à un certain seuil. On a ici choisi de fixer ce seuil à 0,5.

La seconde étape consiste à rassembler les pentagrammes selon leur catégorie dominante et leur contexte. On calcule donc pour chaque pentagramme la catégorie dans laquelle il apparaît le plus souvent ainsi que son appartenance relative à celle-ci (nombre d'occurrences dans cette catégorie divisé par son nombre d'occurrences total). On regroupe les pentagrammes ayant un seul mot différent (pour l'instant on ne s'est intéressé qu'au cas où c'est l'élément central de chaque pentagramme qui varie) et la même

catégorie majoritaire.

Il arrive parfois, que pour un même contexte et une même catégorie, il y ait un grand nombre d'éléments centraux différents. Nous avons, pour cette première exploration, choisi d'ignorer ces cas (un filtre supprime les pentagrammes quand il y a plus de 7 éléments centraux pour un même contexte), mais il pourrait être judicieux, lorsque ce problème se présente, de se rapporter à un contexte plus grand (heptagramme par exemple) pour effectuer le même travail.

Afin de consolider ces équivalences, la dernière étape consiste à aller voir si les mots centraux que l'on souhaite substituer pour cette chaîne se retrouvent également en concurrence dans un autre contexte (il existe un autre contexte pour lequel ces termes ont la même catégorie majoritaire, cette catégorie pouvant être différente de la première).

Au final, on attribue une note à chaque couple d'éléments centraux ("équivalents") pour un contexte. Cette note est calculée de la manière suivante :

$$note(m1, m2) = min(nbOcc(m1|c1), nbOcc(m2|c1))$$
$$+coef \times min(nbOcc(m1|c2), nbOcc(m2|c2))$$
$$+coef \times min(nbOcc(m1|c3), nbOcc(m2|c3)) + ...$$

avec :
$nbOcc(m1|c1)$ le nombre d'occurrences dans sa catégorie majoritaire du pentagramme ayant pour contexte $c1$ et pour élément central $m1$
$coef$ un coefficient multiplicateur pouvant prendre deux valeurs différentes selon que la catégorie majoritaire (discriminante) de ce pentagramme est la même que celle de $c1$ ou non.

Nous avons choisi de fixer $coef$ à 1 si les catégories sont identiques (c'est-à-dire que les mots se sont retrouvés substituables dans des contextes différents mais attribués à la même catégorie) et à 0,5 sinon.

Si la note est supérieure à un certain seuil, le couple de termes est considéré comme équivalent dans le contexte considéré. On applique une transitivité non-contraignante pour déterminer les classes d'équivalences.
Ainsi, pour les propositions :
P1 : m1 m2 m7 m3 m4 catégorie1
P2 : m1 m2 m8 m3 m4 catégorie1
P3 : m1 m2 m9 m3 m4 catégorie1
P4 : m1 m2 m10 m3 m4 catégorie1
P5 : m1 m2 m5 m3 m4 catégorie1

Si le couple (P1, P2) est équivalent (note supérieure au seuil) et le couple (P2, P3) est équivalent, cela suffit pour déterminer que (P1, P3) l'est aussi. Si (P4, P5) est équivalent, mais que ni P4 ni P5 ne sont équivalents avec P1, P2 ou P3, on obtient au final pour ce contexte deux classes d'équivalences : P1<->P2<->P3 et P4<->P5.

Il est à noter que cette méthode ne fait pas intervenir le pouvoir discriminant pour déterminer les équivalences, cela reste toutefois en perspective.

Voici les premiers résultats retournés par cette méthode (avec le seuil pour la note fixé à 4). Chaque ligne correspond à un pentagramme pour lequel les mots contenus entre parenthèses sont substituables. Ainsi, pour le premier exemple, les deux chaînes *"vous passer un bon moment"* et *"vous passer de bon moment"* sont considérées comme équivalentes.
vous passer (un | de) bon moment
il devenir (très | plus) difficile de
un prétexte (jeu | action) . note
. vous (pouvoir | aller) jouer rôle
moment sur (leur | votre) GBA .
ce qui (est | faciliter) d' autant
même si (on | ça) rester quand
de choisir (votre | son) camp .
de ce (titre | soft) . il
entrée sur (pc | ps2) et nous
il est (regrettable | dommage) de devoir
ne est (cependant | donc | même) pas encore
. la (jouabilité | maniabilité) rester donc
vie . (titre | jeu) en lui
vous pouvoir (toujours | même) vous amuser

Certains des résultats sont visuellement intéressants (regroupement de *il est regrettable de devoir* et *il est dommage de devoir*), d'autres le semblent moins (*ce qui (est | faciliter) d' autant*). Nous avons remplacé, dans les corpus d'apprentissage et de test, les pentagrammes équivalents par un représentant de la classe d'équivalence (par exemple, on a remplacé toutes les occurrences de *il est dommage de devoir* par *il est regrettable de devoir*). Puis, nous avons effectué une catégorisation afin d'étudier l'impact de ces regroupements. Les résultats se sont améliorés d'environ 0,2%, ce que nous ne considérons pas comme significatif [3].

Ces travaux n'ont pas été continués plus en avant et n'ont pas été appliqués sur les autres corpus présentés dans ce chapitre. Toutefois, la méthode a déjà donné quelques résultats intéressants.

3.3.2 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-

3. Bien que tous les paramètres utilisés ici aient été choisis arbitrairement (hormis le seuil pour le critère *note* qui a été testé à différentes valeurs : 4 étant celle qui retourne des équivalences qui semblent visuellement plus intéressantes et 1,5 celle qui nous a permis d'obtenir le plus fort gain en catégorisation) et non en cherchant à les optimiser, il s'agissait juste d'évaluer la pertinence de continuer ou non ces travaux

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, normalement fonction de la qualité des prestations lors de ce récent contact. Les opérateurs rappellent ainsi des clients et leur posent les questions suivantes :

- Q1 : Sur la base de cette prestation, dans quelle mesure recommanderiez-vous EDF à vos amis et vos collègues sur une échelle de 1 à 10 ?
- Q2 : Quels sont les facteurs qui motivent cette réponse ?
- Q3 : Que pourrait faire EDF Entreprise selon vous pour que vous mettiez 9 ou 10 ?
- Q4 : Accepteriez vous que votre interlocuteur commercial vous rappelle afin d'approfondir les points évoqués ?

Les réponses sont retranscrites par l'opérateur au cours de l'entretien. Les objectifs de cette enquête sont doubles. Premièrement, la durée courte du questionnaire et l'ouverture des questions permettent d'obtenir des réponses aux questions ouvertes riches en enseignement : on peut à la fois en extraire un classement thématique général sur les motifs de satisfaction ou d'insatisfaction des clients ou de leurs attentes, ainsi que des remarques détaillées, par exemple en effectuant un travail de fouille de données permettant d'extraire des passages des réponses aux questions que l'on retrouverait assez fréquemment. Deuxièmement, 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 » pour Net Promoting Score) 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, à la fois au niveau national et au niveau régional. 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 proposons ici d'effectuer une catégorisation d'opinion, en utilisant les réponses aux questions (Q2) portant sur les raisons justifiant la note attribuée, pour essayer de retrouver la catégorie du client (Promoteur, Détracteur ou Neutre). D'autre part, la méthode présentée en section 2.2 nous permet d'extraire des chaînes caractéristiques de chacune de ces trois catégories, faisant ainsi ressortir des remarques fréquemment rencontrées parmi les clients Détracteurs ou Promoteurs. Notons que nous avons choisi de rechercher non pas la note exacte, ce qui présente peu d'intérêt, mais la catégorie, qui à la fois est l'objectif visé par l'entreprise (pour calculer le NPS) et permet d'éviter en partie les problèmes de subjectivité de notation présentés en fin de cette section. 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 souvent dans le domaine (mais représente un cas d'utilisation réel) : 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 3.6. Nous présentons dans le tableau 3.7 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.

	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

TABLE 3.6 – *Répartition du nombre d'entretiens pour le corpus 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 est 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 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 pouvant être, à terme, de travailler directement sur des sorties de système de reconnaissance automatique de la parole), 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 (pas de filtre sur les mots (anti-dictionnaire) ou sur les catégories morpho-syntaxiques (POS)).

3.3.2.1 Développement

Nous avons testé les différents paramètres suivants pour les catégorisations par cosine (Basique et Gini) :

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

TABLE 3.7 – *Statistiques sur le corpus NPS07-09*

- filtres de sélection des collocations : nous avons testé les couples de paramètres suivants [valeur de LRV ; nombre d'occurrences] d'après l'observation des propositions retournées par le système (les filtres sont globalement moins sévères que dans le cas du corpus Deft07-jeuxvidéo étant donné que le corpus est moins fourni) : [25 ;2], [25 ;3], [25 ;4], [25 ;5], [25 ;6], [30 ;3], [30 ;4], [30 ;5], [30 ;6], [40 ;3], [40 ;4], [40 ;5], [40 ;6], [50 ;4], [50 ;6], [50 ;7], [60 ;5], [60 ;7], [60 ;10], [75 ;5], [75 ;10] ;
- repli : les mots composants les chaînes ont été ajoutés aux comptes des mots isolés avec les valeurs : 0 ; 0,2 ; 0,5 ; 1.

Dans le cas du cosinusBasique, les paramètres de filtre de collocations [50 ;4] sont les plus optimaux que nous ayons trouvé. 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 3.3. 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 moyen 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 peut déduire du fait que le gain de F-score moyen soit inférieur au gain en nombre de documents, 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 confusion issues de ces catégorisations (présentées en tableaux 3.8 et 3.9). 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 caractéristiques de 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 %.

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

TABLE 3.8 – *Matrice de confusion pour l'itération 1 de la catégorisation par cosinusGini pour le filtre [75;5], pour les trois catégories : détracteur (DET), neutre (NEU) et promoteur (PRO)*

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

TABLE 3.9 – *Matrice de confusion pour l'itération 8 de la catégorisation par cosinusGini pour le filtre [75;5], pour les trois catégories : détracteur (DET), neutre (NEU) et promoteur (PRO)*

Les résultats de catégorisation par SVM pour trois classes sont présentés dans le tableau 3.10. 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.

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

TABLE 3.10 – *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*

3.3.2.2 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 3.4. On

remarque que la correspondance filtre optimal/classifieur n'est pas forcément la même entre le développement et le test (par exemple, le filtre [50 ;4] était sur le développement le meilleur pour le cosineBasique, alors que sur le test, ce filtre est plus efficace pour le cosineGini que pour le cosineBasique). 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. Ces résultats correspondent aux expériences réalisées en intégrant le corpus de développement dans l'apprentissage ; les expériences n'utilisant que les 10029 documents de l'apprentissage ont donné des résultats légèrement moins bons. (F-score inférieur de 0,1 à 1 % avec utilisation des chaînes – en revanche, sans utilisation de chaînes, les résultats sont légèrement meilleurs sans intégrer le corpus de développement dans l'apprentissage (59,4 au lieu de 59,1 % dans le cas du cosineBasique et 60,0 au lieu de 59,3 % dans le cas du cosineGini) – ces résultats restent toutefois 2 % moins bons que ce que l'on obtient au final par l'utilisation des chaînes).

Pour ce qui est des SVM, nous avons réalisé les tests présentés dans le tableau 3.11, 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 pourcents 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).

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

TABLE 3.11 – *F-scores obtenus sur la catégorisation du corpus de test de NPS07-09 par les différentes méthodes présentées précédemment*

3.3.2.3 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, avec les paramètres de filtre [50 ;4]. Malgré le filtre plus laxiste que celui qui a mené aux exemples de la section 3.3.1.3, 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 dizaine de chaînes exprimant le problème de n'avoir pas d'interlocuteur privilégié : « avoir-pas-d-interlocuteur-direct », « on-n-avoir-jamais-le-même-interlocuteur », « on-n-avoir-jamais-le-même-personne », « on-ne-tomber-jamais-sur-le-même-personne », « j-avoir-avoir-plusieurs-personne », ... ; 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* », « tout-s-être-très-bien-passer »... Un certain nombre de variantes sont purement linguistiques. « *résolution-du-problème* », « *résoudre-le-problème* », « *régler-le-problème* » ; « avoir-répondre-mon-attente », « avoir-répondre-mon-demande », « avoir-répondre-mon-question » ; 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 « être + [ADJECTIF | PASSE COMPOSE] » : « être-agréable », « être-aimable », « être-charmant », « être-clair », « être-compétent », « être-correct », « être-efficace », « être-gentil », « être-parfait », « être-régler », « être-satisfaire », « être-sympa ». Ou encore, pour une règle basée sur « 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* ». Pour une règle du type « bien + VERBE » : « bien-accueillir », « bien-conseiller », « bien-dérouler »,

« bien-expliquer », « bien-passer », « bien-renseigner ». 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-j-attendre-toujours » ; et à l'inverse : « *avoir-répondre-rapidement* », « *avoir-être-faire-rapidement* », « *cela-avoir-être-très-rapide* », « *dans-le-jour-qui-avoir-suivre* », « dans-le-journée », « *dans-le-semaine* », « *en-temps-et-en-heure* », « *quelque-jour-après* », « *rapide-et-efficace* », « *le-rapidité-de-le-réponse* », « traiter-rapidement ». Celles exprimant une durée peuvent tout de même être propres à certaines catégories : des marqueurs longues durées ne sont jamais un signe positif : « 1-an » ou « 1-mois » ne se trouvant que dans les classes Détracteur ou Neutre, à l'encontre de « 1-semaine » qui ne se retrouve que dans les discours des clients Promoteurs. En durée intermédiaire, on peut trouver « 15-jour », ce délai pouvant être positif ou négatif selon ce qu'on attend.

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 (c'est le cas de « *coordination-entre-le-service* » que l'on ne retrouve que dans chez les Détracteurs).

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

3.3.3 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 catégorisation 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. La catégorisation d'opinions sur des critiques de films est réputée pour être une tâche difficile. Quelques explications de ce phénomène sont données dans (Turney, 2002), où l'auteur applique un même classifieur, avec des résultats différents, sur quatre corpus d'opinions portant sur des thèmes différents : Automobiles (84%), Banques (80%), Films (66%) et Destinations de voyages (71%). Le problème vient notamment du fait que la description du film peut contenir le même vocabulaire que celui employé pour critiquer le film (un "personnage mauvais" n'implique pas forcément que le film est **mauvais**, alors qu'un "mauvais employé de banque" incite à conclure que la banque l'est). Le problème existe aussi pour les critiques de livres par exemple.

Les 600 premières critiques de chacune des deux catégories 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 poussé à n'utiliser que 200 critiques pour le développement, ce que nous avons considéré comme suffisant pour adapter seulement deux paramètres des SVM. 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 les 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.

3.3.3.1 Développement

Nous avons testé les filtres de sélection de collocations suivants : [25 ;6], [30 ;2], [30 ;3], [30 ;4], [30 ;5], [30 ;6], [40 ;3], [40 ;4], [40 ;5], [40 ;6], [50 ;4], [50 ;5], [50 ;6], [60 ;4], [60 ;5], [60 ;6], [70 ;4], [70 ;5], [75 ;5], [75 ;7], [80 ;4], [80 ;5], [80 ;6], [90 ;4], [90 ;5], [90 ;6], [90 ;7], [100 ;4], [100 ;9], [100 ;10], [100 ;11], [150 ;9], [150 ;10], [150 ;11].

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.

3.3.3.2 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] où le pourcentage de critiques correctement étiquetées passe ainsi de 85 à 85,8 (515 critiques), ce qui est légèrement endessous du cosineBasique qui obtient 86,5 % (519 critiques) sans utilisation de chaînes. 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 subjectivité 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 3.2. 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 pensons qu'il pourrait être intéressant de tenter 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. Notamment car le corpus étant assez petit, le fait d'utiliser 1800 critiques pour l'apprentissage (comme c'est le cas pour les articles présentés précédemment qui utilisent une validation croisée en 10 sous-parties) devrait permettre d'améliorer les modèles par rapport à notre apprentissage basé sur 1400 critiques (mais dans ce cas, un problème se poserait pour l'interprétation des résultats où les écarts seraient encore moins significatifs).

3.3.3.3 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* », « *get-the-feeling-that* », « *because-of-the-fact-that* », « *his-performance-be* », des expressions du domaine annonçant une opinion « *digital-effects* », « *first-twenty-minutes* », « *the-dialogue-be* », ou celles présentant des genres « *black-comedy* », « *romantic-comedy* », « *horror-film* », « *horror-movies* », qui peuvent être quantifiées et comparées d'une catégorie à l'autre. On trouve ensuite des expressions d'opinion négative sur l'ensemble du film « *,-there-is-nothing* », « *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* », « *keep-the-audience* », « *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* » , « *the-phantom-menace* » 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* », ce qui peut nous permettre de savoir qui on va embaucher sur notre film (si l'on désire plaire aux internautes).

3.4 Discussion

3.4.1 SVM à noyaux de mots

L'intérêt des chaînes que nous extrayons 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 :

- 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 $maxSubsequenceLength = 4$, $subsequenceLength = 2$, 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.
- les chaînes prises en compte par ces méthodes SVM 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 $TF \times IDF \times pGini$).

3.4.2 La catégorie Neutre

La catégorie regroupant les documents dits "Neutres" est un sujet délicat en détection d'opinion. Celle-ci est fréquemment écartée des corpus, comme c'est le cas pour le *Movies Polarity Dataset v2.0*, pour lequel n'ont été gardées que des critiques positives ou négatives. Dans la pratique, cette catégorie peut, selon les tâches, avoir deux définitions : soit elle regroupe des documents réellement **neutres** c'est-à-dire n'exprimant aucune opinion (un entretien dans lequel un client n'aurait aucune critique à formuler, mais ne ferait pas d'éloge non plus), soit elle regroupe les documents exprimant à la fois des opinions positives et négatives (dans une critique de film, il s'agit d'une personne qui a attribué une note moyenne, parce qu'il aime bien le scénario mais pas le jeu des acteurs). Dans la première définition, le problème peut être abordé par une catégorisation en deux passes : la première consiste en une détection de la subjectivité (est-ce qu'on trouve l'expression d'une opinion dans ce document), si le texte n'est pas subjectif, il est neutre, dans le cas contraire, on applique une deuxième passe le catégorisant comme positif ou négatif. Pour la seconde, de nombreuses méthodes sont envisageables : on peut se baser sur les scores de similarité avec les classes positive et négative, si la différence entre les deux est maigre, il peut s'agir d'une critique intermédiaire ; ou on peut extraire des éléments détaillés (les chaînes) et si on en trouve des connues comme étant positives et d'autres comme négatives, on peut décider que la critique est neutre ; on peut encore subdiviser la catégorie Neutre de l'apprentissage en plusieurs sous-catégories par une méthode de clustering dont on attend qu'elle regroupe les documents neutres légèrement positifs d'un côté, les neutres plutôt négatifs de l'autre - nous avons tenté quelques expériences liées à cette dernière approche, en subdivisant la catégorie Neutre du corpus *Deft07-jeuxvidéos* par un algorithme de k-means, mais nous n'avons pas obtenu les résultats escomptés (les résultats se sont dégradés).

