Université de Montréal

Résumé automatique des commentaires de consommateurs

par
Olga Feiguina

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de M.Sc.
en informatique

Mai 2006

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Résumé automatique des commentaires de consommateurs

présenté par
Olga Feiguina

a été évalué par un jury composé des personnes suivantes :

Philippe Langlais
président-rapporteur

Guy Lapalme
directeur de recherche

Claude Frasson
membre du jury

Résumé

Ce mémoire présente notre travail dans le domaine de la fouille d'opinions pour le traitement automatique de commentaires de consommateurs. Nous proposons une architecture d'application de résumé automatique d'une collection de commentaires à partir d'une requête qui utilise une ontologie du domaine comme ressource principale. Un prototype de cette application est décrit et un exemple de fonctionnement est présenté. Ceci éclaire la nécessité de résumer les commentaires de consommateurs et le grand potentiel de notre architecture pour l'accomplir.

L'ontologie, qui décrit les produits commentés et d'autres entités pertinentes, est créée semi-automatiquement. Nous exposons une nouvelle méthodologie d'extraction automatique des caractéristiques des produits qui utilise un extracteur de terminologie qui n'exige qu'une analyse superficielle du texte. L'évaluation dans le domaine spécifique de l'électronique donne une bonne précision d'environ 80%. Ce résultat est difficilement comparable avec les méthodes précedentes d'extraction de caractéristiques à cause de différences dans la définition de la tâche.

Ensuite nous présentons notre tentative de regroupement des caractéristiques extraites, entre 400 et 800 pour les produits dans notre corpus de commentaires en anglais. Nous avons trouvé que les co-occurrences de caractéristiques ne sont pas utiles pour la découverte des relations sémantiques entre les caractéristiques extraites. Les possibilités de regroupement à base de similarité sémantique en utilisant la ressource lexicale WordNet sont plus prometteuses. Cette expérience comprend le développement et l'évaluation d'un algorithme de désambiguïsation de sens dans un domaine spécifique. La précision obtenue est d'environ 53%, ce qui est comparable à celle d'autres méthodes de désambiguïsation de sens dans un domaine particulier.

Pour étudier la possibilité de baser notre résumé sur une question en langue naturelle au lieu d'une requête, nous avons aussi travaillé sur la classification de questions en fonction du type de réponse. Nous présentons notre expérience avec la classification des questions factuelles à base des n-grammes en utilisant plusieurs algorithmes d'apprentissage machine. Les meilleures précisions obtenues sont environ 80% pour l'anglais et le français ; en utilisant AdaBoost avec les arbres de décision.

Mots-clés : informatique, intelligence artificielle, traitement des langues naturelles, résumé automatique, analyse des émotions

Abstract

In this thesis, we present our work in the domain of opinion mining, in which we focus on automatic processing of customer reviews. We propose the architecture of an application that summarises automatically a collection of customer reviews based on a query using an ontology as its main resource. A prototype of this application is described and its functioning is explained using an example. This elucidates the need to summarise customer reviews and the real potential of our architecture to do so.

The ontology, which describes products commented in the reviews as well as other relevant entities, was created semi-automatically. We present a novel method of automatic product feature extraction that uses a terminology extractor which requires only superficial syntactic analysis of the text. Evaluating this method on our corpus of reviews of electronic products (in English) gave a good precision of about 80%. This accuracy cannot be directly compared to those of previous methods because of differences in the task definition.

We then present our attempts to group the extracted product features of which there were between 400 and 800 for the products in our corpus of electronics reviews. We found that co-occurrences of features are not useful in the search of semantic relations between the features. Possible groupings found using WordNet-based measures of semantic similarity look more promising. Experiments with this method included the design and evaluation of a domain-specific word sense disambiguation algorithm. The accuracy obtained was about 53%, which is comparable to that of other methods of domain-specific word sense disambiguation.

To study the possibility of basing our summaries on natural language questions instead of queries, we also worked on the classification of questions in terms of their answer type. We present out experiments with classifying factual questions using various machine learning algorithms with n-grams as features. The best accuracies obtained were 80% for questions both in French and English; they were obtained using AdaBoost with decision trees as base learners.

Keywords: computer science, artificial intelligence, natural language processing, automatic summarisation, sentiment analysis

Table des matières

Liste des tableaux

Table des figures

Remerciements

L'auteur désire remercier Guy Lapalme pour sa direction douce et encouragement illimité, Alexandre Patry pour son extracteur de terminologie, Shahzad Khan pour son crawler, Balazs Kégl pour les discussions d'apprentissage machine et Fabrizio Gotti pour son aide technique.

Chapitre 1

Introduction

Mon mémoire est basé sur le problème pratique de traitement de commentaires de consommateurs. Plusieurs sites web fournissent maintenant des commentaires des "autres qui ont acheté ce produit" : quelques-uns le font pour les produits qu'ils vendent (*amazon.com*), et d'autres ne s'occupent que de collectionner des commentaires sur plusieurs types de produits (*epinions.com*). La longueur des commentaires varie d'un passage de quelques lignes à une ou deux pages. Un exemple assez court est présenté à la Figure 1.1.

Plusieurs personnes s'intéressent à l'information dans les commentaires de consommateurs (consommateurs potentiels, agents de marketing, etc), mais l'utilisation de ces collections est assez difficile. Pour plusieurs produits, les sites web offrent des centaines de commentaires (Hu & Liu 2004a). Quelqu'un qui veut consulter les commentaires pour un certain produit est confronté à une quantité énorme de commentaires sur plusieurs sites web. Ils sont souvent répétitifs (plusieurs commentaires d'un téléphone mentionnent, par exemple à la Figure 1.1, l'utilité de *speaker phone* ou *speakerphone*), et ils incluent souvent des anecdotes personnelles non pertinentes pour la plupart de lecteurs (par exemple à la Figure 1.1, l'information au sujet des trois téléphones dans la famille).

Pour avoir un accès plus efficace à l'information dans les commentaires de consommateurs, on voudrait qu'ils soient résumés pour qu'on puisse lire seulement ce qui est pertinent et qu'une seule fois. Le travail présenté dans ce mémoire étudie la possibilité d'un traitement automatique de commentaires de consommateurs qui permettrait de produire ce genre de résumés.

Titre : Excellent phone, excellent service.
Commentaire : I am a business user who heavily depend on mobile service. There is much which has been said in other reviews about the features of this phone, it is a great phone, mine worked without any problems right out of the box. Just double check with customer service to ensure the number provided by Amazon is for the city / exchange you wanted. After several years of torture in the hands of AT&T customer service I am delighted to drop them, and look forward to august 2004 when *I will convert our other 3 family-phones from AT&T to t-mobile* ! I have had the phone for 1 week, the signal quality has been great in the Detroit area (suburbs) and in my recent road trip between Detroit and Northern Kentucky (Cincinnati) I experienced perfect signal and reception along i-75, far superior to AT&T's which does not work along several long stretches on that same route. I have owned Motorola , Panasonic and Nokia phones over the last 8 years and generally prefer Nokia , this phone combines many of the best Nokia features, the only feature missing for me is the voice recognition. My favorite features, although there are many, are the *speaker phone*, the radio and the infrared. The speaker phone is very functional and I use it in the car, very audible even with freeway noise. The infrared is a blessing if you have a previous Nokia and want to transfer your old phone book to this phone , saved me hours of re-entering my numbers. The combination of the Nokia 6610 and T-mobile service (signal, price, service) is a winner, I highly recommend it.

FIG. 1.1 – Commentaire d'un consommateur de *amazon.com* sur un téléphone cellulaire (pris du corpus HL qui sera présenté au chapitre 3). *Speaker phone* est une caractéristique des téléphones cellulaires qui est traitée dans ce commentaires et plusieurs autres. *I will convert our other 3 family-phones from AT&T to t-mobile* est un exemple d'information anecdotique non intéressante pour la plupart des lecteurs.

1.1 Contexte : l'extraction d'information

Un but du domaine de traitement automatique de la langue naturelle (TALN) est de donner aux humains des outils pour extraire facilement et rapidement l'information dont ils ont besoin à partir d'une masse de données textuelles. L'extraction d'information du texte est un processus à plusieurs facettes.