3.5 Conclusion

Dans ce chapitre, nous avons présenté l'application de la méthode décrite au chapitre précédent, dans un contexte de catégorisation d'opinion, permettant d'extraire des chaînes de mots relatives à des opinions diverses exprimées sur des produits et services. Ces chaînes offrent la possibilité de visualiser des expressions exprimant une opinion, soit directement (« *avoir-vraiment-rien-de-transcendant* »), soit sous forme d'accroche (« *force-être-de-constater-que* »). Ceci peut permettre, soit globalement d'extraire les remarques fréquemment exprimées dans l'ensemble des documents dont nous disposons, soit dans un document de visualiser les passages exprimant des opinions (par exemple en y surlignant les chaînes connues).

Au-delà des améliorations que nous proposons dans le chapitre précédent puis en fin de cette section, 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), différentes tailles de corpus ;
- la robustesse de l'approche vis-à-vis des classes d'opinion considérées (*promoteur/neutre/détracteur* pour les enquêtes de satisfaction, positif/neutre/négatif (ou positif/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 catégories permet à la fois d'avoir un aperçu des points les plus abordés, des remarques formulées les plus fréquemment, de repérer 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, menant ainsi à la problématique du croisement entre les thématiques et les opinions. Elles peuvent ensuite être utilisées pour améliorer les performances des systèmes de catégorisation automatique de textes en opinion car elles peuvent présenter l'avantage d'être plus discriminantes que des mots pris isolément (ce qui n'est pas le cas de toutes, étant donné qu'on ne les a pas sélectionnées selon ce critère).

Dans un contexte de détection d'opinions, 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 catégorisation a été montré sur les trois corpus présentés dans ce chapitre. En particulier, le couple cosinusGini + 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, sauf 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 relativement 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 fau-

drait 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, nous avons ici testé l'impact de l'accroissement de la quantité de données utilisées pour l'apprentissage (en incluant ou non le corpus de développement dans la construction des modèles utilisés pour traiter le corpus de test). Ceci est notamment intéressant dans un cas d'entreprise pour savoir quand arrêter de collecter des données (quelle quantité il est nécessaire de récupérer, puis d'annoter, ce qui est un processus coûteux), sur quelle périodicité, etc. Pousser plus loin cette étude fait partie de nos travaux futurs.

Positif	toujours-aussi-efficace	clairement-orienter	pire-encore
aller-encore-plus-loin	tout-ce-qui-avoir-faire-le-succès	clairement-destiner	pièrte-qualité
amateur-de-sensation-fort	tout-simplement-impressionnant	se-avérer-relativement	plus-grave
apporter-un-peu-de-fraîcheur	tout-simplement-somptueux	se-en-sortir-très-bien	plus-mauvais
assez-bien-rendre	tout-bonnement-impressionnant	grief-que-on-pouvoir-faire	plus-moche
autant-plus-intéressant	tout-le-monde-d'accord	autre-défaut	plus-que-limité
avoir-avoir-le-bon-idée	tout-à-fait-remarquable	autre-problème	point-faible
avoir-déjà-faire-son-preuve	tout-à-fait-réussir	bien-réaliser	poser-problème
avoir-mériter-de-être	tout-à-fait-satisfaisant	autant-le-dire-tout-de-suite	ressentir-aucun
avoir-être-revoir-à-la-hausse	tout-à-fait-soigner	le-seul-problème	rien-de-extraordinaire
beaucoup-mieux	un-très-bon-surprise	amélioration-notable	se-contenter-de
ce-être-tant-mieux	un-vrai-bonheur	agréablement-surprendre-par	simplifier-à-le-extrême
comme-le-référence	un-vrai-régal	arriver-pas-à-la-cheville	toujours-de-la-même-manière
comme-on-le-aimer	un-véritable-régal	apporter-pas-grand---chose	toujours-le-même-chose
complètement-inédit	un-véritable-réussite	apporter-un-peu-de-fraîcheur	tout-le-temps-le-même-chose
convaincant-et-surtout	pas-grand-chose-envier	apporter-un-peu-de-variété	un-manque-flagrant
dans-le-lignée-de-son-prédécesseur	valoir-le-détour	une-fois-ne-être-pas-coutume	cela-ne-suffire-pas
dans-le-plus-pur-style	être-de-bon-facture	pouvoir-se-targuer-de	mériter-pas
de-anthologie	être-de-bon-qualité	pouvoir-se-vanter-de	mérite-de-ne-pas
de-excellent-facture	ce-qui-ne-être-pas-pour-nous-déplaire	ressembler-à-se-y-méprendre	même-pas-le-peine
de-grand-classe	du-grand-art		en-avoir-vite
de-grand-moment	le-prise-en-main-être-immédiat	**Négatif**	le-plus-clair-de-son-temps
de-grand-qualité	se-intégrer-parfaitement	,-ce-ne-être-pas-vraiment	artificiellement-gonfler
de-son-qualité	se-montrer-à-la-hauteur	,-le-développeurs-ne	rien-de-bien-transcendant
de-surprise-en-surprise	se-prendre-au-jeu	aider-pas-vraiment	
de-très-bon-facture	tomber-sous-le-charme	ne-se-embarrasser-pas	
de-très-bon-moment	toujours-autant-de-plaisir	aucun-plaisir-de-jeu	
de-très-bon-qualité	toujours-aussi-beau	avoir-beaucoup-de-mal	
de-très-joli	toujours-aussi-bon	avoir-bien-du-mal	
de-un-richesse-incroyable		difficilement-supportable	
devoir-faire-le-bonheur-des	**Neutre**	couper-le-son	
dimension-spectaculaire	malheureusement-,-on	cruellement-de-intérêt	
développeurs-avoir-savoir	même-se-il-être-possible-de	du-plus-mauvais	
en-avoir-pour-votre-argent	quoi-que-il-en-être-,	désagréable-impression	
en-mettre-plein-le-oeil	rien-de-révolutionnaire	en-tout-et-pour-tout	
en-prendre-plein-le-mirettes	rien-de-très-original	faible-nombre-de	
encore-plus-impressionnant	seul-ombre-au-tableau	faible-qualité	
encore-plus-spectaculaire	,-ce-qui-gâcher	faire-le-impasse-sur	
et-riche-en-sensation	,-ce-qui-être-tout-de-même	grave-problème	
facile-prendre-en-main	,-ce-être-indéniable	gros-déception	
faire-durer-le-plaisir	,-force-être-de-constater-que	gros-défaut	
faire-pas-le-affront-de	,-force-être-de-reconnaître-que	gros-point	
fort-appréciable	autant-plus-dommage-que	gros-problème	
foule-de-possibilité	autant-plus-regrettable-que	grossier-et	
gage-de-qualité	avoir-au-moins-le-mérite-de	il-être-impossible-de	
gameplay-solide	avouer-rapidement-son-limite	jeter-le-éponge	
gameplay-novateur	en-rebuter-plus-d'un	jeu-ne-avoir-rien	
goûter-aux-joie	essayer-avant-d'-acheter	lasser-rapidement	
haut-de-gamme	et-on-avoir-parfois-du-mal	lasser-très	
le-plaisir-de-jeu-être	limiter-en-raison-du	lassitude-gagner	
le-plus-grand-bonheur-des	mais-force-être-de-constater-que-le	laisser-désirer	
lettre-de-noblesse	montrer-vite-son-limite	le-résultat-ne-être	
mettre-plein-le-oeil	ne-plaire-pas-tout-le-monde	n'importe-comment	
mettre-plein-le-vue	ne-être-pas-suffisamment	ne-aider-en-rien	
mais-on-avoir-aimer	ne-être-pas-très-convaincant	ne-aider-pas-vraiment	
mériter-amplement-son-place	parvenir-tout-de-même-pas	ne-avoir-rien-de-très	
ne-avoir-pas-être-oublier	relever-le-intérêt	ne-avoir-vraiment-rien-de	
ne-être-pas-décevoir	relever-le-niveau	ne-faire-pas-mieux	
ne-être-pas-en-rester	rester-sur-son-faim	ne-parvenir-pas-convaincre	
nous-surprendre	revers-de-le-médaille	ne-pouvoir-décemment	
nous-transporter	risquer-de-être-décevoir	ne-pouvoir-même-pas	
pour-le-plus-grand-bonheur-des	se-avérer-plutôt	ne-pouvoir-plus-basique	
pour-notre-plus-grand-bonheur	se-suivre-et-se-ressembler	ne-rien-arranger	
pouvoir-se-targuer-de	somme-toute-assez	ne-y-avoir-pas-grand-chose	
prendre-beaucoup-de-plaisir	somme-toute-classique	ne-y-avoir-rien	
remettre-au-goût-du-jour	valoir-le-coup-de-oeil	ne-être-pas-à-la-hauteur	
revenir-sur-le-devant-de-le-scène	être-agréablement-surprendre	nettement-en-retrait	
richesse-du-gameplay	être-simplement-dommage-que	nettement-moins	
richesse-incroyable	quelque-idée-intéressant	on-passer-son-temps	
réellement-passionnant	quelque-incohérence	on-regretter	
référence-du-genre	quelque-petit-problème	on-être-en-droit-de-attendre	
référence-en-matière-de	de-manière-dommageable	pas-à-la-hauteur	
	ce-que-on-être-en-droit-de-attendre	passer-à-côté-de	

FIGURE 3.2 – *Exemples de chaînes caractéristiques de chacune des catégories d'opinion (positif, neutre ou négatif) du corpus Deft07-jeuxvideo.*

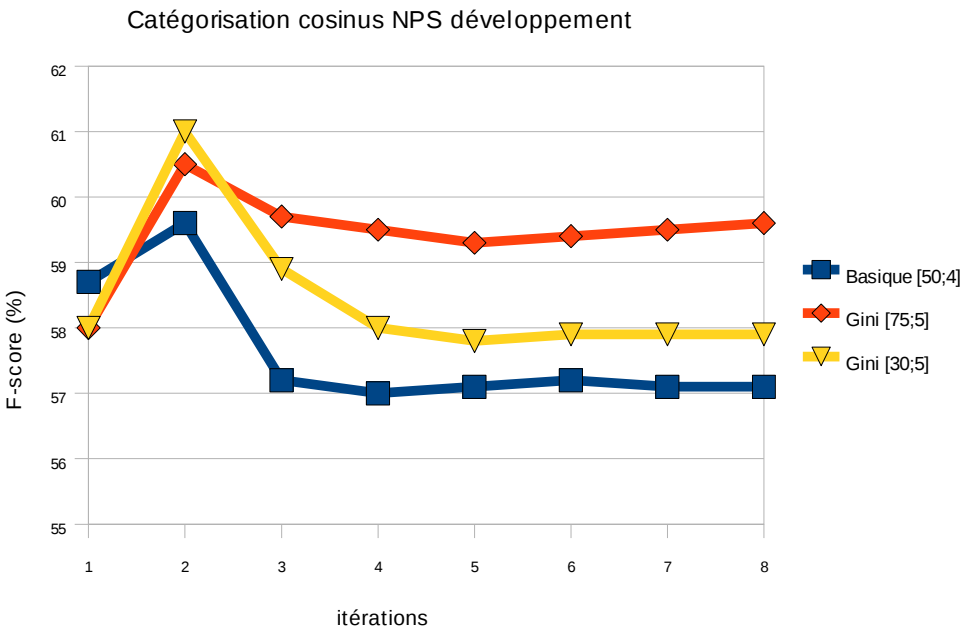

FIGURE 3.3 – *F-scores des catégorisations par cosinus sur le corpus de développement de NPS07-09*

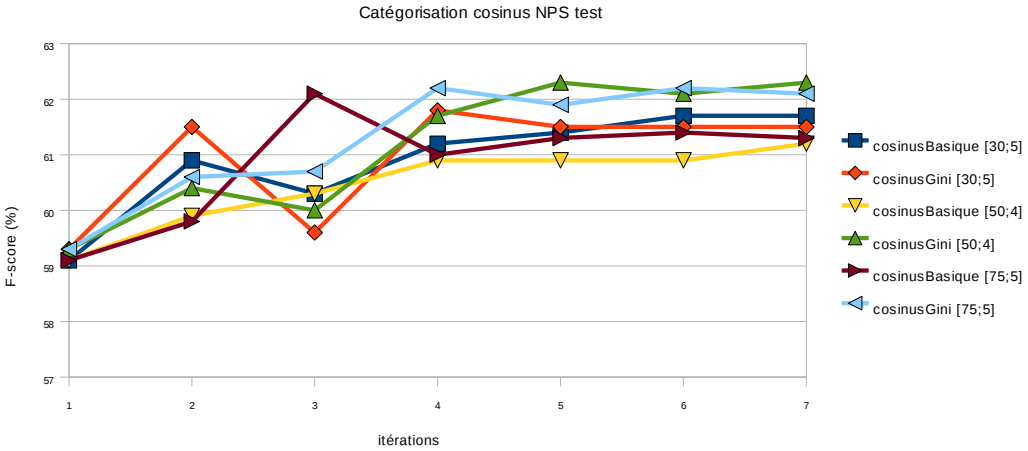

FIGURE 3.4 – *F-scores des catégorisations par cosinus sur le corpus de test de NPS07-09*

Chapitre 4

Des chaînes reliant des sujets et des opinions exprimées à leur propos

Sommaire

Résumé

Ce chapitre présente une nouvelle approche permettant d'extraire des portions de phrases dans lesquelles une opinion est exprimée sur un sujet précis. Il s'agit dans un premier temps de définir la liste des sujets qu'il est intéressant de surveiller, soit manuellement, soit par une méthode automatique. Puis extraire du corpus d'apprentissage une liste de ce que peuvent être des n-grammes porteurs d'opinion. Enfin, on peut rechercher dans le corpus de test tous les segments de textes s'étendant d'un sujet de la liste à un n-gramme d'opinion connu trouvé dans un voisinage proche.

4.1 Introduction

Nous nous intéressons dans ce chapitre au ciblage d'opinions précises exprimées dans un document, c'est-à-dire plutôt que de trouver l'opinion globale d'une critique, aller rechercher les éléments suivants :

- qui parle de quoi ?
- de quel point précis ?
- quelle est l'opinion exprimée ?
- à quel moment ?

C'est ce que (Liu, 2010) représente sous la forme d'un quintuple $(o_j, f_{jk}, oo_{ijkl}, h_i, t_l)$ où :
o_j est un objet
f_{jk} est une des caractéristiques de l'objet o_j
oo_{ijkl} est la polarité de l'opinion (positive/négative ou avec une échelle de valeurs plus détaillée) exprimée sur la caractéristique f_{jk}
h_i est la personne (physique ou morale) exprimant l'opinion
t_l est le moment où l'opinion est exprimée

Les deux derniers points ne concernent pas les travaux effectués dans cette thèse. Il s'agirait par exemple de répondre à une question du type : "Que pensent les personnalités politiques du projet de loi de taxation sur les très hauts revenus proposé par le candidat ?" en explorant des articles de journaux afin de retrouver des phrases telles que : "Au lendemain de cette annonce, le Président de la République a déclaré : «Tout ça donne une impression d'improvisation et d'amateurisme qui est assez consternante»". Dont nous pourrions extraire ces éléments :
h_i ="le Président de la République"
t_l ="au lendemain de cette déclaration", qui peut nous permettre de déduire la date exacte
Dans ce premier exemple, cette tâche de recherche de la personne peut sembler aisée : le Président de la République est le sujet d'un verbe d'action, qui plus est précédant une citation. Mais il existe de nombreux cas plus complexes : "La proposition du candidat a mis dans l'embarras le responsable du pôle budget et finance de l'équipe de campagne, qui l'a découverte au moment de l'interview.", expliquant que ce sujet de recherche soit encore exploré à l'heure actuelle (Wiegand et Klakow, 2012).

Cependant, compte tenu des données sur lesquelles nous travaillons, ces deux points sont pour nous explicites : qu'il s'agisse de critiques écrites par des internautes ou d'enquêtes réalisées par une entreprise, nous disposons, si toutefois cela nous intéressait, de la date précise où l'opinion a été exprimée (pas forcément intéressant pour un produit culturel, tel qu'un film où toutes les opinions exprimées depuis sa sortie sont intéressantes et n'ont pas de raison d'évoluer - si ce n'est pour des commentaires tels que "la bande-son a bien vieilli" - puisque le produit reste le même ; en revanche, pour une entreprise, la temporalité peut avoir de l'importance afin de juger de l'évolution des avis selon les périodes), ainsi que d'un moyen (plus ou moins anonyme) d'identifier l'auteur de la critique.

Restent donc les trois premiers tuples. L'objet o_j étant, pour sa part, généralement connu (un film), nous nous intéresserons ici au problème de trouver les opinions exprimées sur certaines caractéristiques d'un produit ou d'un service. Ces deux problèmes ont été abondamment traités dans la littérature. Soit séparément : trouver les caractéristiques sur lesquelles une opinion peut être exprimée ou les expressions d'opinions dans un corpus (Wilson et al., 2005), (Breck et al., 2007). Soit de manière conjointe.

Nous ne nous attarderons pas ici sur le premier point, l'extraction des caractéristiques. Nous noterons toutefois que le vocable employé pour décrire ce problème est assez variable selon les auteurs ou la tâche précise. On trouvera alors des dénominations telles que aspect, caractéristique (feature), cible (target), attribut, composant, ...

Nous parlerons pour notre part de sujet sur lequel une opinion est exprimée (un sujet déclencheur de l'expression d'une opinion), par rapport à un domaine (la critique de film, l'avis des utilisateurs sur l'entreprise). Nos sujets ne correspondant pas forcément à des composants du domaine au sens cela est généralement entendu. Ainsi, pour l'entreprise, les sujets recouvrent des concepts tels que le nom de l'entreprise, la qualité du service, le temps de réponse, l'échange (en termes de communication), le traitement (d'une demande), ... Il s'agit donc dans notre cas, de sujets déclencheurs d'une opinion et non pas nécessairement d'une caractéristique d'un produit comme cela est généralement abordé dans les travaux concurrents (nombre d'entre eux portent sur les critiques de produits électroniques, les caractéristiques sur lesquelles on recherche des opinions sont alors, par exemple pour des appareils photos : le prix, la qualité de l'image, l'objectif, la prise en main, la taille, ...). Il arrive fréquemment que dans des phrases, pourtant porteuses d'une opinion, on observe l'absence de la mention du produit ou d'une de ses caractéristiques ("Cet achat était vraiment décevant."). Ce dernier point justifiant notre approche ne se basant pas sur les caractéristiques du produit, mais sur un ensemble de sujets, qui recouvre des concepts plus vastes.