Premièrement, l'extraction de l'information pertinente peut être effectuée à plusieurs niveaux, selon les besoins et le type du texte : recherche de documents pertinents, de paragraphes pertinents ou de phrases pertinentes. Dans le cas de commentaires de consommateurs, le niveau approprié est celui des phrases parce que la plupart des commentaires ne sont pas divisés en paragraphes.

Deuxièmement, le domaine de l'extraction d'information comprend aussi des travaux sur la fusion d'information de sources différentes dans une réponse cohérente, y compris l'élimination des répétitions et l'ordonnancement des morceaux d'information extraits. Ceci est très important dans le traitement de commentaires de consommateurs.

Troisièmement, la communication avec l'usager peut être effectuée de plusieurs façons : un dialogue humain-machine serait idéal parce que c'est le plus naturel pour les humains, mais jusqu'à date ce sont plutôt des mots clés que les usagers doivent utiliser pour ex-

primer leurs besoins. Ce mémoire présente quelques expériences sur le traitement de questions en langue naturelle.

Finalement, différents types de textes exigent des traitements variés. Les méthodes du traitement de champs textuels dans des bases de données, des textes factuels ou des textes dans lesquels les auteurs expriment leurs opinions, sont reliés mais différents. Dans ce mémoire, je me suis concentrée sur le traitement de ce dernier type de texte.

1.1.1 La fouille d'opinions (opinion mining)

La fouille d'opinions est un sous-domaine du TALN qui s'occupe de traitement d'opinions exprimées en langue naturelle. Les textes visés sont des articles de journaux, des blogs, des forums en ligne (Wiebe, et al. 2005). De plus en plus de gens expriment leurs opinions en ligne, et une capacité d'analyser ce qu'ils disent automatiquement permettrait obtenir le poulx d'opinions publiques sur les sujets discutés. Ceci pourrait intéresser des politiciens, le gouvernement, des agents de marketing, des journalistes, etc. Dans ce sous-domaine, il est important de distinguer trois choses liées mais séparées : les émotions d'une personne, les opinions d'une personne à propos d'entités (qui sont basées sur ses émotions), et le texte (les commentaires) où une personne exprime ses opinions.

Les commentaires de consommateurs sont un type de données populaire dans ce sous-domaine pour deux raisons principales :

- Ils sont normalement annotés avec des évaluations soumises par les auteurs (par exemple, 1 à 5 étoiles).
- Ils éliminent l'aspect de l'identification des sources d'opinion : on suppose que chaque commentaire présente l'avis de son auteur, et que chaque auteur n'écrit qu'un commentaire sur chaque produit.

Ceci permet de se concentrer sur d'autres défis tels que

- L'identification de passages ou phrases où des opinions sont exprimées (par exemple, *The speakerphone works better than any speakerphone I've ever had.*).
- L'identification du type d'opinion : dans le cas le plus simple, si l'attitude est positive ou négative (pour cet exemple, positive).
- L'identification de l'objet de cette opinion (*speakerphone*)[1].

[1]Ceci est aussi simplifié pour les commentaires de consommateurs où l'objet est, en général, le produit qui est connu (souvent disponible dans l'étiquette du commentaire). Néanmoins, normalement les consommateurs commentent des caractéristiques spécifiques des produits en plus du produit lui-même, donc le problème n'est pas éliminé.

D'autres ensembles de données ont été rendues disponibles grâce à Janice Wiebe et son équipe (Wiebe et al. 2005), qui ont développé une méthodologie d'annotation de textes d'avis et l'ont appliquée à un corpus d'articles de journaux[2]. Ils ont aussi commencé à travailler sur l'identification de sources d'opinions, qui est un problème moins pertinent aux commentaires de consommateurs mais très important dans le traitement de textes exprimant des opinions en général[3].

1.1.2 Ressources

Dans le domaine de l'extraction d'information, on utilise en général trois types de connaissances. Plusieurs méthodes ne les utilisent pas tous, mais toutes en utilisent au moins un.

- Les *connaissances statistiques* sont des observations à base d'un grand corpus de texte. Ce sont des patrons découverts (par exemple que certains mots apparaissent souvent dans la même phrase), souvent avec des probabilités associées. L'identification de patrons pertinents dans une masse de données textuelles est une application d'apprentissage machine, un domaine de l'intelligence artificielle.
- Les *connaissances linguistiques* décrivent le fonctionnement de la langue en question sur plusieurs niveaux (grammaire, lexique, morphologie, etc).
- Les *connaissances du monde* sont des informations sur le domaine en question (par exemple, la botanie) ou le monde en général (par exemple, le fait que le ciel est bleu est un élément possible des connaissances du monde). Elles peuvent être représentées de plusieurs façons. Les meilleures façons de représentation qui permettent utiliser les connaissances pour raissoner est récherché dans le domaine de l'intelligence artificielle.

Pour chaque type de connaissances, il y a plusieurs méthodes de création et d'utilisation. Les connaissances statistiques sont très populaires parce qu'aucun effort humain n'est exigé une fois que les algorithmes d'extraction d'information pertinente ont été développés. Néanmoins, ils sont rarement suffisants pour une analyse intelligente du texte.

Les connaissances linguistiques sont créées avec l'aide de linguistes et augmentées en utilisant des méthodes statistiques. Ces connaissances sont normalement spécifiques à une langue ce qui constitue son plus grand désavantage. Un autre désavantage est que les connaissances linguistiques disponibles sont difficiles à appliquer sur le langage informel.

[2]Le corpus est disponible de http ://www.cs.pitt.edu/mpqa/databaserelease/

[3]Ce problème a une dimension intéressante de plusieurs niveaux de sources d'opinion où *quelqu'un pense que quelqu'un pense ... que X*.

Ceci limite leur utilisation à des textes écrits professionnellement (journaux, etc), mais le traitement de textes informels peut quand même en profiter.

Les connaissances du monde visent à remplacer pour l'ordinateur l'expérience disponible aux humains qui peuvent ressentir les objets réels. Mettre ces connaissances dans une forme utilisable par une machine est un processus exigeant et long, donc beaucoup d'effort est fait pour l'automatiser. Des ressources disponibles pour l'automatisation sont les ressources lexiques (par exemple, les dictionnaires déjà créés par les lexicographes), et les textes eux-mêmes (par exemple, les descriptions de choses dans le monde sont utiles).

Parce que la création de ressources de connaissances est un problème en soi, qui nécessite normalement une intervention manuelle, cette approche est d'habitude appliquée à des domaines spécifiques. A cause de l'exigence de la création de ce type de connaissances, il n'est pas aussi populaire que les connaissances statistiques. Néanmoins, l'effort vaut bien la peine : l'analyse à base de connaissances peut être précis, coopératif (ceci sera discuté en chapitre 2), et ne dépend pas d'un corpus de données.

On constate que même dans ce survol très bref, les types de connaissances sont très liés. Dans mon mémoire, j'utilise les connaissances statistiques et des lexiques pour créer une ressource de connaissance du monde semi-automatiquement, et les connaissances statistiques pour classifier des questions en langue naturelle.

1.2 Survol du mémoire

Dans ce mémoire, je présente mon travail sur le traitement de commentaires de consommateurs à base de connaissances du monde. A cause de la disponibilité des données, j'ai choisi l'électronique comme domaine d'expérimentation.

Le mémoire est organisé de la façon suivante. En me basant sur des travaux précédents présentés dans le Chapitre 2, j'ai développé l'architecture d'une application qui résume un ensemble de commentaires sur un produit (chapitre 4). Le corpus de données est présenté au Chapitre 3. La création semi-automatique de l'ontologie comprend une partie manuelle (Chapitre 5), et une partie automatique : l'extraction automatique de termes du domaine (Chapitre 6) et l'organisation de ces termes d'une façon cohérente (Chapitre 7). De plus, j'ai travaillé sur l'analyse de questions pour rapprocher la possibilité de remplacement des requêtes par un dialogue humain-machine en langue naturelle via des méthodes statistiques (Chapitre 8). Les travaux futurs et la conclusion forment les deux derniers chapitres.