La recherche des opinions exprimées sur ces caractéristiques fait aussi l'objet de nombreuses recherches, l'expression d'une opinion se résumant rarement à "Ce produit est mauvais.", mais pouvant prendre des formes très variées, posant de nombreux problèmes qu'il est difficile d'interpréter pour un ordinateur. Cela va de la nuance ("Ce film était presque bien en fait"), à la comparaison ("Ce film était moins bon que le précédent"), en passant par l'ironie, plus ou moins facile à percevoir ("Je vous recommande chaudement d'aller voir ce film, pour peu que vous fussiez un adolescent attardé."), ("Bref, comme vous l'avez compris, le meilleur film du monde", suivant un paragraphe entier consacré à en citer tous les défauts), etc.

Nous essayons, dans nos travaux, de contourner une partie de ces problèmes en nous basant non pas, comme cela est couramment le cas, sur des listes prédéfinies d'adjectifs ("bien", "mauvais"), d'adverbes ("admirablement", "maladroitement") ou de verbes ("je recommande", "je vous le déconseille") porteurs d'opinion, mais en allant automatiquement extraire du corpus d'apprentissage des n-grammes porteurs d'opinion, sans souci de leur composition, dans le but de récupérer par exemple des expressions telles que "Cela aurait pu être bien" et de s'affranchir du problème des mots qui n'ont pas la même polarité selon le domaine dans lequel ils sont employés.

On pourra à ce propos, citer l'exemple favori de Bing Liu : le verbe "to suck" qui est généralement employé pour une expression négative ("This camera sucks"), mais peut au contraire servir à exprimer la qualité de certains produits ("This vacuum cleaner really sucks").

Parmi les travaux ayant déjà abordé ce problème de ciblage d'opinion sur un point précis, on en trouve un certain nombre, notamment dans le cas de critiques de produits électroniques, se basant sur des indices linguistiques ou des connaissances externes : WordNet, dictionnaires, règles d'association, listes de termes connus comme étant porteurs d'opinion, règles pour la gestion des négations, synonymes et antonymes, POS, ontologies, grammaires de dépendances et analyses syntaxiques (Somprasertsri et Lalitrojwong, 2010),(Wu et al., 2009), (Popescu et Etzioni, 2005), utilisant parfois en plus des corpus d'apprentissage dans lesquels des mots ou phrases ont été manuellement étiquetés selon l'opinion qu'ils représentent (Jin et al., 2009), ou bien dans le domaine des films (Zhuang et al., 2006).

Nous avons ici pris le parti d'utiliser le minimum de connaissances afin de proposer une méthode la plus générique possible, qui sera ainsi transposable dans différentes langues, différents domaines (puisque le même mot peut exprimer des opinions contraires selon le domaine considéré) et tentant d'éviter au maximum les embûches présentées précédemment. Nous utiliserons donc uniquement l'étiquette globale attribuée aux documents du corpus considéré, un lemmatiseur et un corpus généraliste. Pour cela, nous partirons d'un ensemble de sujets sur lesquels une opinion peut-être exprimée dans le domaine qui nous intéresse. Ces sujets sont définis manuellement en fonction des besoins de l'entreprise (les sujets pour lesquels l'entreprise veut étudier les opinions des client), mais nous proposons aussi une approche automatique permettant de faciliter cette étape. Nous présentons ensuite un moyen d'obtenir automatiquement des n-grammes porteurs d'opinions - en nous basant sur leur répartition dans les différentes catégories d'opinions des textes dans lesquels on les retrouve - ceux-ci devant permettre de prendre implicitement en compte des nuances, négations, etc. Nous disposons bien entendu de la polarité associée à ces n-grammes (positive ou négative). Enfin, nous extrayons des documents, des portions de textes (dits "segments") recouvrant un sujet et l'opinion exprimée à son propos. Le fait d'extraire des segments entiers permet d'offrir un aperçu assez fiable de ce qui est réellement dit par la personne laissant la critique : si une nuance s'est glissée entre le sujet et l'opinion et que cette nuance était assez peu fréquente dans le corpus d'apprentissage, il se peut que le système ne la reconnaisse pas, alors que l'humain qui lira le résumé produit par le système (l'ensemble des segments retournés) pourra la visualiser. Ceci n'empêchant pas par ailleurs de retourner uniquement des couples (sujet ; opinion) dans un but de synthèse automatique.

4.2 Méthode

Le but de ces travaux est d'extraire des segments de texte dans lesquels des opinions sont exprimées à propos de certains sujets. La méthode est ainsi divisée en trois étapes. En premier lieu, il s'agit de définir les sujets sur lesquels une opinion peut être

exprimée (par exemple *soundtrack* dans le cas de critiques de films). Deuxièmement, nous recherchons des n-grammes employés dans le corpus d'apprentissage pour exprimer une opinion (*very effective*). Finalement, sur le corpus de test, nous extrayons des segments dans lesquels une opinion est donnée à propos d'un des sujets connus (*soundtrack is also very effective*).

4.2.1 Définir les sujets

Nous avons employé séparément les approches suivantes pour définir deux listes de sujets sur lesquels des opinions peuvent être exprimées :

- **extraction manuelle** : nous avons, dans un premier temps, utilisé notre connaissance des domaines considérés pour établir une première liste de termes que nous pensions pertinents (par exemple, dans le domaine du cinéma, on peut facilement imaginer que les critiques porteront sur les acteurs, le réalisateur, le scénario, ...) ; nous avons ensuite étendu cette première liste en lisant un échantillon du corpus d'apprentissage (quelques centaines de documents) et en sélectionnant des termes sur lesquels une opinion est souvent exprimée.
- **extraction automatique** : nous avons utilisé le calcul du Zscore (formule 4.1), introduit par (Savoy, 2009). Le Zscore indique, pour chaque mot d'un corpus (ci-après "corpus à étudier"), sa représentativité par rapport à un autre corpus ("corpus de référence") : si son Zscore est élevé, le mot est sur-représenté dans le corpus à étudier, si le Zscore est faible (celui-ci pouvant être négatif), le mot est sous-représenté dans le corpus à étudier. Dans (Savoy, 2009), cette mesure a, par exemple, été utilisée en comparant des corpus représentant des discours et programmes électoraux des quatre grands partis politiques suisses (les textes d'un parti servent de corpus à étudier, ceux des trois autres sont utilisés comme référence), afin de faire ressortir les mots sur (ou sous)-employés par chacun des partis par rapport aux trois autres, que ces termes soient représentatifs de thématiques ("culture", "culturelle", "artiste", "autogestion" vs "armée", "défense", "sécurité", "militaire", pour des morceaux choisis parmi les 10 vocables les plus représentatifs, les uns des discours du Parti Socialiste, les autres (lesquels ?) du Parti Radical Démocratique) ou représentatifs de manières de s'exprimer (sur-représentation de "nous" pour le Parti Du Centre).
  Dans nos expériences, le but étant de faire ressortir des termes propres au domaine étudié, nous avons comparé le corpus sur lequel nous travaillions avec un corpus généraliste dans la même langue. Le corpus le plus généraliste auquel nous avons pensé est tout naturellement Internet, dont des échantillons sont disponibles grâce au corpus ClueWeb09 [1], régulièrement utilisé pour les campagnes d'évaluation TREC (Clarke et al., 2009). Ce corpus a été constitué en amassant des sites Web découverts en parcourant des pages de lien en lien. Il existe des échantillons disponibles pour une dizaine de langues. Nous avons ainsi obtenu une liste ordonnée des mots composant le corpus à étudier et choisi de sélectionner les x plus hauts Zscores, indiquant les mots les plus représentatifs du domaine.

1. http ://lemurproject.org/clueweb09/

$$Zscore(\omega) = \frac{a - n' \times p}{\sqrt{n' \times p \times (1-p)}} \tag{4.1}$$

où :
$p = (a+b)/n$
a est le nombre d'occurrences du mot ω dans le corpus à étudier
b est le nombre d'occurrences du mot ω dans le corpus de référence
n est le nombre total de mots des deux corpus (corpus à étudier + corpus de référence)
n' est le nombre total de mots du corpus à étudier

Ces deux approches (automatique et manuelle) ont conduit, pour chaque corpus, à obtenir deux listes différentes de sujets. Ces listes n'ont pas été mixées, mais ont été utilisées dans des expériences différentes, dans un but de comparaison.

Relativement à l'approche par extraction automatique, un autre corpus facilement procurable, relativement généraliste et disponible dans un plus grand nombre de langues (pour conserver au maximum l'aspect générique/transposable de nos méthodes) est l'encyclopédie Wikipedia. Nous l'avons cependant écarté pour les trois raisons suivantes. Premièrement, ce corpus n'est pas autant généraliste que voulu : en tant qu'encyclopédie, il recouvre plutôt des sujets... encyclopédiques ; ainsi on peut s'attendre à trouver en grande proportion des articles relatifs au cinéma (qui sera le thème d'un des corpus sur lequel nous travaillerons), fiches des films, biographies d'acteurs, réalisateurs, doubleurs, etc., ce qui fait que lorsque l'on comparera le corpus étudié avec ce corpus de référence, les termes propres à ce domaine ne ressortiront pas forcément en tête de liste. Deuxièmement, toujours en tant qu'encyclopédie, les articles sont censés être objectifs, et, si ce n'est pas le cas, cela est plutôt fait dans un style encyclopédique (on va mettre en valeur certaines parties de la biographie de quelqu'un, passer sous silence certains aspects peu glorieux, etc.), mais se jouera moins souvent sur le vocable employé (on ne trouvera pas "le **grandissime** acteur Christian Clavier" (étonnamment)), ainsi, les termes porteurs d'opinions présents dans le corpus d'opinion (critiques, enquêtes) étudié seront sur-représentés par rapport au corpus Wikipedia. Enfin, en grande partie par le fait que l'encyclopédie est collaborative, les articles sont globalement écrits dans un langage académique, pour ce qui est de la construction des phrases et le respect des règles grammaticales et orthographiques : le corpus est donc beaucoup moins bruité que ne peut l'être un corpus "spontané", qu'il soit écrit (liberté dans l'orthographe pour les articles de blogs, critiques d'internautes) ou oral ("euh") ; ainsi, si on compare un corpus moins respectueux des règles de la langue (ce qui est fréquemment le cas en analyse d'opinion) à un corpus encyclopédique, on pourra retrouver comme sur-représentés certains néologismes ou termes moins châtiés, sans qu'ils soient pour autant représentatifs de la thématique considérée. La comparaison avec un corpus de type Web permet de s'affranchir de ces problèmes en étant plus proche, d'un point de vue linguistique, de ce que l'on pourra avoir dans les corpus que nous étudierons (quels que soient ces corpus finalement, le Web étant plus représentatif de tous les types de langages, de par sa nature plus hétéroclite).

4.2.2 Trouver les n-grammes porteurs d'opinion

Pour définir les n-grammes porteurs d'opinion, nous avons exploré le corpus d'apprentissage de la manière suivante : pour chacun des sujets connus, nous avons extrait tous les n-grammes apparaissant dans son contexte, en considérant les 5 mots précédant le sujet et les 7 suivants. Nous avons choisi des tailles de fenêtre différentes en partant du pré-supposé qu'une opinion est plus proche de son sujet quand elle se trouve à sa gauche plutôt qu'à sa droite (*a bad movie* vs *this movie is bad*). De plus, il arrive, notamment en français, que l'on critique assez loin vers la droite (*Le scénario décousu ne ressemble absolument à rien.*) ce genre de formulation nous semblant se retrouver moins en contexte gauche. Par "tous les n-grammes apparaissant dans son contexte", nous entendons : tous les n-grammes, pour toutes les valeurs possibles de n, de 1 à la taille de la fenêtre, non-nécessairement contigus au sujet, mais composés de mots qui eux le sont entre eux. Ainsi, considérant l'extrait suivant :
... is a really fun action movie ...
movie appartient à la liste des sujets connus ; dans le contexte gauche, pour une fenêtre de taille 5, nous extrayons :
unigrammes : *is, a, really, fun* and *action*
bigrammes : *is a, a really, really fun* and *fun action*
trigrammes : *is a really, a really fun* and *really fun action*
quadrigrammes : *is a really fun* and *a really fun action*
pentagramme : *is a really fun action*

Nous obtenons à cette étape deux listes de n-grammes, une pour le contexte gauche (contenant tous les n-grammes - de l'unigramme au pentagramme - trouvés dans le contexte gauche d'au moins un des sujets dans le corpus d'apprentissage) et une pour le contexte droit (tous les n-grammes, de l'unigramme à l'heptagramme, trouvés dans le contexte droit d'au moins un des sujets). Pour chaque liste, nous avons appliqué, de manière distincte, la procédure suivante.
Premièrement, pour chaque n-gramme, nous avons calculé trois valeurs :

- le nombre de documents dans lequel le n-gramme de contexte gauche (respectivement droit) apparaît dans le contexte gauche (respectivement droit) d'au moins un des sujets prédéfinis ; c'est-à-dire que le n-gramme peut se trouver dans le contexte d'un sujet différent que celui qui nous a permis initialement de l'extraire, on le comptera quand même (il s'agit en fait de voir si ce n-gramme est souvent proche d'un des sujets, quel qu'il soit) - si ce n-gramme apparaît à l'intérieur du même document dans le contexte de plusieurs sujets (différents ou le même répété) on n'incrémente que de 1 (il s'agit du nombre de documents, pas du nombre de contextes).
- l'IDF du n-gramme i sur l'ensemble du corpus d'apprentissage, selon $IDF(i) = -\log\left(\frac{\#docs\ contenant\ i}{\#docs}\right)$.
- le critère de pureté de Gini, représentant le pouvoir discriminant du n-gramme i selon sa répartition dans les diverses opinions k (positif, négatif), par $pGini(i) =$

$\sum_k P(k|i)^2$; la probabilité $P(k|i)$ est calculée sur l'ensemble du corpus d'apprentissage, en considérant une seule occurrence de i par document (par exemple, si i apparaît deux fois dans un même document étiqueté comme positif et une fois dans trois documents négatifs, alors $pGini(i) = (1/4)^2 + (3/4)^2$).

Puis, nous avons utilisé trois seuils pour filtrer les n-grammes selon ces critères : ils doivent être présents dans le contexte d'un sujet dans un certain nombre de documents, l'IDF est utilisé pour supprimer les n-grammes trop courants et la valeur de *pGini* nous donne une indication sur le fait que le n-gramme soit principalement employé dans les documents appartenant à une certaine opinion ou non. Les n-grammes passant ces trois seuils sont alors considérés comme porteurs d'opinion. Nous obtenons ainsi deux listes de n-grammes porteurs d'opinions (une par contexte), indépendamment des sujets autour desquels ils ont pu apparaître.

4.2.3 Extraire les segments

À l'issue des deux étapes précédentes, nous disposons d'une liste de sujets et deux listes de n-grammes considérés comme porteurs d'opinion. Nous parcourons alors le corpus de test, en quête d'un des sujets prédéfinis. Quand nous en trouvons un, nous recherchons dans son contexte gauche (respectivement droit), tous les n-grammes porteurs d'opinion en contexte gauche (respectivement droit) sélectionnés par la méthode décrite en sous-section précédente. Bien entendu, nous faisons de même pour tous les sujets que nous reconnaissons dans le corpus (y compris si le même sujet est présent plusieurs fois dans le même document). Pour chaque paire (sujet, n-gramme d'opinion) que nous trouvons, nous extrayons le segment de texte s'étendant du sujet au n-gramme porteur d'opinion (ou du n-gramme d'opinion au sujet, s'il s'agit d'un n-gramme de contexte gauche). Grâce à cette méthode, nous pouvons extraire des expressions jamais rencontrées dans le corpus d'apprentissage : le n-gramme porteur d'opinion avait été retenu car on l'avait rencontré dans le contexte de sujets différents de celui à côté duquel il apparaît dans le test (c'est-à-dire qu'on considère que les n-grammes porteurs d'opinion sont propres à un domaine, mais pas à un sujet précis dans ce domaine), de plus, ce qui sépare le sujet du n-gramme porteur d'opinion peut être différent. Ces mots intermédiaires peuvent être inutiles à l'expression d'opinion, peuvent permettre des variantes, des nuances, des négations. Dans ce dernier cas, on espère que la forme avec négation a été rencontrée dans le corpus d'apprentissage (ce sera le cas pour les plus courantes) et qu'en plus du n-gramme porteur d'opinion en question, on aura aussi retenu dans nos listes le même n-gramme avec la marque de négation (son nombre d'occurrences sera plus faible, mais son *pGini* plus élevé). À l'heure actuelle, notre système présente dans ce cas les deux segments, en indiquant pour quelles raisons il a retenu chacun des deux (permettant de voir que celui avec négation possède un *pGini* plus élevé, ce qui peut être un critère de sélection pour choisir de ne retenir que celui-ci), ceci fait partie des problèmes de recouvrements inhérents à la méthode qui seront discutés ultérieurement dans ce chapitre.

Pour des raisons de lisibilité, nous étendons ensuite ce segment, en y ajoutant le mot

précédent et les deux mots suivants, parce que nous avons remarqué que cela aidait à la compréhension dans un certain nombre de cas. Par exemple, à partir du sujet *movie* et du n-gramme d'opinion *exceptional*, nous avons pu extraire *the [movie and make it an **exceptional**] romantic comedy* : le segment originel étant entre crochets, l'ajout de *romantic comedy* permet de mieux comprendre de quoi il s'agit.

4.3 Expériences

Nous avons appliqué la méthode sur deux corpus différents : des critiques de films en anglais et une enquête de satisfaction d'entreprise en français. Les seules connaissances externes utilisées sont : le corpus généraliste de référence (ici le Web) et la lemmatisation, effectuée par TreeTagger, sans utiliser les informations morpho-syntaxiques fournies sur les mots. Dans les cas où TreeTagger proposait plusieurs lemmes pour un même mot, nous avons arbitrairement choisi de conserver le dernier (d'après l'observation d'un échantillon de ces cas, il nous a semblé que celui-ci était le plus souvent pertinent, sans que la différence ne soit réellement significative). Ceci pourrait être évité, ce qui impliquerait d'ajouter toutes les formes fléchies possibles des sujets dans la liste des sujets sur lesquels on recherche des opinions et d'adapter les seuils utilisés pour le filtrage des n-grammes porteurs d'opinion.

4.3.1 Movies Polarity Dataset v2.0

4.3.1.1 Description

Nous avons travaillé sur le corpus *Movies Polarity Dataset v2.0* (Pang et Lee, 2004), déjà utilisé en section 3.3.3, avec la même répartition (1200 documents pour l'apprentissage, 200 pour le développement et 600 pour le test - contenant chacun 50% de critiques positives et autant de négatives).