Chapitre 2

Travaux connexes

Notre méthodologie de résumé de commentaires de consommateurs a été inspirée par plusieurs travaux précédents. Dans ce chapitre, nous faisons une revue de ces travaux, en commençant avec ceux qui sont déjà utilisés dans le traitement de commentaires de consommateurs, suivis par ceux que nous avons jugé pertinents même s'ils n'ont pas encore été appliqués à ce problème.

2.1 Sémantique lexicale

Plusieurs méthodes de traitement des commentaires de consommateurs utilisent WordNet et les mesures de similarité sémantique de concepts associés. Notre approche au regroupement de caractéristiques de produits extraits automatiquement (Chapitre 7) est basée sur une méthodologie semblable. Dans cette section, nous présentons d'abord WordNet, suivi du problème de désambiguïsation de sens et les mesures de similarité sémantique à base de WordNet. Cette discussion est basée sur (Patwardhan, et al. 2003).

2.1.1 WordNet

WordNet est un dictionnaire anglais développé à l'Université de Princeton disponible en format électronique[1]. Son organisation est différente de la plupart de dictionnaires. Il a été développé dans un laboratoire de sciences cognitives comme un modèle de représentation humaine de connaissances lexicales ; ce modèle est basé sur des études psycholinguistiques. L'initiative de EuroWordNet a développé des dictionnaires semblables pour d'autres langues, mais celui pour l'anglais reste toujours le plus complet.

[1]http ://wordnet.princeton.edu/

Les concepts synonymes sont mis dans des `synsets` (contraction pour `synonym sets`), par exemple, {*car, auto, automobile, machine, motocar*}. Chaque synset a une définition, comme celles des dictionnaires courants. En plus, WordNet contient des liens entre les synsets qui représentent plusieurs relations : *is-a (est-un(e))*, *part-of (fait-partie-de)*, *synonymie*, *antonymie*, etc. Ceci crée un réseau de concepts qui est l'avantage principal de WordNet qui est maintenant devenu une ressource standard en TALN.

WordNet contient des noms (117,097 dans version 2.1), verbes (11,488), adjectifs (22,141) et adverbes (4,601). Il n'y a pas jusqu'à date beaucoup de liens entre les mots de categories différentes, mais des nouvelles connexions sont ajoutées à chaque nouvelle version. Par exemple, dans la version 2.1, les adjectifs *light* et *heavy* sont liés au nom *weight*.

2.1.2 Désambiguïsation de sens (Word Sense Disambiguation)

Chaque mot dans WordNet (comme dans tous les dictionnaires) a un ou plusieurs sens. L'exemple classique est *bank*, dont les sens (les synsets et les définitions) comme nom sont présentés en Figure 2.1.

* depository financial institution, bank, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) "he cashed a check at the bank" ; "that bank holds the mortgage on my home"
* bank (sloping land (especially the slope beside a body of water)) "they pulled the canoe up on the bank" ; "he sat on the bank of the river and watched the currents"
* bank (a supply or stock held in reserve for future use (especially in emergencies))
* bank, bank building (a building in which the business of banking transacted) "the bank is on the corner of Nassau and Witherspoon"
* bank (an arrangement of similar objects in a row or in tiers) "he operated a bank of switches"
* savings bank, coin bank, money box, bank (a container (usually with a slot in the top) for keeping money at home) "the coin bank was empty"
* bank (a long ridge or pile) "a huge bank of earth"
* bank (the funds held by a gambling house or the dealer in some gambling games) "he tried to break the bank at Monte Carlo"
* bank, cant, camber (a slope in the turn of a road or track ; the outside is higher than the inside in order to reduce the effects of centrifugal force)
* bank (a flight maneuver ; aircraft tips laterally about its longitudinal axis (especially in turning)) "the plane went into a steep bank"

FIG. 2.1 – L'entrée du nom *bank* dans WordNet : 10 sens (synsets) avec ses définitions.

Les sens sont ordonnés par leur fréquence dans un grand corpus annoté manuellement. Les nombres moyens de sens par mot sont : 1.23 pour les noms, 2.16 pour les verbes, 1.41 pour les adjectifs, et 1.24 pour les adverbes.

La polysémie est une barrière à l'utilisation des connaissances de WordNet parce que des textes contiennent des mots et non des sens. Pour obtenir les synonymes d'un mot, par exemple, il faut donc savoir quel sens du mot est approprié.

Faire ceci automatiquement est le problème du WSD (*word-sense disambiguation* ou la désambiguïsation de sens). La plupart des méthodes qui visent à résoudre ce problème utilisent le contexte du mot pour déterminer son sens. Par exemple, pour la phrase *He has money in the bank.*, le mot *money* peut être utilisé. Des fois, le contexte n'aide pas et il y a ambiguïté même pour les humains : *He went to the bank.* – est-il allé vers la rive où à la banque ? Il est donc important d'être capable d'identifier les cas où la désambiguïsation n'est pas possible.

Les sens des mots sont définis par des lexicographes et sont souvent discutables. Par exemple, le sens de *bank (the funds held by a gambling house or the dealer in some gambling games)* est-il vraiment un sens séparé de celui de la banque où on conserve l'argent ? Certains prétendent que les sens de WordNet sont trop détaillés et que les différences entre eux sont trop fines. Le problème de désambiguïsation est donc en pratique moins difficile qu'on ne le suppose a priori.

2.1.3 Similarité sémantique

Comment peut-on utiliser le mot *money* pour choisir le bon sens de *bank* ? On espère que dans le réseau de WordNet, *money* et le bon sens de *bank* sont plus proches que *money* et les mauvais sens. Ici, proche fait référence à une distance ou similarité sémantique[2].

Il y a bien sûr d'autres applications de similarité sémantique. Par exemple, nous allons l'utiliser pour regrouper des caractéristiques de produits. (Hirst & St-Onge 1998) l'ont utilisée pour la détection de chaînes sémantiques dans un texte.

Le concept lui-même est difficile à définir. Essentiellement, la similarité sémantique est la probabilité d'occurrence de deux concepts dans le même contexte. On ne peut pas dire que les humains peuvent bien mesurer la distance sémantique parce que la notion dépend des connaissances et de la culture. Par exemple, pour un habitant d'Afrique, *transport* et *camel* sont beaucoup plus proches que pour quelqu'un de Montréal. Néanmoins, quelques

[2]Une notion reliée est celle de voisinage sémantique ; ceci est plus général que la similarité et peut inclure des antonymes, etc.

paires de concepts sont clairement plus proches que d'autres (*book,shelf* versus *book,wheel*), donc la notion de distance sémantique est raisonnable malgré le flou de sa définition.

Mesures de similarité sémantique

Il y a plusieurs propositions de mesure de cette distance. Il existe des mesures de similarité sémantique qui n'utilisent pas WordNet ; par exemple, ceux à base de thésaurus. Ici, nous présentons trois mesures de similarité sémantique à base de WordNet qui sont pertinentes à notre travail.

Dans notre expérience, nous utilisons le module Perl de WordNet développé par Ted Pedersen et ses étudiants. Ce module s'appelle *WordNet : :Similarity* et il implante plusieurs mesures de similarité à base de WordNet[3]. Les noms des mesures utilisées ici viennent de ce module, présenté dans (Pedersen, et al. 2004).

Les mesures de similarité sémantique à base de WordNet utilisent généralement la longueur des chemins les plus courts entre des concepts dans le réseau de relations. Quelques-uns se limitent à la relation *is-a*, d'autres incluent toutes les relations.

path : calcule le nombre minimum de noeuds entre deux synsets, et retourne l'inverse de cette valeur comme réponse. Pour deux mots synonymes (dans le même synset), il retourne 1. Seulement les liens représentant la relation *is-a* sont utilisés.

res Cette mesure proposée par Resnik en 1995 se base sur le concept de *Information Content (IC)*. Ceci est une valeur donnée à chaque noeud (synset) de WordNet. IC est estimé à base de fréquence des concepts dans un grand corpus de telle sorte que l'IC de mots spécifiques (par exemple, *candle wick*) soit petit et que l'IC de mots généraux (par exemple, *entity*) soit grand. Resnik a proposé de calculer IC de cette façon : $IC(concept) = -log(P(concept))$ en utilisant un corpus où les sens de mots sont annotés manuellement[4]. Etant donné l'IC de chaque noeud, selon Resnik, la similarité sémantique de deux concepts est égale à l'IC du noeud le plus bas dans la hiérarchie qui est ancêtre des deux concepts en question (*lowest common subsumer*).