4.3.1.2 Sujets

La liste des sujets définis manuellement est composée de 40 lemmes liés au domaine du cinéma, certains généraux (*movie, film*), d'autres plus spécifiques (*director, actor, cast, image, scenario, dialog, photography*), mais nous avons aussi ajouté des termes qui sont couramment employés comme base pour l'expression d'une opinion (par exemple, *oscar*, comme dans *it doesn't deserve an oscar*). Dans ce cas, l'appellation de "sujet" est légèrement inadéquate et on pourrait plutôt parler de "déclencheur".

La liste complète est la suivante : *dialogue, photography, oscar, storyline, dialog, movie, music, filmography, subject, mise, actress, shoot, price, comedy, performer, set-up, cast, story, scenario, shooting, edit, director, thriller, direction, editor, film, interpretation, act, script, scene, play, award, camera, image, actor, parody, costume, plot, action, soundtrack*.

Pour obtenir la liste des sujets extraits de manière automatique, nous avons calculé le Zscore (comme présenté en secion 4.2.1), en comparant les mots du corpus d'apprentissage avec un sous-ensemble du corpus ClueWeb09, consistant en 29 millions de pages Web en anglais. Il s'agit des 50 millions de pages du sous-ensemble "Catégorie B", auxquelles ont été retirées les 21 millions de pages considérées comme spam selon la catégorisation proposée par (Gordon V. Cormack, 201), quand le score retourné par la fusion des classifieurs est inférieur à 70 [2]. Les nombres exacts de documents et de mots du corpus résultant sont présentés en tableau 4.1.

nombre de documents	29 038 220
nombre de mots uniques	33 314 740
nombre de mots total	22 814 465 842

TABLE 4.1 – *Statistiques générales sur l'échantillon anglais du corpus ClueWeb09.*

Pour des raisons techniques, la version du ClueWeb09 dont nous disposons a été indexée avec Indri (Strohman et al., 2005) en utilisant les racines des mots, obtenues par le stemmer de Krovetz et un anti-dictionnaire (stoplist) contenant les ponctuations et certains mots-outils. Toutefois, ces mots-outils ne seraient pas censés avoir eu un Zscore élevé, dans la mesure où ils sont aussi très fréquents sur Internet. Nous avons choisi de ne considérer que les stemmes apparaissant au moins 100 fois dans le corpus d'apprentissage, ce pour deux raisons : premièrement, si nous avions conservé tous les mots, un néologisme employé par son seul créateur [3] se serait vu attribuer un Zscore élevé étant donné que sa fréquence n'est pas beaucoup plus élevée sur Internet (en expérimentant sans ce filtre, nous avions vu ressortir le nom du célèbre réalisateur *hichtcoc*) ; deuxièmement, nous n'avons pas voulu admettre qu'un mot (ou en ensemble de formes fléchies dont la racine est présente) présent moins de 100 fois dans un corpus de plusieurs milliers de critiques puisse être considéré comme un sujet sur lequel on exprime souvent une opinion (sachant que l'on compte toutes les occurrences des mots, y compris plusieurs fois par document). Nous avons ainsi obtenu une liste ordonnée, selon leur Zscore, des termes du corpus.

Nous avons conservé les 60 premiers stemmes, correspondant à 101 lemmes. La plupart d'entre eux sont effectivement relatifs au domaine cinématographique (plus exactement à la critique cinématographique), qu'ils correspondent à notre vision d'un sujet [4] ou non [5]. Nous trouvons aussi un certain nombre d'éléments plus liés à l'aspect critique qu'à l'aspect cinématographique (*funny* en 7^{e}, *bad* en 9^{e}). Enfin, il ressort aussi des stemmes correspondant à notre idée de ce que peut être un sujet mais qui nous avaient échappé à l'extraction manuelle (*filmmaker* en 46^{e}).

Ces 101 lemmes sont : *sequel, suspense, sequence, scream, guy, cleverly, obviously, ridiculousness, screenwriter, dumb, bored, moment, casting, cop, role, funny, story, goodness, humor, bore, thriller, makings, flick, laughing, stupid, good, bad, making, watch, jackie, film,*

2. comme recommandé sur le site des auteurs : http ://plg.uwaterloo.ca/ gvcormac/clueweb09spam/
3. Il en aurait été de même pour une faute de frappe peu fréquente
4. Les 5 premiers stemmes de la liste ordonnée sont *film, movie, scene, character, plot*
5. *alien* figure en 13^{e} position

ending, alien, clich, unfortunately, storey, unfortunate, obvious, ridiculously, hilarious, actor, man, pretty, watching, plot, stupids, man's, titanics, cliche, dialogue, stupidity, titanic, stupidly, badness, cleverness, bear, movie, cliched, dumbing, stupidness, awfulness, obviousness, character, manly, playing, screenwriterly, killer, end, thrillerism, girlfriend, hilariously, comedy, annoyed, cast, filmmakers, predictable, audience, predictably, director, badly, awfully, joke, horror, filmmaker, laugh, guiness, screenplay, playes, make, script, boring, awful, scene, clever, play, ridiculous, predictability, thing, annoy, performance, hollywood.

Le même corpus a été parallèlement racinisé avec le stemmer de Krovetz (pour pouvoir être comparé au ClueWeb indexé) et lemmatisé avec TreeTagger : en partant des formes fléchies, nous avons établi une correspondance entre une racine et ses différents lemmes. Au final, quand le Zscore propose un stemme, nous considérons que toutes les formes lemmatisées qui y correspondent sont des sujets. Cela entraîne certains problèmes. Ainsi, un exemple de cas qui n'a pas été bien traité est *thrillerism*, devenu *thriller* dans sa forme racinisée pour le calcul du Zscore. Quand nous avons recherché toutes les formes lemmatisées de *thriller* qui a obtenu un Zscore élevé, nous avons retrouvé *thrillerism*, qui a ainsi été ajouté à la liste des sujets, ce qui n'est pas forcément pertinent - nous envisageons, en même temps que les prochaines améliorations du processus, de travailler à partir d'une version lemmatisée du ClueWeb (dont nous ne disposons pas à l'heure actuelle). D'une manière générale, tous les mots présents dans cette liste qui ont l'air un peu ésotériques (*playes*, *manly*...), proviennent de la combinaison lemmatisation-racinisation (avec des systèmes différents et leurs erreurs respectives).

Cela fait partie des raisons qui nous ont poussé à extraire plus de sujets par la méthode automatique que par la méthode manuelle. Les autres étant le fait que la méthode manuelle est consommatrice de ressources (en temps-homme), alors que le Zscore nous permet d'obtenir un grand nombre de termes relatifs au domaine sans intervention humaine (un simple coup d'œil suffit pour s'apercevoir qu'au-delà du 60^{e}, la majorité des stemmes ne semblaient plus pertinents) ; de plus, ces quantités différentes peuvent être utilisées pour étudier les variations de couverture (nombre de n-grammes porteurs d'opinion que l'on peut extraire d'après ces sujets, nombre de segments correspondants).

Seuls 13 lemmes sont présents à la fois dans la liste des sujets définis manuellement et dans celle des sujets extraits automatiquement. Cela est selon nous dû à deux raisons :

– dans la méthode manuelle, nous commençons par définir une première liste de termes, selon nos connaissances du domaine, des concepts sur lesquels une opinion peut être exprimée, de ceux sur lesquels il nous plairait de connaître les avis exprimés (avant de lire un échantillon du corpus afin d'en ajouter de nouveaux), cependant certains de ces termes peuvent être sous-représentés dans le corpus. Par exemple, nous avons décidé que le *scenario* était un élément important dans une critique de film, mais peut-être que les internautes ne l'entendent pas de cette oreille, ou encore qu'ils préfèrent en parler en termes de *story* ou de *script*, qui eux sont favorablement considérés par le Zscore. De même, nous avons défini *subject* en tant que sujet potentiellement intéressant, mais il ne se trouve qu'en 586^{e} position dans la liste des Zscores (avec 107 occurrences dans le corpus d'apprentissage).

– la méthode automatique attribue des Zscores élevés à tous les stemmes sur-représentés dans ce corpus, nous obtenons donc des termes propres à la ***critique** de **films***, incluant ainsi notamment des adjectifs et adverbes eux-mêmes porteurs d'opinion ou encore des termes assez spécifiques tels que des noms propres (un réalisateur ou titre de film souvent cités), alors que dans la méthode manuelle nous avons choisi de ne considérer que les hyperonymes (*movie, director*).

Nous avons tenté quelques approches simples basées sur des pondérations selon les TF, IDF, pGini (permettant d'écarter les termes discriminants, tels que les adjectifs subjectifs) pour rapprocher les résultats automatiques de ceux de la liste manuelle, ce que nous n'avons pas réussi à faire de manière significative. Cependant, cette idée n'a pas été explorée plus en avant, notamment car on peut s'interroger sur la pertinence d'un tel objectif : peut-on considérer que la liste manuelle est une référence, à la fois en termes de précision (voir l'exemple de *scenario*) et en termes de rappel, la méthode automatique nous permettant justement de faire ressortir des éléments auxquels nous n'avions pas pensé intuitivement (*filmmaker*).

4.3.1.3 N-grammes porteurs d'opinions

Nous avons appliqué la méthode présentée en section 4.2.2, avec les paliers suivants : le n-gramme doit être présent dans au moins 5 documents dans le contexte d'un sujet, son IDF doit être supérieur à 1,99 et son pGini à 0,65.

Nous avons obtenu :
– pour les sujets extraits manuellement : 254 n-grammes utilisés pour exprimer une opinion dans le contexte gauche d'un sujet et 354 dans le contexte droit ; certains d'entre eux se recoupant, par exemple *bad* et *be so bad*.
– pour les sujets extraits automatiquement : 573 dans le contexte gauche et 872 dans le contexte droit ; nous extrayons plus de n-grammes dans ce cas étant donné que nous avons conservé les mêmes filtres tout en se basant sur plus de sujets (il y a donc plus de chances que les n-grammes apparaissent dans le contexte d'un sujet).

Parmi ces n-grammes, on peut distinguer plusieurs sous-ensembles :
– les n-grammes couramment employés pour exprimer une opinion, reposant généralement sur des adjectifs connus pour être orientés, par exemple : *waste, be terrible, stupidity, a pointless* pour les négatifs, ou *make sense, realistic, be an excellent* pour les positifs.
– ceux utilisés avec une polarité dans ce corpus, bien que dans d'autres contextes ils puissent être considérés comme neutres : *hollywood's* (qui est généralement employé dans ce corpus de manière positive : *the hollywood's new dramatic actor* ou *one of hollywood's favorite author* - aucune de ces expressions n'apparaît suffisamment de fois pour être reconnue par notre méthode, donc on capture juste *hollywood's*), *italian* (généralement associé aux opinions positives, que cela concerne un personnage, un style, un acteur ou un réalisateur).
– ceux régulièrement présents dans le contexte d'un sujet et dans un type de critique (positive ou négative), mais qui n'expriment pas réellement une opinion :

could have be, nothing to do with the annoncent une opinion négative, mais nous ne savons pas laquelle, ni quel est son support.
– des noms propres qui sont le plus souvent cités dans les critiques positives (comme *tom hanks*) ou dans les négatives (*john frankenheimer*) - en tout cas dans le contexte d'un sujet.

4.3.1.4 Segments textuels

En utilisant les sujets définis manuellement, nous reconnaissons en moyenne 19,7 sujets par document, permettant d'extraire 2,3 segments (sujets comprenant un des n-grammes porteurs d'opinion connu dans leurs contextes). Avec les sujets extraits automatiquement, nous retrouvons 36 sujets par document conduisant à 6,1 segments (il y a plus de sujets et de n-grammes à rechercher).

Nous présentons ici quelques exemples des segments extraits sur des documents appartenant au corpus d'apprentissage, dans le cas des sujets définis manuellement. Les sujets sont soulignés et les n-grammes porteurs d'opinion sont en gras ; une valeur positive (respectivement négative) indique que le n-gramme a été majoritairement rencontré dans des documents positifs (respectivement négatifs) dans le corpus d'apprentissage :

document id : 725_10103 **catégorie :** positif
– *the [movie and make it an* ***exceptional****] romantic comedy*, pGini : +0,76
– *the [movie* ***be perfect****], from*, pGini : +0,82
– *an [****exceptional*** *romantic comedy]. everything*, pGini : +0,76
– *. [****everything about*** *the movie] be perfect*, pGini : -0,65

Dans cet exemple, le premier segment a été retourné car le système a trouvé *exceptional* (qui est connu pour être positif avec un pouvoir discriminant de 0,76 - il est à noter que ce pouvoir ne représente pas la force de l'opinion sur une échelle de valeurs du type c'est un peu positif, moyennement positif, beaucoup positif, ... mais uniquement la répartition du mot dans les différentes catégories, selon l'observation qui en a été faite dans le corpus d'apprentissage) dans le contexte de *movie* - les deux mots suivants ajoutés au segment (*romantic comedy*) sont utiles pour permettre une meilleure compréhension ; *comedy* appartient aussi à la liste des sujets et *exceptional* est à la fois dans la liste des n-grammes porteurs d'opinion en contexte gauche et en contexte droit, nous obtenons donc le troisième segment. Le second segment nous indique juste que le film est parfait, mais un contexte plus large pourrait être intéressant. Le quatrième segment est associé aux opinions négatives car le n-gramme *everything about* se retrouve principalement dans ce genre de critiques. Celui-ci est inutile est aurait pu être évité en utilisant un filtre plus strict (son pGini est de 0,65, ce qui est égal au minimum requis). De plus, ce segment appartient à la même phrase que le second "everything about the movie is perfect". Ceci est un exemple de cas de recouvrement.

Dans l'exemple suivant, nous présentons les segments extraits sur une critique négative :
document id : 994_13229 **catégorie :** négatif

– *the [film be outright **stupid**] , but*, pGini : -0,7
– *'s [**inane** dialogue] be drown*, pGini : -0,72
– *this [**awful** script] ! ?*, pGini : -0,74

Ici, le système a correctement extrait trois points négatifs cités à propos de différents aspects du film : une opinion générale concernant le *film* et des plus spécifiques à propos du *dialogue* et du *script*. Le premier segment représente bien l'intérêt, d'extraire une portion du texte et non pas uniquement un couple (sujet ; opinion) : il est assez compliqué de créer un système arrivant à reconnaître toutes les nuances employées par 2 milliards de personnes qui ont accès à Internet, nous pouvons ici visualiser cette nuance, le film est catégoriquement stupide. Les n-grammes exprimant des opinions sont ici seulement des adjectifs (unigrammes). Dans l'exemple qui suit, nous pouvons trouver d'autres types de n-grammes :

document id : 991_19973 **catégorie :** négatif
– *, [**recycle** the classic story] of a*, pGini : -0,68
– *some [**of the bad** dialogue] write in*, pGini : -0.92
– *the [**repetitive** dialogue] from the*, pGini : -0,66
– *- [**avoid this** movie] at all*, trigger : -0.68

Encore une fois, nous pouvons observer des jugements à propos de points précis, l'histoire, le dialogue (commenté deux fois) et un conseil général concernant le film. Le dernier exemple sur les sujets manuels présente des segments construits avec des n-grammes qui n'expriment pas spécialement une opinion (*doesn't even*, *bother to*, *absolutely nothing* and *whatsoever*) :

document id : 995_23113 **catégorie :** négatif
– *the [film **doesn't even**] bother to*, pGini : -0,71
– *the [film doesn't even **bother to**] explain what*, pGini : -0,79
– *horror [film , because there be **absolutely nothing**] scary about*, pGini : -0,67
– *the [scene have any dramatic weight **whatsoever**] . the*, pGini : -0,73

Pour le dernier segment, les mots contenus entre le sujet *scene* et l'unigramme *whatsoever* permet de comprendre le problème exprimé. Pour le troisième segment, le contexte étendu est nécessaire pour comprendre ce qui est discuté (le problème est que c'est censé être un film d'horreur mais qu'il n'y a absolument rien d'effrayant). Les deux premiers segments sont construits sur *doesn't even* et *bother to*. Ceci est un autre exemple de cas de recouvrement. Celui-ci est inutile, car avoir ces deux segments ne nous aide pas à obtenir un meilleur aperçu de ce qui est critiqué.

Nous présentons maintenant quelques exemples obtenus à partir de l'extraction automatique des sujets. Le premier correspond aux segments trouvés pour le document 991_19973, sur lequel nous avons précédemment commenté les segments basés sur les sujets définis manuellement :

document id : 991_19973 **catégorie :** négatif
– *, [**recycle** the classic story] of a*, pGini : -0,68
– *some [**of the bad** dialogue] write in*, pGini : -0.92
– *the [**bad dialogue** write in a film] within recent*, pGini : -0,88
– *the [**repetitive** dialogue] from the*, pGini : -0,66

– *- [**avoid this** movie] at all*, pGini : -0,68

Sur ce document, les segments obtenus à partir des sujets extraits automatiquement sont assez similaires à ceux basés sur les sujets manuels. En fait, il y a juste un segment supplémentaire reposant sur *bad dialogue* que l'on trouve dans la liste des n-grammes porteurs d'opinions en contexte gauche des sujets automatiques mais pas des manuels.

Il y a aussi des documents pour lesquels les segments sont plutôt inintéressants, par exemple :
document id : 702_11500 **catégorie :** positif

– *some [pretty wicked story twist that we **really don't**] see come*, pGini : -0,68
– *wicked [story twist that we **really don't**] see come*, pGini : -0,68
– *to [play a small **part in the**] stage kidnap*, pGini : +0,70
– *the [movie **work well**] on its*, pGini : +0,72
– *. [**thank to a** juicy cast] – especially*, pGini : +0.68

La majorité des segments obtenus pour ce document sont inutiles. Qui plus est, les deux premiers sont considérés comme négatifs - alors que le document est étiqueté positif - car ils sont basés sur le n-gramme *really don't*. Il a ici été retrouvé dans le contexte de deux sujets définis automatiquement : story et *pretty*, lequel était malheureusement présent dans les 60 premiers rangs du Zscore.

Toutefois, avoir à la fois des segments positifs et négatifs (en réalité, c'est le n-gramme qui est considéré comme tel), peut-être intéressant dans la mesure où toutes les critiques ne sont pas extrêmes : même si on considère que le film est globalement mauvais, il peut y avoir certains aspects qui le sauvent du désastre, ou, à l'inverse, un bon film peut tout de même pêcher par certains points. Considérons :
document id : 838_25886 **catégorie :** négatif

– *the [action scene (though technically **excellent**]) be*, pGini : +0,66
– *action [scene (though technically **excellent**]) be*, pGini : +0,66
– *and [story update simply **embarrassing**] . to*, pGini : -0,71
– *this [film can boast **of great**] sound ,,* pGini : +0.76
– *achievement [**save this** film] from be*, pGini : -0.71
– *to [**be a fun** action] film .*, pGini : +0,86
– *to [**be a fun** action film] . it*, pGini : +0,86

Sur cette critique négative, le troisième segment exprime un jugement négatif, mais il y a aussi des remarques positives à propos des scènes d'action, de la bande-son, etc. Nous apprenons aussi que *achievement save this film*, ce qui est cependant considéré comme négatif par le système. Les deux premiers segments sont basés sur le même adjectif d'opinion (*excellent*) rencontré à la fois dans le contexte de *action* et *scene*. Ces deux segments auraient pu être fusionnés.