lin La mesure de similarité sémantique proposé par Lin en 1997 se base aussi sur le concept d'IC. La différence est que Lin a utilisé l'IC de concepts (c1, c2) dont la similarité est mesurée. La similarité est donnée par $\frac{2*res(c1,c2)}{IC(c1)+IC(c2)}$: la quantité d'information nécessaire pour décrire la partie commune de ces concepts divisé par la quantité d'information nécessaire pour décrire les concepts eux-mêmes.

[3]Il est disponible librement sur *http ://www.d.umn.edu/˜tpederse/similarity.html*.
[4]Le même corpus est utilisé pour ordonner les sens pour chaque mot.

Plusieurs autres mesures ont été proposées (Patwardhan et al. 2003). Ayant introduit la similarité sémantique, nous pouvons procéder à la présentation de méthodes existantes de traitement de commentaires de consommateurs.

2.2 Résumé de commentaires de consommateurs

Depuis une dizaine d'années, les commentaires de consommateurs sont un type de données populaire. Les principaux travaux qui les traitent se concentrent sur les aspects suivants :

- l'identification automatique des aspects des produits (Hu & Liu 2004a), (Popescu & Etzioni 2005)
- l'extraction des phrases exprimant des opinions sur des aspects (Yu & Hatzivassiloglou 2003), (Dave, et al. 2003), (Morinaga, et al. 2002)
- la classification de ces phrases comme positive ou négative (ou neutre) (Dave et al. 2003), (Gamon 2004), (Beineke, et al. 2004), (Morinaga et al. 2002), (Turney 2002)

Nous avons travaillé plutôt sur le premier aspect ; les travaux précedents pertinents sont discutés dans la prochaine section. Les deux derniers aspects sont des sous-problèmes de l'analyse des émotions (sentiment analysis), un module important de notre application présentée au chapitre 4 ; les travaux précedents pertinents à ce module seront présentés au section 2.4.

Le seul système complet de résumé des commentaires de consommateurs a été proposé par (Hu & Liu 2004a). Leur système crée des résumés structurés de tous les commentaires sur un produit spécifique (par exemple, *Nokia cellphone C16*). Ils traitent de l'identification automatique des caractéristiques des produits et de la classification des mots d'opinion (par exemple, *excellent*) trouvés. Comme résumé, on présente à l'usager une liste des aspects du produit et le nombre des phrases positives/négatives trouvées pour chacun. Si l'usager veut en savoir plus que le nombre de phrases positives/négatives au sujet d'une caractéristique, il peut obtenir la liste complète de ces phrases, souvent longues et répétitives.

Hu & Liu ont aussi travaillé sur la visualisation d'informations trouvées dans les commentaires, ce qui facilite la comparaison de produits (Liu, et al. 2005). En plus, ils ont traité des commentaires structurés comme les listes de "pro's" et "con's", mais ceci n'est pas relié à notre travail sur les commentaires libres.

Nous présentons maintenant les détails de leur méthodologie pour les parties pertinentes à notre travail. Hu & Liu ont aussi travaillé dans le domaine d'électronique ; leur

corpus de données sera présenté dans le chapitre suivant.

2.3 Traitement de la terminologie

On appelle les caractéristiques (ou les aspects) de produits la terminologie du domaine. Hu & Liu s'intéressent à l'extraction automatique de la terminologie parce que leurs résumés structurés y sont basés.

2.3.1 Hu & Liu

(Hu & Liu 2004b) ont utilisé une partie d'un algorithme de "association rule mining" dont le but était de trouver les ensembles de mots avec co-occurrence dans la même phrase plus grand qu'un seuil. Puis ils utilisent deux heuristiques pour nettoyer la liste de candidats :

- Pour les candidats complexes (de plusieurs mots) : si les mots dans le candidat n'apparaissent jamais directement à côté, le candidat est éliminé.
- Pour tous les candidats : si le candidat fait partie d'un autre candidat (complexe), et n'apparaît pas comme nom ou phrase nominale plus qu'un certain nombre de fois dans le corpus, on le juge redondant et l'élimine.