4.3.2 NPS07-09

4.3.2.1 Description

Nous appliquons ici la méthode sur une sous-partie du corpus tiré de l'enquête Net Promoting Score, présentée en section 3.3.2. Parmi les 12 124 entretiens dont nous disposions, les transcriptions les plus courtes (moins de 500 caractères) des réponses à la question Q2 (Quels sont les facteurs qui motivent la note de satisfaction attribuée à la question précédente ?) ont été annotées selon la polarité majoritairement exprimée : positive (pos), négative (neg), ou la réponse contient à la fois des éléments positifs et des éléments négatifs (posneg). Les 3814 premiers entretiens (1301 neg, 2034 pos, 479 posneg) de l'année 2007 nous ont servi de corpus d'apprentissage, les 1000 suivants (331 neg, 566 pos, 103 posneg) de développement et les 1427 (825 neg, 518 pos, 84 posneg) entretiens de 2008 et 2009 de corpus de test.

4.3.2.2 Sujets

La liste des sujets manuellement définis est composée de 26 termes relatifs à l'entreprise, au service et à la qualité de contact : *temps, réponse, répondre, edf, qualité, prestation, demande, accueil, délai, accueillir, contact, attendre, traiter, traitement, expliquer, explication, relationnel, relation, échange, suivi, renseignement, renseigner, gens, communication, démarche, dialogue.*

Pour obtenir la liste des sujets automatiquement définis, nous avons calculé le Zscore en comparant les racines (obtenues par le stemmer de Krovetz) des mots composant le corpus d'apprentissage à un sous-ensemble du corpus ClueWeb09, consistant à environ 51 millions de pages Web en français (des statistiques sur cet échantillon sont présentées en tableau 4.2). Nous avons choisi de ne considérer que les stemmes présents au moins 50 fois dans le corpus d'apprentissage. Nous avons conservé les stemmes ayant les 50 Zscores les plus élevés, ce qui correspond à 55 lemmes dans le corpus : *compteur, obligé, expliquer, interlocuteur, raccordement, duplicater, agréable, rappeler, d'interlocuteurs, réponse, falloir, faire, personne, branchement, edf, duplicata, réclamation, l'interlocuteur, satisfait, délai, l'électricité, téléphonique, facturer, facturation, satisfaire, rapidement, passer, contact, prélèvement, téléphone, appeler, demander, recevoir, d'edf, demande, devoir, problème, rapidité, répondre, d'attente, bien, téléphoner, réactivité, contrat, d'attentes, l'accueil, coupure, obliger, l'écoute, facture, renseigner, fait, aimable, passé, d'interlocuteur.* On note pour certains éléments de cette liste, la présence d'un déterminant précédant le nom ("*d'edf*"). Il s'agit d'erreurs commises par le système de lexicalisation pour le français. Cela entraîne quelques problèmes mineurs : premièrement, étant donné que cette séparation en mots est faite de manière différenciée pour la recherche des sujets (par Indri) et pour les travaux ultérieurs (par TreeTagger), les mots précédés d'un déterminant-apostrophe ne se retrouveront probablement pas dans les corpus que nous traiterons par la suite (si TreeTagger ne commet pas les mêmes erreurs), ce qui en soi n'est pas vraiment un problème, on a juste des sujets qui seront inutilisés. Deuxièmement, pour le calcul du Zscore, les mots "*d'edf* et "*edf*" correspondent à deux entrées différentes,

leurs comptes ne sont donc pas additionnés, ce qui leur aurait permis de se retrouver plus haut dans la liste des meilleurs Zscores (à l'un d'entre eux en tout cas, l'autre aurait disparu). Ceci signifie également qu'il se peut que certains mot qu'il aurait été pertinent de retrouver dans la liste des sujets n'y figurent pas car leurs comptes avec et sans apostrophes n'ont pas été cumulés.

Pour les mêmes raisons que celles avancées pour le corpus précédent, seuls 8 lemmes se retrouvent à la fois dans la liste des sujets définis automatiquement et ceux définis manuellement.

nombre de documents	508 83 172
nombre de mots uniques	50 070 867
nombre de mots total	35 746 680 579

TABLE 4.2 – *Statistiques générales sur l'échantillon français du corpus ClueWeb09.*

4.3.2.3 N-grammes porteurs d'opinions

Pour ce corpus, les paliers ont été fixés de telle sorte que : le n-gramme doit apparaître dans le contexte d'un sujet dans au moins trois documents, son IDF doit être supérieur à 2,1 et son pGini supérieur à 0,681. Pour le calcul du pGini, on ne considère pas les documents étiquetés comme posneg, puisqu'ils sont à la fois positifs et négatifs, donc cela réduirait le pouvoir discriminant de ce critère. Nous obtenons ainsi :

- pour les sujets extraits manuellement : 1250 n-grammes porteurs d'opinion dans le contexte gauche et 1810 dans le droit ;
- pour les sujets extraits automatiquement : 3093 n-grammes porteurs d'opinion dans le contexte gauche et 4629 dans le droit.

4.3.2.4 Segments textuels

En utilisant les sujets extraits manuellement, on capture 1,6 sujets par document, apportant 3,5 segments. Avec les sujets automatiques, on capture 3,9 sujets par document, donnant 12,1 segments (beaucoup de recouvrements). Par rapport au corpus précédent, on trouve beaucoup moins de sujets par document. Ceci s'explique par deux raisons : il y a moins de sujets dans nos listes initiales et le corpus présente une chronologie, donc il se peut que les sujets extraits sur le corpus d'apprentissage soient devenus obsolètes deux ans plus tard (un sujet qui n'est plus discuté, un problème ponctuel résolu depuis, ...).

Voici un exemple de segments extraits pour un document, présentant un recouvrement :

document id : 9173 **catégorie :** pos

– *savoir [répondre , précis et **concis**] .*, pGini : +1

– *, [**il avoir bien** savoir répondre] , précis*, pGini : +0,94

– *il [**avoir bien** savoir répondre] , précis*, pGini : +0,89

Dans ce cas, les trois segments que l'on a trouvés pour cette réponse tournent autour du sujet *répondre*, le premier trouvé en raison de l'opinion *concis*, le second par le n-gramme *il avoir bien*. Ces deux-ci peuvent être considérés comme intéressants : ils apportent deux informations complémentaires sur la qualité de la réponse fournie par l'agent, bien que l'on puisse éventuellement considérer qu'une des deux est superflue, il est difficile de décider laquelle, même pour un humain (est-il plus intéressant de savoir que le conseiller a donné une bonne réponse ou que la réponse était précise et concise, mais peut-être fausse ?). Le dernier segment présente pour sa part moins d'intérêt : il présente la même information que le deuxième, un des deux est donc inutile.

Les extractions obtenues sur le document suivant offrent plusieurs aspects intéressants à étudier :
document id : 8759 **catégorie :** pos

– *bon [contact **clair**] dans son*, pGini : +0,76

– *bon [contact clair dans son proposition avoir **bien répondre**] à tout*, pGini : +0,93

– *bon [contact clair **dans son**] proposition avoir*, pGini : +0,89

– *bon [contact clair dans son **proposition**] avoir bien*, pGini : +0,74

– *bon [contact clair dans son proposition **avoir bien répondre**] à tout*, pGini : +1

– *bon [contact clair dans son proposition **avoir bien**] répondre à*, pGini : +0,89

– *bien [répondre **à tout**] le question*, pGini : +0,76

– *bien [répondre **à tout le**] question .*, pGini : +0,82

– *bien [répondre **à tout le question**] .*, pGini : +1

– *bien [répondre à **tout le question**] .*, pGini : +1

– *son [**proposition** avoir bien répondre] à tout*, pGini : +1

– *clair [**dans son** proposition avoir bien répondre] à tout*, pGini : +0,89

– *proposition [**avoir bien** répondre] à tout*, pGini : +0,89

Les deux premiers segments retournés ainsi que *bon [contact clair dans son proposition **avoir bien répondre**] à tout* et *bon [contact clair dans son proposition **avoir bien**] répondre à* présentent le cas où le n-gramme porteur d'opinion n'est pas retourné avec le su-

jet sur lequel il porte réellement. Dans ces deux exemples ainsi que dans le deuxième de la liste, le système a fait correspondre *avoir bien répondre* avec le *contact*, alors que le client parlait du fait qu'avec la personne il avait eu un bon contact et que la personne avait bien répondu à sa demande. De même, sur le premier segment retourné, *clair* est considéré par le système comme portant sur *contact* alors qu'en réalité cela est l'expression d'une opinion sur la *proposition*. Les segments 3 et 4 sont construits à partir de n-grammes que l'on a considérés comme porteurs d'opinions, alors que nous pouvons voir qu'ils n'en sont pas (*dans son* et *proposition*). Le premier aurait pu être évité par un filtre plus restrictif sur les nombres d'occurrences (il n'apparaît que 17 fois dans le corpus d'apprentissage), le second par le pGini (celui-ci est de 0.74, *proposition* se retrouvant dans 50 réponses positives et 9 négatives). Enfin, les segments se basant sur le sujet *répondre*, sont retournés en raison de la présence dans leur voisinage de n-grammes référencés par le système comme étant porteurs d'opinion : *à tout*, *à tout le*, *à tout le question* et *tout le question*. Ces n-grammes ne sont pas, à nos yeux, porteurs d'opinion, cependant, ils sont statistiquement fortement corrélés à des opinions positives. Ainsi, quand la personne interrogée parle de "toutes les questions", c'est pour signaler que l'agent avec qui elle a eu un entretien a bien répondu à toutes ses questions (l'extrême opposé aurait été que l'agent n'ait répondu à *aucune* de ses questions). Le n-gramme *tout le question* est donc lié uniquement à la catégorie pos (pGini=1). Les n-grammes moins précis, tels que *à tout* sont moins discriminants (pGini de 0,76 pour celui-ci). Cependant, même si le système commet ce genre d'erreurs, les segments étendus permettent de comprendre de quoi il s'agit : *bien [répondre **à tout**] le question* -> le n-gramme porteur d'opinion n'est pas pertinent, mais grâce aux mots *bien* et *le question* on dispose d'une information intéressante sur la qualité du service.

De la même manière, dans l'exemple suivant, les expressions parlant de *la prise en charge* sont considérées comme porteuses d'opinion et liées au sujet *contact*. Étant donné que ces n-grammes ont été retenus, c'est qu'ils avaient un pGini supérieur au seuil à partir duquel on pouvait les considérer comme intéressants. De fait, *prise en charge* est cité 11 fois dans les commentaires positifs et 1 fois seulement dans les commentaires négatifs. On en déduit que quand les gens parlent de la prise en charge, c'est généralement pour en dire du bien.

document id : 8919 **catégorie :** pos
– *le [**prise en charge** le contact] .*, pGini : +0,85

– *: [**le prise** en charge le contact] .*, pGini : +0,70

L'exemple qui suit présente 8 n-grammes exprimant une opinion portant sur le renseignement, cependant seuls les derniers sont cohérents. La phrase complète est *il m'a bien renseigné et il a été aimable*. Or, si *renseigner* fait partie de la liste des sujets à rechercher, ce n'est pas le cas de *il* qui peut correspondre à beaucoup trop de concepts potentiellement inintéressants. Ceci ouvre des perspectives sur la résolution d'anaphores qui pourrait permettre de prendre en compte ce genre de cas. Ceci sera discuté en fin de chapitre.

document id : 9308 **catégorie :** pos

– *bien [renseigner et **il avoir être**] aimable :*, pGini : +1

– *bien [renseigner et il **avoir être aimable**] : nsp*, pGini : +1

– *bien [renseigner et il avoir **être aimable**] : nsp*, pGini : +0,94

– *bien [renseigner et **il avoir être**] aimable :*, pGini : +0,77

– *bien [renseigner et il avoir être **aimable**] : nsp*, pGini : +0,88

– *[**il m'a** bien renseigner] et il*, pGini : +0,71

– *il [**m'a bien** renseigner] et il*, pGini : +1

– *[**il m'a bien** renseigner] et il*, pGini : +1

Enfin, l'exemple suivant présente les segments extraits sur un document étiqueté posneg. Si certains d'entre eux sont basés sur des n-grammes positifs et d'autres des négatifs, ce n'est pas forcément à bon escient. Ainsi, le premier (**octobre**) est négatif - ce qui est souvent le cas des expressions temporelles - alors que le segment correspondant ne l'est pas obligatoirement (bien qu'il puisse s'agir de quelqu'un déclarant que le premier contact était suffisant, les suivants n'ont donc servi à rien). Toutefois, *octobre* possède un nombre d'occurrences relativement faible (10) qui pourrait lui valoir d'être écarté, notamment parce que cela réduit la fiabilité de la valeur de pGini. Le second est basé sur le n-gramme *satisfaisant* qui est reconnu par le système comme étant positif, mais ici le segment exprime visiblement une critique négative. Le n-gramme *non satisfaisant* ne remplissait pas les conditions nécessaires (en termes de nombre d'occurrences) à son intégration dans la liste des porteurs d'opinions négatives. Dans ces deux cas, le fait de présenter des segments entiers plutôt que juste des couples (sujet ; opinion) permet une meilleure compréhension (par exemple dans le cas où on s'en sert comme d'un résumé du document) : dans le premier segment pour comprendre que le contact était suffisant, dans le second car cela met en lumière la négation. Les deux derniers segments retournés sont cohérents, bien que recouvrants : la personne se plaint d'avoir à faire avec plusieurs contacts, *plusieurs* étant connu comme exprimant une opinion négative, justement parce qu'il se trouve souvent employé pour signaler le problème d'avoir eu plusieurs interlocuteurs, de devoir appeler plusieurs fois, etc.

document id : 8957 **catégorie :** posneg
– *le [contact du 8 **octobre**] être suffisant*, pGini : -0,82

– *plusieurs [contact successif non **satisfaisant**] car erreur*, pGini : +0,87

– *à [**plusieurs** contact] successif non*, pGini : -0,7

– *suite [**à plusieurs** contact] successif non*, pGini : -0,91

4.4 Évaluation

Dans cette section, nous présentons quelques résultats de catégorisations effectuées dans le but d'évaluer la pertinence des segments extraits par la méthode présentée dans ce chapitre. Il s'agit ici en fait simplement de voir jusqu'à quel point il est possible de retrouver l'opinion générale exprimée dans le document en se basant sur les n-grammes porteurs d'opinion présents dans les segments retournés. On catégorise donc à partir des seuls n-grammes du document qui se trouvaient dans le contexte d'un des sujets. Ceci permet d'avoir un aperçu du bien-fondé des segments, en s'affranchissant d'une évaluation manuelle qui consisterait à lire en parallèle les documents et les segments que nous proposons pour chacun d'entre eux et de les noter (une évaluation de ce type est toutefois envisagée et pourra être effectuée ultérieurement, quand la méthode aura été améliorée par les perspectives évoquées en fin de chapitre). Pour l'évaluation présentée ci-après, nous gardons bien à l'esprit que toutes les expressions d'opinion capturées dans une critique n'ont pas la même polarité. Dans un document positif, il se peut que l'on retrouve des opinions négatives, mais nous partons du pré-supposé qu'elles seront moins nombreuses que les positives ou moins fortes (*le film est excellent, bien que la qualité de l'image soit discutable* : dans l'idéal, excellent devrait faire pencher fortement vers le positif, alors que l'aspect négatif cité devrait être moins prégnant).

Pour les expériences suivantes basées sur des SVM, nous avons choisi, pour les raisons déjà évoquées en section 3.2, de ne considérer que le noyau linéaire, au travers de l'outil LibLinear. Les vecteurs d'entrée sont les lemmes du corpus avec leur fréquence dans le document à catégoriser. Les résultats sont évalués en termes de précision, rappel et Fscore moyens (macro-moyenne, voir formule 3.1).

Comme première approche, nous avons tenté de catégoriser chaque document du corpus de test (lui attribuer l'étiquette positif ou négatif) en utilisant l'information fournie par les segments. Nous avons choisi de considérer les valeurs des pGini des n-grammes porteurs d'opinion sur lesquels les segments retournés sont basés. C'est-à-dire, pour un document donné, si nous avons trouvé au moins un segment, nous sommons tous les scores pGini des n-grammes positifs d'un côté et les scores des n-grammes négatifs de l'autre. La somme la plus élevée nous donne la catégorie à attribuer. Les résultats pour le corpus de critiques de films sont présentés en tableau 4.3. Sur cette tâche, les SVM proposent 84% de bonne catégorisation. Les résultats pour le corpus d'enquête téléphonique sont présentés en tableau 4.4. Nous avons retiré les 84 documents étiquetés comme à la fois positifs et négatifs, étant donné que nous n'avons pas réussi à trouver de méthode permettant de les gérer correctement. Il nous reste donc 1343 entretiens à catégoriser en tant que positif ou négatif. Les SVM atteignent sur ce corpus 88% de bonne catégorisation (Fscore=0,88).

Notre seconde approche a consisté à combiner la catégorisation par pGini avec des SVM. Notre but était ici de définir un seuil de confiance pour les segments : si nous considérons que nous avons suffisamment d'informations en utilisant les segments (leurs n-grammes porteurs d'opinion), nous catégorisons avec ceux-ci, sinon nous nous reposons sur les SVM. Nous avons fixé le niveau de confiance en observant les résultats

Sujets	#docs tot	#docs segm	#docs corr	FS docs segm
manuels	600	444	335	78%
automatiques	600	568	442	78%

TABLE 4.3 – *corpus Movies Polarity Dataset v2.0 : résultats de catégorisation en utilisant uniquement les sommes des valeurs de pGini, pour les documents sur lesquels des segments ont pu être extraits.* ***Sujets*** *: méthode (automatique ou manuelle) utilisée pour extraire les sujets.* ***#docs tot*** *: nombre total de documents du corpus de test.* ***#docs segm*** *: nombre de documents sur lesquels au moins un segment a été extrait.* ***#docs corr*** *: nombre de documents correctement catégorisés en utilisant les segments.* ***FS docs segm*** *: FScore moyen calculé uniquement sur les documents pour lesquels au moins un segment a été extrait.*

Sujets	#docs tot	#docs segm	#docs corr	FS docs segm
manuels	1343	842	696	84%
automatiques	1343	1243	1053	85%

TABLE 4.4 – *corpus NPS07-09 : résultats de catégorisation en utilisant uniquement les sommes des valeurs de pGini, pour les documents sur lesquels des segments ont pu être extraits.* ***Sujets*** *: méthode (automatique ou manuelle) utilisée pour extraire les sujets.* ***#docs tot*** *: nombre total de documents du corpus de test.* ***#docs segm*** *: nombre de documents sur lesquels au moins un segment a été extrait.* ***#docs corr*** *: nombre de documents correctement catégorisés en utilisant les segments.* ***FS docs segm*** *: FScore moyen calculé uniquement sur les documents pour lesquels au moins un segment a été extrait.*

de catégorisation : le but était d'obtenir avec cette combinaison un Fscore, sur l'ensemble du corpus, supérieur ou égal à celui obtenu par les SVM seuls. Le niveau de confiance est défini par deux critères : le nombre de segments que nous avons réussi à extraire pour le document et le pouvoir relatif des sommes des valeurs de pGini. Nous avons, pour ce dernier point, additionné les pGini des n-grammes d'opinion pour chacune des catégories et divisé la plus élevée de ces sommes par le total des pGini du document.

Les résultats pour le corpus de critiques de films sont présentés en tableau 4.5. Concernant le niveau de confiance, nous avons choisi de catégoriser avec les pGini si nous avions extrait au moins 3 segments pour ce document et que le pouvoir relatif des valeurs de pGini était supérieur à 0,85 ; dans les autres cas, nous avons utilisé la sortie des SVM.

Les résultats pour le corpus d'enquêtes téléphoniques sont présentés en tableau 4.6. Le niveau de confiance a été fixé à un minimum de 5 segments et un pouvoir relatif des valeurs de pGini supérieur à 0,7.

Les résultats de catégorisation avec cette combinaison sont supérieurs ou égaux à ceux obtenus par les SVM seuls : pour les critiques de films on obtient 85% par la combinaison basée sur les sujets manuels et 84% par la combinaison basée sur les sujets automatiques contre 84% pour les SVM seuls ; pour le corpus d'enquête téléphonique, 88% (manuels) et 91% (automatiques) contre 88% pour les SVM seuls). Nous pouvons donc en déduire que, pour les documents situés au-dessus du seuil de confiance, la

Sujet	FS tot	#docs segm tot	#docs segm corr	Prec segm
manuels	85%	111	106	95%
automatiques	84%	201	181	90%

TABLE 4.5 – *corpus Movies Polarity Dataset v2.0 : résultats de catégorisation en combinant les pGini avec les SVM, pour l'ensemble du corpus de test.* ***Sujet*** *: méthode (automatique ou manuelle) utilisée pour extraire les sujets.* ***FS tot*** *: Fscore moyen sur l'ensemble du corpus de test.* ***#docs segm tot*** *: nombre de documents que le système tente de catégoriser en utilisant les valeurs de pGini (niveau de confiance élevé).* ***#docs segm corr*** *: nombre de documents qui ont été correctement catégorisés en utilisant les valeurs de pGini.* ***Prec segm*** *: Précision de la catégorisation par pGini sur les documents qui avaient un niveau de confiance élevé (#docs segm corr / #docs segm tot).*

Sujet	FS tot	#docs segm tot	#docs segm corr	Prec segm
manuels	88%	333	310	93%
automatiques	91%	527	509	97%

TABLE 4.6 – *corpus NPS07-09 : résultats de catégorisation en combinant les pGini avec les SVM, pour l'ensemble du corpus de test.* ***Sujet*** *: méthode (automatique ou manuelle) utilisée pour extraire les sujets.* ***FS tot*** *: Fscore moyen sur l'ensemble du corpus de test.* ***#docs segm tot*** *: nombre de documents que le système tente de catégoriser en utilisant les valeurs de pGini (niveau de confiance élevé).* ***#docs segm corr*** *: nombre de documents qui ont été correctement catégorisés en utilisant les valeurs de pGini.* ***Prec segm*** *: Précision de la catégorisation par pGini sur les documents qui avaient un niveau de confiance élevé (#docs segm corr / #docs segm tot).*

catégorisation avec les pGini est au minimum aussi bonne qu'avec les SVM. Nous en déduisons que, sur un échantillon réduit des documents, les segments extraits sont fiables. Par exemple, sur le corpus d'enquêtes téléphoniques, avec les sujets définis manuellement, la combinaison procure le même Fscore que les SVM seuls. 333 documents ont été catégorisés avec les pGini, parmi lesquels 310 l'ont été de manière correcte. La précision de cette catégorisation est ainsi de 93%. Nous pouvons ainsi considérer que les segments extraits sur ces 333 documents sont fiables. Ils peuvent alors être utilisés pour extraire de l'information sur les autres documents (ceux étiquetés comme posneg par exemple).

Bien sûr, évaluation manuelle (qui reste le meilleur moyen d'estimer la pertinence des segments retournés) permettrait de confirmer plus fortement la validité de l'approche proposée.

4.5 Discussion

La méthode que nous proposons ici est générique, ce qui lui permet de fonctionner dans plusieurs contextes : différents types de corpus (critiques, enquêtes, etc.), moyens d'expressions avec leurs syntaxes et vocabulaires spécifiques (entretiens transcrits, blogs, etc.), langues, ... Nous discutons dans cette section quelques caractéristiques de la méthode et comment modifier certaines parties afin de l'améliorer en l'adaptant pour des tâches spécifiques.

Concernant les sujets :

– dans les expériences présentées ci-dessus, les sujets sont uniquement des unigrammes car nous n'avions pas réellement besoin de considérer des bigrammes pour nos applications (nous avons seulement repéré quelques rares cas où nous aurions éventuellement pu prendre en compte des bigrammes, tels que "action scene", mais cela n'aurait pas eu un impact significatif - ici par exemple, "action scene" était capturé par "scene" qui lui était connu) ; toutefois, la méthode pourrait facilement être modifiée pour gérer des sujets multimots.
– pour certaines tâches, la liste des sujets pourrait être améliorée par l'utilisation de méta-données ou de connaissances externes ; par exemple, dans le cas des critiques de films, nous aurions pu ajouter une liste de noms propres (réalisateurs, acteurs, titres de films, ...), permettant de retourner aussi les segments dans lesquels les personnes sont explicitement nommées (dans le système tel qu'il est présenté, nous pouvons reconnaître les concepts dans "*an improbable plot , bad acting by the main* ***bad actors***" (critique 705_11973) mais pas dans "***mcconaughay is awful*** *, especially because he is trying to pull off his usual hollywood charm with a 3 week beard and torn jeans*" (critique 233_17614).

Concernant les n-grammes porteurs d'opinion :

– l'utilisation d'une source de connaissances externe (dictionnaire, WordNet, ...) devrait généralement conduire à une meilleure précision pour la sélection de termes subjectifs (notre méthode propose *part in the* par exemple), mais au détriment du rappel ; ainsi, une méthode entièrement probabiliste nous permet d'extraire différents types de mots, quelles que soient leurs catégories morpho-syntaxiques (adjectifs, verbes, ...), mais aussi des multimots ; cela permet aussi d'extraire des n-grammes qui ne sont, dans la plupart des cas, pas liés à des opinions, mais sont utilisés comme tels dans le domaine considéré : par exemple, pour le corpus d'enquêtes téléphoniques, nous avons trouvé des expressions temporelles, généralement associées à des opinions négatives, correspondant à des usagers se plaignant des délais ; cette méthode retourne aussi des expressions indirectes (négations, "should have been", ...).
– dans toutes nos expériences, nous avons utilisé un ensemble figé de paliers pour sélectionner les n-grammes ; ces paramètres pourraient être adaptés à la longueur des n-grammes : le filtre pourrait être moins sévère pour les trigrammes que pour les unigrammes.

Concernant les segments :

– nous avons choisi ici de présenter des segments étendus de 1 mot vers la gauche et 2 vers la droite : ceci pourrait être élargi en fonction de la tâche, du type de corpus, de sa taille et du temps que l'utilisateur accepte de consacrer à l'exploration des résultats : dans le cas d'une enquête où les réponses sont courtes, les éléments sont condensés et il n'est pas forcément nécessaire d'aller chercher très loin pour saisir le sens de ce qui est exprimé - si on est dans un style plus prolixe, ce qui est généralement le cas des commentaires d'internautes, il se peut que le complément d'information se trouve assez éloigné dans l'espace du segment initial.
– nous avons observé dans nos résultas un certain nombre de segments se recouvrant : cela se produit quand le système trouve plusieurs n-grammes porteurs

d'opinion dans le contexte d'un même sujet ou un tel n-gramme dans le contexte de plusieurs sujets ; il n'est, à notre avis, pas évident de créer une règle générale pour décider quel segment doit être conservé dans ce cas : parfois le plus court est le mieux (e.g., il y a une plus forte probabilité que le n-gramme soit relatif au sujet le plus proche, ou il apporte moins d'information inutile) et parfois le plus long est plus représentatif (son pGini est plus élevé, il est plus discriminant, ou il offre un meilleur aperçu de ce qui est discuté). Il y a d'ailleurs un parti à prendre entre le nombre d'occurrences du n-gramme, qui indique que l'élément est important puisque souvent avancé et le pGini qui indique l'appartenance à une des opinions (le nombre d'occurrences servant aussi à assurer la fiabilité du critère discriminant - dans le cas extrême, un n-gramme n'apparaissant que dans un seul document possède un pGini=1) ; ces deux facteurs entrant généralement en conflit : un n-gramme contenant un mot de moins (donc au minimum aussi fréquent) qu'un autre possède un pGini inférieur ou égal à ce second, est-il pour autant moins pertinent ? Est-ce que le mot supplémentaire est une négation et qu'on a donc extrait un avis et son contraire ?

Concernant l'ensemble du processus :

- nous avons ici choisi d'expérimenter une approche intégralement probabiliste, mais pour des objectifs plus ciblés (un contexte, langue, langage spécifique), l'ajout d'indices stylistiques pourrait permettre d'améliorer la méthode : par exemple, une analyse syntaxique des phrases pourrait résoudre les cas où un même n-gramme d'opinion apparaît dans le contexte de plusieurs sujets.
- plusieurs paramètres (seuils) sont utilisés tout au long du processus : nous avons ici considéré l'extraction de segments en tant qu'aide au résumé d'opinions et nous avons ensuite observé qu'ils pouvaient aussi être utilisés pour améliorer des résultats de catégorisation effectués par des SVM, mais peut-être que ces deux objectifs ne reposent pas sur le même ensemble de paramètres optimaux, ces derniers pourraient donc être définis distinctement pour ces deux tâches.

4.6 Conclusion

Dans ce chapitre, nous avons présenté une approche permettant d'extraire, dans un corpus d'opinions, des segments textuels dans lesquels une opinion est exprimée à propos de certains sujets. Nous avons testé l'approche à partir de deux moyens de définir ces sujets : manuellement ou automatiquement. La liste des sujets extraits automatiquement a ici été utilisée telle quelle, bien qu'elle nous semble imparfaite, mais pourrait aussi être vue comme une proposition à filtrer : il est plus rapide de parcourir les 200 premiers termes retournés par le Zscore que de lire trois fois plus de documents afin de savoir quels seront les éléments pertinents. Ces travaux sont à rapprocher de ceux de (Scaffidi et al., 2007) où les auteurs extraient des caractéristiques de produits en recherchant les noms et noms composés du corpus et en comparant, par une mesure basée sur la loi de Poisson, leurs fréquences par rapport à celles observées dans un corpus généraliste de langue anglaise de 100 millions de mots.

Ces segments peuvent permettre d ?améliorer légèrement les résultats d ?une catégorisation des textes en opinion, mais leur principal intérêt réside dans le fait qu ?ils offrent un aperçu rapide des opinions exprimées dans un corpus, ce qui est crucial pour une tâche de catégorisation d'opinions. Ils peuvent dans ce cas être utilisés de deux manières : soit, pour un des sujets (les costumes ou un des services que l'entreprise propose), extraire dans l'ensemble du corpus tous les segments dans lesquels une opinion est exprimée à son sujet (le sujet) et quelle est sa polarité - on peut ainsi obtenir l'opinion globale (numérique) sur le sujet et des exemples de critiques précises ; soit, document par document, extraire tous les segments possibles pour présenter un résumé des opinions exprimées sur les sujets que nous y retrouvons. Etant donné qu'un commentaire est rarement intégralement positif ou négatif, nous pouvons représenter des nuances, à la fois dans un segment puisque les n-grammes d'opinion que l'on recherche permettent de les prendre en compte, et à l'échelle du document en présentant aussi bien des segments positifs que des négatifs.

En fonction du but poursuivi, les seuils pour la recherche (tailles de fenêtres) et le filtrage des n-grammes peuvent être ajustés afin de fournir plus (on s'attendra à un meilleur rappel) ou moins (meilleure précision) de segments.

Les perspectives envisagées pour ce travail commencent par la recherche d'un moyen de prendre en compte les opinions exprimées de part et d'autre d'un sujet - un bout dans le contexte gauche, un bout dans le droit - comme dans la phrase : *this is the worst movie I have ever seen* voire *i have seen lots of bad actors, but james duval has got to be the worst*. Nous nous pencherons aussi sur le regroupement, la fusion ou la sélection de segments similaires (recouvrants) dans le but de permettre une meilleure compréhension des opinions exprimées. Par ailleurs, il est des cas où le sujet n'est pas cité explicitement ou pas dans le contexte proche : *for a schwarzenegger movie, this is not bad at all. in fact, it's very good.*, dans ce cas, même si on retrouve le "movie [...] not bad at all", il y a peu de chances que l'on parvienne à rattacher le "very good" à "movie". Cet écueil pourrait être contourné par la résolution d'anaphores (on retrouve quel était l'antécédent d'un pronom). Ce genre de travaux a été mené par (Jakob et Gurevych, 2010) avec des résultats mitigés. Pour une tâche semblable à la nôtre, les auteurs ont tenté d'améliorer leur extraction de couples (cible ; opinion) en testant deux systèmes différents de résolution d'anaphores : la première, principalement statistique (système MARS (Mitkov, 1998)), bien qu'utilisant quelques connaissances, a permis d'augmenter le rappel, mais au prix d'une perte de précision ; la seconde est basée sur le système CogNIAC (Baldwin, 1997), utilisant des règles, mais peu de connaissances (à l'exception des catégories morpho-syntaxiques), auxquelles on en a rajouté certaines du style "si on ne trouve pas d'antécédent proche, on remplace le pronom impersonnel (it) ou démonstratif (this) par movie ou film" ou "étant donné que dans les critiques, un même sujet est fréquemment critiqué par une série d'arguments, si il y a ambiguïté sur l'antécédent, on va chercher celui-de la phrase précédente sur lequel une opinion est exprimée", et permet d'augmenter à la fois le rappel et la précision. La première méthode n'améliorant pas l'extraction d'opinion (et utilisant tout de même quelques connaissances, même limitées) et la seconde étant basée sur des règles spécifiques, il nous semble difficile de les appliquer sans endommager la généricité de notre méthode. Cependant, cette idée

serait à creuser.

Enfin, nous pensons que la liste des sujets pourrait être améliorée à l'aide d'un processus itératif : en partant d'une liste de sujets réduite, extraire les n-grammes porteurs d'opinion, puis, utiliser ces n-grammes pour rechercher dans leur contexte de nouveaux sujets. Les travaux présentés dans (Qiu et al., 2011) exploitent ce principe. Les auteurs proposent une méthode pour extraire simultanément des mots porteurs d'opinion et des cibles sur lesquelles ils peuvent porter. Partant d'un premier ensemble de mots connus comme servant à exprimer une opinion, ils utilisent des relations de dépendances afin d'extraire des cibles (des caractéristiques des produits critiqués), qui serviront à extraire de nouveaux termes porteurs d'opinion, en fonctionnant par itérations qui s'arrêtent lorsque plus aucune cible ou opinion n'est trouvée. Nous voudrions pour notre part, continuer à utiliser des indices probabilistes, par exemple des mots qui ont un pGini faible (donc équirépartis dans les différentes opinions), avec une fréquence assez élevée dans le contexte d'un n-gramme porteur d'opinion et un IDF élevé (pour éviter les mots-outils).

Chapitre 5

Conclusion et perspectives

Dans cette thèse, nous avons travaillé sur l'extraction de chaînes de mots (segments textuels) relatives à des catégories (thématiques, rôles, opinions). Nous avons, dans un premier temps, proposé une méthode basée sur une métrique de recherche de collocations, que nous appliquons de manière distincte sur les documents liés à la même catégorie et qui, par itérations, nous permet d'obtenir des chaînes caractéristiques de cette catégorie. Ces chaînes sont alors employées pour améliorer les performances de systèmes de catégorisation de textes ou dans un but d'extraction de connaissances. Nous avons ensuite proposé une seconde méthode permettant de rechercher, dans un corpus d'opinions, des n-grammes exprimant des jugements sur des sujets prédéfinis. Nous pouvons alors extraire des segments textuels représentant l'expression d'une opinion sur un des sujets cibles. Nous avons aussi montré que les n-grammes d'opinion ainsi extraits permettaient d'améliorer quelque peu les résultats d'une catégorisation automatique de textes selon l'opinion globale qu'ils expriment.

Cependant, nous avons pu observer, pour une méthode comme pour l'autre, que les deux objectifs qu'elles poursuivaient (extraction de connaissance et amélioration de la catégorisation) n'étaient pas forcément soumis aux mêmes forces. Des paramètres entrent en effet en jeu dans la mise en application de ces systèmes (filtres de sélection des éléments pertinents). Or, les paramètres optimaux ne sont pas nécessairement les mêmes selon que l'on poursuive l'un ou l'autre de ces objectifs. Dans les expériences présentées dans cette thèse, nous avons proposé, pour chaque contexte applicatif, un jeu de paramètres, que nous avons cherché à optimiser selon un de ces aspects, généralement l'impact sur la catégorisation. Si les segments ainsi retournés nous ont semblé dans l'ensemble intéressants aussi pour l'aspect explicatif, il est possible que d'autres jeux de paramètres permettent d'améliorer encore la qualité des segments extraits (ceci restant difficilement évaluable).

D'une manière générale, les méthodes ici proposées font intervenir un certain nombre de paramètres, que nous avons pour l'heure définis empiriquement, mais nous envisageons pour des travaux ultérieurs de proposer une estimation de paramètres efficaces. Ceci pourrait par exemple être fait par inférence à partir de paramètres optimaux déjà trouvés pour certains corpus. Ainsi, en prenant en compte un ensemble de caracté-

ristiques des corpus (nombre de mots des documents, de catégories, répartition dans celles-ci, variété du vocabulaire, notamment discriminant, etc.), il pourrait être possible de déduire un jeu de paramètres permettant un traitement efficace d'un nouveau corpus. Il serait par ailleurs intéressant de travailler sur des méthodes de combinaison de paramètres optimaux trouvés sur plusieurs sous-ensembles dans un contexte de validation croisée.

D'un point de vue catégorisation, nous avons soulevé un certain nombre de pistes qui seraient à explorer. Il en va ainsi de l'application d'une catégorisation "hiérarchique" ou en plusieurs étapes, qui pourrait s'avérer pertinente entre autres pour certains des corpus sur lesquels nos méthodes ont été testés. Pour la catégorisation thématique appliquée aux conversations enregistrées en centre d'appels, il s'agirait par exemple de rechercher dans une première passe certaines catégories que le système est capable de reconnaître avec un faible taux d'erreur, puis de choisir ou non d'effectuer une deuxième passe selon que l'on a observé ou non à l'apprentissage une corrélation entre la première catégorie et une ou plusieurs autres. Dans le cadre d'une catégorisation en opinions incluant la neutralité, il pourrait s'agir de rechercher cette dernière dans un premier temps, puis si on considère que l'on a plutôt affaire à un avis tranché, de décider si celui-ci est positif ou négatif. De même, un apprentissage actif est à envisager, par exemple pour certaines tâches dans lesquelles on pourrait, après avoir catégorisé les documents les plus faciles (ceux pour lesquels la décision a été prise par le classifieur avec une forte confiance), recréer les modèles en y ajoutant ainsi quelques éléments attendus pour aider la catégorisation des documents les moins faciles. De même, et, notamment pour les corpus d'enquêtes ou d'entretiens d'entreprise, on pourrait introduire de la temporalité (des évènements dont on se met à parler, des nouveaux termes).

Les méthodes que nous avons développées l'ont été en veillant à rester le plus générique possible. Nous avons à ce titre pu évaluer leur robustesse (ou tout au moins leur constance) dans différents contextes. Dans le cas où l'on ne s'intéresserait qu'à un contexte restreint (toujours le même domaine, la même langue, la même spontanéité du discours), il est possible d'adapter ces modèles en y incluant la prise en compte de ses spécificités. Des exemples sont en ce sens proposés à la fin du dernier chapitre pour la deuxième approche que nous avons élaborée, mais d'autres encore sont envisageables, notamment la prise en compte de la position de la chaîne dans le document qui peut être intéressante pour certains types de résumés ou encore l'ajout de ressources externes ou de connaissances afin d'obtenir des segments directement reliés au domaine considéré.

Annexes

Annexe A

Vers la détection de nouveauté

Nous présentons ici quelques expériences préliminaires sur l'application de notre méthode d'extraction de chaînes dans un contexte de détection de la nouveauté. Ces expériences ont été réalisées sur un corpus non-annoté. Les chaînes extraites ne sont donc pas caractéristiques d'une catégorie. Elles sont calculées sur un premier corpus (correspondant aux données de la période N) en considérant tous les documents comme appartenant à la même catégorie. Puis nous effectuons le même calcul sur les données de la période N+1 et comparons les chaînes obtenues pour ces deux périodes.

A.1 Le corpus Laura

Les données sur lesquelles nous avons choisi de travailler sont issues des corpus produits par des échanges entre Laura, l'agent conversationnel d'aide d'EDF, et des internautes. Cette conseillère virtuelle, accessible au public sur le site d'EDF, a pour fonction de répondre aux questions que se posent les internautes (clients ou non) à propos d'offres, de services, de procédures, ... Le service se présente sous la forme d'un avatar proposant un champ où l'utilisateur peut poser sa question et un champ où la réponse lui est fournie. Les utilisateurs peuvent ou non s'être identifiés. Un identifiant de session est cependant attribué à chaque utilisateur, permettant de repérer les différents enchaînements de question-réponse effectués par cet internaute.

L'avantage de ces données est que l'on dispose d'un grand nombre d'échanges (environ 200 000 par mois, de janvier à juin 2010). De plus, ces données sont non-traitées et reflètent ainsi les particularités linguistiques des échanges en langue naturelle (orthographe, syntaxe, ...).

A.2 Approche générique

Nous avons dans un premier temps utilisé ces données de manière simpliste, pour tenter d'avoir un aperçu rapide de ce que pourraient donner nos algorithmes d'extraction de chaînes dans un contexte de détection de nouveauté. Les chaînes caractéristiques sont extraites à partir de l'ensemble des textes des questions d'un mois, chaque question posée par un internaute correspondant à un document. On obtient ainsi une liste de chaînes pour chacun des mois. La méthode la plus simple consiste ensuite à aller chercher quelles sont les chaînes présentes au mois N+1 qui ne l'étaient pas au mois N (en réalité, d'après le filtre d'extraction de collocations, une chaîne est considérée comme absente si elle est présente 7 fois ou moins).

Lors de l'étude des chaînes ainsi produites, nous avons observé que, bien que l'on y trouve de nombreuses références métier, le système extrait peu de chaînes exprimant réellement de la nouveauté, il s'agit généralement de chaînes que l'on aurait pu retrouver dans une des autres périodes, à l'exception de chaînes faisant intervenir des dates (par comparaison des chaînes présentes en février mais absentes en janvier, on trouve un certain nombre d'expressions contenant le mot "février") ou occasionnellement quelques évènements ("la tempête" en mars), mais celles-ci restant très rares, sachant que l'on extrait 20 000 chaînes par mois, dont entre 5 000 et 10 000 sont nouvelles par rapport au mois précédent. Les résultats ainsi retournés sont donc peu pertinents vis-à-vis du traitement manuel nécessaire afin de récupérer l'information intéressante. Parmi les principales causes de problèmes rencontrées, on trouve du bruit créé par les personnes posant plusieurs fois la même question d'affilée, qui font ainsi ressortir cette question comme si elle avait plus d'importance, ainsi que beaucoup de bruit dû à l'absence de pré-traitement des données.

A.3 Adaptation

Certains aspects de la méthode ont été modifiés afin d'adapter le processus à ce corpus spécifique et apporter une meilleure lisibilité des chaînes extraites.

En premier lieu, nous considérons dorénavant qu'un document est formé de l'ensemble des questions d'une conversation (plusieurs échanges de questions-réponses) et non plus une seule question. Ceci dans le but d'éviter que la même question posée plusieurs fois par la même personne ait trop d'impact. Nous utilisons pour cela les identifiants de sessions (différent de l'identifiant du client, il s'agit juste d'un numéro permettant de reconnaître que cette personne s'est déjà connectée récemment). On ne compte qu'une seule occurrence au maximum d'un mot par document.

Les mots ont été lemmatisés avec TreeTagger. Afin de gagner en qualité des chaînes extraites, nous avons appliqué un filtre sur les catégories morpho-syntaxiques des mots. Nous avons conservé les catégories TreeTagger : ABK, ABR, ADJ, ADV, INT, NAM, NOM, NUM, SENT, SYM, VER. C'est-à-dire que nous avons retiré les articles, pronoms, ponctuations, conjonctions et prépositions. Dans le cas où TreeTagger propose plusieurs

lemmatisations pour un même mot, nous choisissons arbitrairement la dernière. Pour pallier quelques erreurs de TreeTagger (mauvais étiquetage morpho-syntaxique), nous avons ajouté un anti-dictionnaire (stopliste) contant les mots : l, m, j, d, s, le, la, les, un, une, des, ma, mon, mes, du, de, t.

Enfin, nous proposons maintenant en sortie non plus les chaînes présentes un mois et absentes le précédent, mais le pourcentage d'augmentation de leurs occurrences d'un mois à l'autre.

On trouve en premier lieu dans ces listes, des expressions se rapportant à la date actuelle, par exemple pour les nouveautés entre février et janvier, les premières chaînes (celles ayant le plus fort pourcentage de progression) sont :
02/-2010 350 contre 54 occurrences,
02/-10 108 contre 17 occurrences,
02/-2010. 47 contre 8 occurrences,
février-2010 228 contre 48 occurrences.

On trouve ensuite assez rapidement des expressions métier :
combien-jour-rouge-rester 27 contre 10 occurrences,
dehors-période-relevé 35 contre 13 occurrences,
acceder-avoir-espace-client-. 22 contre 9 occurrences (vraisemblablement une mauvaise lemmatisation de l'expression "*accéder à mon espace client*")
service-être-indisponible 12 contre (?<8) occurrences)
qui signifie que la chaîne est apparue 12 fois en février et moins de 8 fois en janvier (peut-être zéro).

Dans ce cas de figure, le nombre de documents est réduit, et surtout le nombre de mots pris en compte ; le nombre de chaînes extraites est ainsi moins élevé (entre 4 000 et 5 000 sur un mois). Ceci permet par ailleurs de réduire le temps d'exécution (il y a moins de mots à considérer), étant donné que le corpus est assez fourni (200 000 questions par mois).

Bien entendu, ceci n'est qu'une première approche, qui nécessite encore à l'heure actuelle au moins un filtrage manuel des chaînes retournées. Toutefois, celles-ci peuvent servir d'indication pour donner un aperçu, par exemple en ne regardant que les quelques premières centaines de chaînes qui ont la plus forte progression d'une période à une autre (ici la période est d'un mois), ce qui peut être étendu afin de voir celles qui deviennent plus ou moins fréquentes d'un mois sur l'autre pendant une plus longue période (suivre l'évolution d'un concept dans le temps).

Annexe B

Plate-forme OSS - Valorisation industrielle

Une partie de nos travaux a été intégrée dans un outil utilisé actuellement par les analystes fouille de textes de la Direction Commerce d'EDF.

OSS (Outils Statistiques pour la Sémantique) est un logiciel développé par EDF pour l'analyse de corpus textuels. Il permet d'effectuer une exploration des données au travers de plusieurs outils. Son but premier est l'aide à la création de cartouches sémantiques utilisées par l'outil Luxid développé par la société TEMIS [1]. Luxid est un outil de fouille de textes (extraction de connaissances : concepts métiers et opinions) au travers de cartouches sémantiques (Kuznick et al., 2010), correspondant à des patrons linguistiques liés aux concepts métier. OSS prend ainsi un corpus en entrée et permet d'y appliquer plusieurs outils, facilitant le travail des experts chargés d'écrire les patrons linguistiques (ceux-ci étant à l'heure actuelle définis après une lecture approfondie du corpus sur lequel on souhaitera appliquer des traitements). Ainsi, il offre les fonctionnalités suivantes :

- la visualisation de groupes de mots issus d'un co-clustering (documents/mots).
- la recherche de chaînes caractéristiques des catégories par la méthode présentée en section 2.2 dans le cas où les catégories sont déjà connues ; dans le cas où il s'agit d'une première exploration (nouveau corpus) et qu'il n'y a pas de catégories définies, la méthode est appliquée sur l'ensemble du corpus (en considérant tous les documents comme rattachés à la même catégorie) offrant ainsi la possibilité de visualiser les chaînes caractéristiques de cette unique catégorie.
- l'extraction des syntagmes nominaux présents dans le corpus ; par rapport à notre méthode, celle-ci ne gère que l'extraction de syntagmes nominaux (alors que nos chaînes se contentent de regrouper des mots qui ont de bonnes raisons statistiques de l'être, pouvant rassembler des sujets avec leurs verbes, ...), ceux-ci étant plus "propres" (syntaxiquement corrects).

Une capture d'écran du logiciel OSS est proposée en figure B.1.

1. http ://www.temis.com/

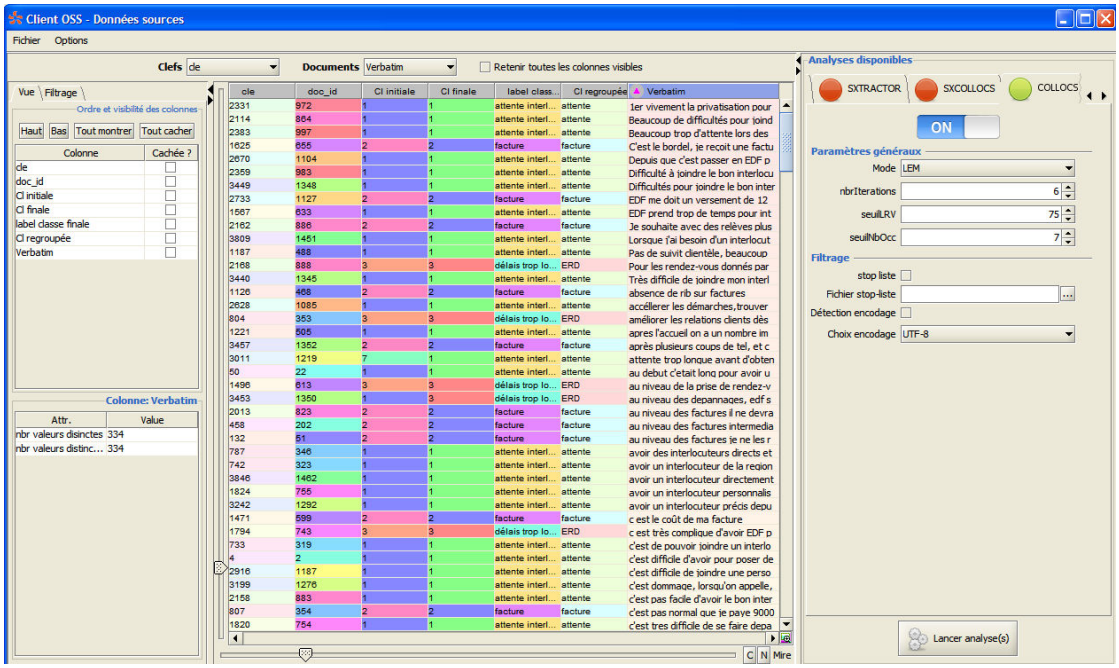

FIGURE *B.1* – *Interface de l'outil OSS : Au centre, le corpus au format tabulé (un document par ligne, avec les différents champs possibles : identifiant du document, catégorie, texte, ...), à gauche, les options de filtrage (sélectionner seulement certains champs), à droite, les différents outils que l'on peut appliquer, ici positionné sur l'extraction de chaînes telle que nous la proposons avec les différentes options paramétrables (filtre pour la pertinence des agglutinations, utilisation des formes fléchies ou lemmatisées, anti-dictionnaire ou pas, ...).*

Annexe C

Quelques autres méthodes pour l'extraction de chaînes

En expériences préliminaires, nous avons testé d'autres méthodes d'extraction de chaînes, basées sur des métriques de recherche de collocations différentes de celle présentée dans ce manuscrit. Le processus global est le même que celui présenté en chapitre 2 : on applique une métrique pour trouver des collocations que l'on agglutine et on recherche des chaînes de plus grande taille par itérations. Les métriques présentées ci-dessous ont toutefois fourni des résultats moins intéressants que le LRV par catégorie du chapitre 2, aussi bien en termes d'intérêt visuel des chaînes extraites, que d'impact sur la catégorisation.

C.1 Logarithme du Rapport de Vraisemblance sans distinction de catégorie

Nous avons utilisé dans ce cas le calcul du LRV - selon la formule 2.1 présentée au chapitre 2 - en considérant de manière indifférenciée l'ensemble des documents du corpus d'apprentissage (toutes catégories confondues). Cette méthode extrait donc des collocations, puis des chaînes, caractéristiques du corpus mais pas des catégories qui le composent. Bien entendu, l'intérêt visuel de ces chaînes est moindre ou tout au moins différent : on ne peut pas avoir un aperçu d'éléments représentatifs de chacune des catégories, mais cela peut toujours servir en tant qu'exploration du corpus. D'un point de vue catégorisation automatique, le gain obtenu par l'utilisation de ces chaînes globales s'est révélé plus faible.

C.2 Critère discriminant

Basée sur le principe de l'information mutuelle, la formule C.1 permet de rechercher des couples de mots propres à une des classes en comparant le nombre d'occurrences du couple dans la catégorie (classe majoritaire) dans laquelle il apparaît le plus $C_{CM}(m1m2)$ avec le nombre d'occurrences de ce couple dans l'intégralité du corpus $C_T(m1m2)$, que l'on met en rapport avec les nombres d'occurrences des mots pris isolément, dans la classe majoritaire $C_{CM}(m1)$ et $C_{CM}(m2)$ et dans l'intégralité du corpus $C_T(m1)$ et $C_T(m2)$.

$$ratio = \frac{\frac{C_{CM}(m1m2)}{\sqrt{C_{CM}(m1) \times C_{CM}(m2)}}}{\frac{0{,}5 + C_T(m1m2) - C_{CM}(m1m2)}{\sqrt{(0{,}1 + C_T(m1) - C_{CM}(m1)) \times (0{,}1 + C_T(m2) - C_{CM}(m2))}}} \tag{C.1}$$

Avec :
$C_T(m2)$ le nombre d'occurrences de $m2$ dans l'intégralité du corpus
$C_T(m1m2)$ le nombre d'occurrences du bigramme $m1m2$ dans l'intégralité du corpus
$C_{CM}(m2)$ le nombre d'occurrences de $m2$ dans la classe dans laquelle il apparaît le plus
$C_{CM}(m1m2)$ le nombre d'occurrences du bigramme $m1m2$ dans la classe dans laquelle il apparaît le plus

Nous obtenons ainsi pour chaque couple un score de représentativité pour la catégorie dans laquelle il apparaît le plus. Nous choisissons ensuite de conserver à chaque itération toutes les propositions ayant obtenu le score maximal (d'autres expériences ont été réalisées en conservant à chaque fois les propositions ayant les n scores les plus élevés).

C.3 Critère réfutant

L'idée est très proche de celle présentée à la section précédente, à la différence près qu'on ne cherche pas ici des collocations propres à une des catégories, mais des couples qui n'appartiennent pas à une certaine catégorie (pour une d'entre elles et une seule, la probabilité de trouver cette collocation est quasiment nulle). Si l'impact sur le système de catégorisation s'est révélé assez intéressant (proche de celui observé pour le LRV par catégorie, quoi que légèrement inférieur), l'intérêt dans la visualisation des chaînes est lui plus restreint : on extrait des chaînes qui n'apparaissent jamais dans une des catégories alors qu'on les trouve dans les documents des autres catégories. Il peut toutefois y avoir certaines applications pour lesquelles cela peut être utile, par exemple dans un cas de détection d'opinion, on peut retrouver une référence à un produit ou service qui n'est jamais cité dans la catégorie positive alors qu'on le trouve dans les autres, ce qui est plutôt mauvais signe.

Liste des illustrations

Liste des tableaux

Bibliographie

(Abbasi et al., 2008) A. Abbasi, H.-H. Chen, et A. Salem, 2008. Sentiment analysis in multiple languages : Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems 26*.

(Androutsopoulos et al., 2000) I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, et C. D. Spyropoulos, 2000. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. Dans les actes de *SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, Athènes, 160–167.

(Baldwin, 1997) B. Baldwin, 1997. Cogniac : High precision coreference with limited knowledge and linguistic resources. Dans les actes de *Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, Madrid, Spain, 38–45.

(Banerjee et Rudnicky, 2006) S. Banerjee et A. Rudnicky, 2006. You are what you say : using meeting participantsÕ speech to detect their roles and expertise. Dans les actes de *HLT-NAACL Workshop on Analyzing Conversations in Text and Speech*, 23–30.

(Barzilay et al., 2000) R. Barzilay, M. Collins, J. Hirschberg, et S. Whittaker, 2000. The rules behind roles : Identifying speaker role in radio broadcasts. Dans les actes de *The 17th National Conference on ArtiÞcial Intelligence (AAAI-2000)*, Austin, 679–684.

(Bigot et al., 2010) B. Bigot, I. Ferrané, J. Pinquier, et R. André-Obrecht, 2010. Detecting individual role using features extracted from speaker diarization results. *Multimedia Tools and Applications 11*, 1–23.

(Bossard et al., 2008) A. Bossard, M. Généreux, et T. Poibeau, 2008. Description of the lipn systems at tac 2008 : Summarizing information and opinions. Dans les actes de *Notebook Papers and Results, Text Analysis Conference (TAC-2008)*, Gaithersburg, Maryland, USA.

(Bourigault, 2002) D. Bourigault, 2002. Upery : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. Dans les actes de *TALN 2006*, Nancy, 75–84.

(Bozzi et al., 2009) L. Bozzi, P. Suignard, et C. Waast-Richard, 2009. Segmentation et classification non supervisée de conversations téléphoniques automatiquement retranscrites. Dans les actes de *TALN 2009*, Senlis, France.

(Breck et al., 2007) E. Breck, Y. Choi, et C. Cardie, 2007. Identifying expressions of opinion in context. Dans les actes de *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India.

(Cailliau et Giraudel, 2008) F. Cailliau et A. Giraudel, 2008. Enhanced search and navigation on conversational speech. Dans les actes de *SIGIR 2008 SSCS Workshop*, Singapore.

(Cailliau et Poudat, 2008) F. Cailliau et C. Poudat, 2008. Caractérisation lexicale des contributions clients agents dans un corpus de conversations téléphoniques retranscrites. Dans les actes de *JADT 2008 : 9es Journées internationales d'Analyse statistique des Données Textuelles*, 267–275.

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

(Cancedda et al., 2003) N. Cancedda, E. Gaussier, C. Goutte, et J. M. Renders, 2003. Word sequence kernels. *The Journal of Machine Learning Research 3*, 1059–1082.

(Clarke et al., 2009) C. L. A. Clarke, N. Craswell, et I. Soboroff, 2009. Overview of the trec 2009 web track. Dans les actes de *The 18th Text REtrieval Conference*, Gaithersburg, Maryland.

(Clemens et al., 2009) C. Clemens, S. Feldes, K. Schuhmacher, et J. Stegmann, 2009. Automatic topic detection of recorded voice messages. Dans les actes de *Interspeech 2009*, Brighton, 872–875.

(Cusin-Berche, 2003) F. Cusin-Berche, 2003. *Les mots et leurs contextes*, 18. Presses Sorbonne Nouvelles.

(Daille, 1996) B. Daille, 1996. Study and implementation of combined techniques for automatic extraction of terminology. *The Balancing Act : Combining Symbolic and Statistical Approaches to Language 1*, 49–66.

(Damerau, 1993) F. Damerau, 1993. Generating and evaluating domain-oriented multiword terms from texts. *Information Processing and Management 29*(4), 433–447.

(Danesi et Clavel, 2010a) C. Danesi et C. Clavel, 2010a. Impact of spontaneous speech features on business concept detection : a study of call-centre data. Dans les actes de *Workshop Searching Spontaneous Conversational Speech ACM Multimedia 2010*.

(Danesi et Clavel, 2010b) C. Danesi et C. Clavel, 2010b. Impact of spontaneous speech features on business concept detection : a study of call-centre data. Dans les actes de *The 2010 international workshop on Searching Spontaneous Conversational Speech*, SSCS '10, New York, NY, USA, 11–14. ACM.

(Dave et al., 2003) K. Dave, S. Lawrence, et D. M. Pennock, 2003. Mining the peanut gallery : Opinion extraction and semantic classification of product reviews. Dans les actes de *WWW-2003*, Budapest.

(Dias et al., 2003) G. Dias, L. Carapinha, R. Trindade, S. Mota, et J. Dias, 2003. Construire et accéder à une base de données d'expressions figées à partir des ressources de la toile. Dans les actes de *TIA-2003*, 92–101.

(Dias et Vintar, 2005) G. Dias et S. Vintar, 2005. Unsupervised learning of multiword units from part-of-speech tagged corpora : Does quantity mean quality ? Dans C. Bento, A. Cardoso, et G. Dias (Eds.), *Progress in Artificial Intelligence*, Volume 3808 de *Lecture Notes in Computer Science*, 669–679. Springer Berlin / Heidelberg.

(Drouin, 2004) P. Drouin, 2004. Spécificités lexicales et acquisition de la terminologie. Dans les actes de *JADT*, 345–352.

(Dunning, 1993) T. Dunning, 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics 19*, 61–74.

(Esuli et Sebastiani, 2006) A. Esuli et F. Sebastiani, 2006. Sentiwordnet : A publicly available lexical resource for opinion mining. Dans les actes de *Language Resources and Evaluation Conference*, Gênes, Italie, 417–422.

(Fan et al., 2008) R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, et C.-J. Lin, 2008. Liblinear : A library for large linear classification. *The Journal of Machine Learning Research 9*, 1871–1874.

(Frantzi et al., 2000) K. Frantzi, S. Ananiadou, et H. Mima, 2000. Automatic recognition of multi-word terms :. the c-value/nc-value method. *International Journal on Digital Libraries 3*, 115–130.

(Garnier-Rizet et al., 2008) M. Garnier-Rizet, G. Adda, F. Cailliau, J.-L. Gauvain, S. Guillemin-Lanne, L. Lamel, S. Vanni, et C. Waast-Richard, 2008. Callsurf - automatic transcription, indexing and structuration of call center conversational speech for knowledge extraction and query by content. Dans les actes de *LREC 2008*, Marrakech, Morocco.

(Gordon V. Cormack, 201) C. L. A. C. Gordon V. Cormack, Mark D. Smucker, 201. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*.

(Grouin et al., 2007) C. Grouin, J.-B. Berthelin, S. E. Ayari, T. Heitz, M. Hurault-Plantet, M. Jardino, Z. Khalis, et M. Lastes, 2007. Présentation de deft 07 (defi fouille de textes). Dans les actes de *L'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, 1–8.

(Grouin et al., 2008) C. Grouin, J.-B. Berthelin, S. E. Ayari, M. Hurault-Plantet, et S. Loiseau, 2008. Présentation de deft'08 (défi fouille de textes). Dans les actes de *L'Atelier DEFT 2008*, Avignon.

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

(Harris, 1969) Z. S. Harris, 1969. Analyse du discours. *Langages 4*, 8–45.

(Hsu et al., 2003) C.-W. Hsu, C. chung Chang, et C. jen Lin, 2003. A practical guide to support vector classification. Rapport technique, Department Of Computer Science.

(Jakob et Gurevych, 2010) N. Jakob et I. Gurevych, 2010. Using anaphora resolution to improve opinion target identification in movie reviews. Dans les actes de *The ACL 2010 Conference Short Papers, ACLShort Õ10*, Stroudsburg, USA, 263–268.

(Jin et al., 2009) W. Jin, H. H. Ho, et R. K. Srihari, 2009. Opinionminer : A novel machine learning system for web opinion mining and extraction. Dans les actes de *The 15th ACM SIGKDD*.

(Joachims, 1998) T. Joachims, 1998. Text categorization with support vector machines : Learning with many relevant features. *Machine Learning : ECML-98 1398/1998*, 137–142.

(Kessler et al., 2008) R. Kessler, J. M. Torres-Moreno, et M. El-Bèze, 2008. E-gen : Profilage automatique de candidatures. Dans les actes de *Traitement Automatique de la Langue Naturelle (TALN 2008)*, Avignon.

(Kleedorfer et Seewald, 2005) F. Kleedorfer et A. Seewald, 2005. Implementation of a string kernel for weka. Rapport technique TR-2005-13, Oesterreichisches Forschungsinstitut fuer Artificial Intelligence, Wien, Austria.

(Kuznick et al., 2010) L. Kuznick, A.-L. Guènet, A. Peradotto, et C. Clavel, 2010. L'apport des concepts métiers pour la classification des questions ouvertes d'enquête. Dans les actes de *TALN*.

(Lin, 1998) D. Lin, 1998. Automatic retrieval and clustering of similar words. Dans les actes de *17th International Conference on Computational Linguistics*, 768–774.

(Liu, 2010) B. Liu, 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*.

(Liu, 2006) Y. Liu, 2006. Initial study on automatic identiÞcation of speaker role in broadcast news speech. Dans les actes de *HLT-NAACL 2006*, 81–84.

(Lodhi et al., 2002) H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, et C. J. C. H. Watkins, 2002. Text classification using string kernels. *Journal of Machine Learning Research 2*, 419–444.

(Manning et Schütze, 2000) C. D. Manning et H. Schütze, 2000. *Foundations of statistical natural language processing*, 151–189. MIT Press.

(Mitkov, 1998) R. Mitkov, 1998. Robust pronoun resolution with limited knowledge. Dans les actes de *The 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Canada, 869–875.

(Moschitti et Basili, 2004) A. Moschitti et R. Basili, 2004. Complex linguistic features for text classification : A comprehensive study. *Lecture Notes in Computer Science 2997*, 181–196.

(Nallapati, 2004) R. Nallapati, 2004. Discriminative models for information retrieval. Dans les actes de *SIGIR*, 64–71.

(Ng et al., 2006) V. Ng, S. Dasgupta, et S. M. N. Arifin, 2006. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. Dans les actes de *COLING/ACL*, Sydney, 611–618.

(Ounis et al., 2007) I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, et I. Soboroff, 2007. Overview of the trec 2006 blog track. Dans les actes de *TREC 2006*, Gaithersburg, USA.

(Pang et Lee, 2004) B. Pang et L. Lee, 2004. A sentimental education : Sentiment analysis using subjectivity summarization based on minimum cuts. Dans les actes de *ACL*, 271–278.

(Pang et Lee, 2005) B. Pang et L. Lee, 2005. Seeing stars : Exploiting class relationships for sentiment categorization with respect to rating scales. Dans les actes de *ACL*, 115–124.

(Pang et al., 2002) B. Pang, L. Lee, et S. Vaithyanathan, 2002. Thumbs up ? Sentiment classification using machine learning techniques. Dans les actes de *The 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 79–86.

(Paroubek et al., 2007) P. Paroubek, J.-B. Berthelin, S. E. Ayari, C. Grouin, T. Heitz, M. Hurault-Plantet, M. Jardino, Z. Khalis, et M. Lastes, 2007. Résultats de l'édition 2007 du defi fouille de textes. Dans les actes de *L'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, 9–17.

(Patin, 2010) G. Patin, 2010. Unsupervised chinese lexicon extraction on a domain specific corpus : Method and evaluation. Dans les actes de *The 23rd International Conference on Computational Linguistics (COLING 2010)*.

(Pearce, 2002) D. Pearce, 2002. A comparative evaluation of collocation extraction techniques. Dans les actes de *Conference on language Resources and Evaluation (LREC)*, Las Palmas, Canary Islands, 1530–1536.

(Pevzner et Hearst, 2002) L. Pevzner et M. A. Hearst, 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics 28*(1), 19–36.

(Platt, 1999) J. C. Platt, 1999. Fast training of support vector machines using sequential minimal optimization. *1*, 185–208.

(Popescu et Etzioni, 2005) A.-M. Popescu et O. Etzioni, 2005. Extracting product features and opinions from reviews. Dans les actes de *The Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.

(Qiu et al., 2011) G. Qiu, B. Liu, J. Bu, et C. Chen, 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics 37*, 9–27.

(Riloff et Wiebe, 2003) E. Riloff et J. Wiebe, 2003. Learning extraction patterns for subjective expressions. Dans les actes de *The Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, 105,112.

(Saggion et Funk, 2009) H. Saggion et A. Funk, 2009. Extracting opinions and facts for business intelligence. *Revue des Nouvelles Technologies de l'Information - E-17*, 119–146.

(Salamin et al., 2009) H. Salamin, S. Favre, et A. Vinciarelli, 2009. Automatic role recognition in multiparty recordings : Using social affiliation networks for feature extraction. *IEEE Transactions on Multimedia 11*, 1373–1380.

(Salton et Buckley, 1988) G. Salton et C. Buckley, 1988. Term weighting approaches in automatic text retrieval. *Information Processing and Management 24(5)*, 513–523.

(Savoy, 2009) J. Savoy, 2009. Indexation et représentation comparative : Application au discours électoral. Dans les actes de *CORIA'09*, Toulon, 185–200.

(Savoy, 2011) J. Savoy, 2011. Quel est l'auteur de ce roman ? Dans les actes de *CORIA'11*, Avignon.

(Scaffidi et al., 2007) C. Scaffidi, K. Bierhoff, E. Chang, M. Felker, et H. N. C. Jin, 2007. Red opal : product-feature scoring from reviews. Dans les actes de *The 8th ACM Conference on Electronic Commerce*, San Diego, USA, 182–191.

(Schapire et Singer, 2000) R. E. Schapire et Y. Singer, 2000. Boostexter : a boosting-based system for text categorization. *Machine Learning 39*(2), 135–168.

(Schmid, 1994) H. Schmid, 1994. Probabilistic part-of-speech tagging using decision trees. Dans les actes de *International Conference on New Methods in Language Processing*, Manchester, 44–49.

(Sebastiani, 2002) F. Sebastiani, 2002. Machine learning in automated text categorization. *ACM Computing Surveys 34(1)*, 1–47.

(Seretan et al., 2004) V. Seretan, L. Nerima, et E. Wehrli, 2004. Using the web as a corpus for the syntactic-based collocation identification. Dans les actes de *International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, 1871–1874.

(Sitbon et Bellot, 2007) L. Sitbon et P. Bellot, 2007. Topic segmentation using weighted lexical links (wll). Dans les actes de *SIGIR*, Amsterdam, Pays Bas, 737–738.

(Smadja, 1993) F. Smadja, 1993. Retrieving collocations from text : Xtract. *Computational Linguistics 19*(1), 143–178.

(Smadja et McKeown, 1990) F. A. Smadja et K. R. McKeown, 1990. Automatically extracting and representing collocations for language generation. Dans les actes de *The 28th annual meeting on Association for Computational Linguistics*, 252–259.

(Somprasertsri et Lalitrojwong, 2010) G. Somprasertsri et P. Lalitrojwong, 2010. Mining feature-opinion in online customer reviews for opinion summarization. *Journal of Universal Computer Science 16*, 938–955.

(Spriet et El-Bèze, 1999) T. Spriet et M. El-Bèze, 1999. Introduction of rules into a stochastic approach for language modelling. *NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES 169*, 350–355.

(Strohman et al., 2005) T. Strohman, D. Metzler, H. Turtle, et W. B. Croft, 2005. Indri : A language model-based search engine for complex queries. Dans les actes de *IA Õ05 : The International Conference on Intelligence Analysis*.

(Thanopoulos et al., 2002) A. Thanopoulos, N. Fakotakis, et G. Kokkinakis, 2002. Comparative evaluation of collocation extraction metrics. Dans les actes de *Language Resource and Evaluation (LREC)*, Las Palmas, Canary Islands, 620–625.

(Torres-Moreno et al., 2007) J.-M. Torres-Moreno, M. El-Bèze, F. Béchet, et N. Camelin, 2007. Comment faire pour que l'opinion forgée à la sortie des urnes soit la bonne ? application au défi deft 2007. Dans les actes de *L'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, 119–133.

(Turney, 2002) P. D. Turney, 2002. Thumbs up or thumbs down ? semantic orientation applied to unsupervised classification of reviews. Dans les actes de *The 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, 417–424.

(Vinciarelli, 2007) A. Vinciarelli, 2007. Speakers role recognition in multiparty audio recordings using social network analysis and duration distribution modeling. *IEEE Transactions on Multimedia 9*, 1215–1226.

(Viterbi, 1967) A. Viterbi, 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory 13*, 260–269.

(Wiebe et al., 2001) J. Wiebe, T. Wilson, et M. Bell, 2001. Identifying collocations for recognizing opinions. Dans les actes de *The ACL/EACL Workshop on Collocation*, Toulouse, France.

(Wiegand et Klakow, 2012) M. Wiegand et D. Klakow, 2012. Generalization methods for in-domain and cross-domain opinion holder extraction. Dans les actes de *EACL2012*, Avignon, 325–335.

(Wilson et al., 2005) T. Wilson, J. Wiebe, et P. Hoffmann, 2005. Recognizing contextual polarity in phrase-level sentiment analysis. Dans les actes de *The 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, Vancouver, 417–424.

(Wu et al., 2009) Y. Wu, Q. Zhang, X. Huang, et L. Wu, 2009. Phrase dependency parsing for opinion mining. Dans les actes de *The 2009 Conference on Empirical Methods in Natural Language Processing*.

(Yang et Liu, 1999) Y. Yang et X. Liu, 1999. A re-examination of text categorization methods. Dans les actes de *The 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, Berkeley, California, USA, 42–49.

(Yu et al., 2003) J. Yu, Z. Jin, et Z. Wen, 2003. Automatic detection of collocation. Dans les actes de *The 4th Chinese lexical semantics workshop*, Hong-Cong.

(Zhuang et al., 2006) L. Zhuang, F. Jing, et X.-Y. Zhu, 2006. Movie review mining and summarization. Dans les actes de *The ACM SIGIR Conference on Information and Knowledge Management (CIKM)*.

Bibliographie personelle

R. Lavalley, C. Clavel, P. Bellot
Extraction probabiliste de chaînes de mots relatives à une opinion
Traitement Automatique des Langues (TAL), p. 101 à 130, vol. 51-3, 2011

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

R. Lavalley, C. Clavel, P. Bellot, M. El Bèze
Combining text categorization and dialog modeling for speaker role identification on call center conversations
Dans les actes de Interspeech 2010, Japon

R. Lavalley, P. Bellot, M. El Bèze
Interactions entre le calcul de collocations et la catégorisation automatique de textes
Dans les actes de 6è Conférence en Recherche d'Informations et Applications (CORIA 2009), France

E. Charton, N. Camelin, R. Acuna-Agost, P. Gotab, R. Lavalley, R. Kessler et S. Fernandez
Pré-traitements classiques ou par analyse distributionnelle : application aux méthodes de classification automatique déployées pour DEFT08
Actes de l'atelier Défi Fouille de Textes, 2008, France