Leurs méthodes pour les étapes suivantes sont simplistes : une liste de mots d'attitude (*wonderful*, *disappointing*, etc) est créée automatiquement en cherchant les adjectifs à côté des caractéristiques extraites, et les caractéristiques moins fréquentes sont cherchées en utilisant cette liste (le nom ou phrase nominale le plus proche à l'adjectif). L'algorithme est évalué en utilisant le corpus annoté (designé HL) que nous allons décrire dans le chapitre suivant.

(Hu & Liu 2004b) utilisent WordNet pour regrouper les termes extraits d'une façon très conservative : seulement les termes dans le même synset sont regroupés. (Hu & Liu 2004b) n'utilisent aucune désambiguisation de sens, et seuls les deux sens les plus fréquents sont considérés. Le regroupement de termes est un élement mineur de leur travail.

2.3.2 Carenini et al.

(Carenini, et al. 2005) est un article dédié entièrement au regroupement de termes. Leur méthodologie est basée sur des taxonomies prédéfinies et des mesures de similarité sémantique. Les termes qu'ils regroupent sont ceux dans l'annotation du corpus HL (qui

sera présenté dans le chapitre suivant) pour deux produits : appareils photo numériques (101 termes) et lecteurs de DVDs (116 termes). Ils les projètent sur une taxomonie de termes créée à la main (pris de *activebuyers.com*). Nous décrivons brèvement les taxonomies et les mesures de similarité sémantique utilisées.

La figure 2.2 montre une partie de la taxonomie à trois niveaux pour les appareils photo. Cette taxonomie d'appareils photo a été changée parce qu'elle mélangeait les caractéristiques de l'appareil lui-même avec les caractéristiques des images qu'il produit ; elle était donc séparée en deux taxonomies pour des appareils photo et des images. Au cours d'expériences, un autre noeud a été déplacé pour corriger une "erreur", ce qui a amélioré le résultat rapporté par 20% (de 0.44 à 0.35). Ceci démontre la subjectivité de la structure de taxonomies de termes : il y a plusieurs structures à peu près correctes.

Les deux taxonomies finales ont les spécifications suivantes :

- Pour lecteurs de DVD : 38 noeuds ; profondeur de 2 niveaux.
- Pour appareils photo numériques :
  - pour l'appareil : 73 noeuds, profondeur de 3 niveaux.
  - pour l'image : 13 noeuds, profondeur de 3 niveaux.

Partial view of *UDF* taxonomies for digital camera.

Camera	Image
Lens	Image Type
Aperture Modes	TIFF
Optical Zoom	JPEG
...	...
Editing/Viewing	Resolution
Viewfinder	Effective Pixels
...	Aspect Ratio
Flash	...
...	

FIG. 2.2 – Une partie de taxonomies pour appareils photo et ses images (Carenini et al 2005).

Pour projeter les termes sur les noeuds de la taxonomie, Carenini et al. ont utilisé les mesures de similarité sémantique et, comme Hu&Liu, la présence dans le même synset. Le WSD est fait à la main pour tous les mots dans les termes. Pour les termes complexes, la moyenne des scores de similarité de toutes les paires de mots a été utilisée. Chaque terme a été projeté sur le noeud le plus proche (dans le sens de similarité sémantique) si le score de leur similarité dépassait un seuil déterminé empiriquement. Projeter un terme

sur un noeud signifie que le terme est devenu un enfant du noeud dans la taxonomie.

Pour l'évaluation, une référence a été créée à la main. La précision du placement est mesurée par la distance moyenne en nombre de liens entre noeuds entre le placement automatique et la référence. Les résultats rapportés sont environ 0.35 pour les appareils photo, et environ 0.27 pour les lecteurs de DVD.

Ce travail est plutôt de l'enrichissement d'une taxonomie. En se basant sur ces expériences, nous allons présenter au chapitre 7 un essai d'organisation d'un ensemble de termes sans une taxonomie prédéfinie.

2.3.3 Popescu et al.

Le travail de (Popescu & Etzioni 2005) ne peut pas être présenté en détail parce que le système pour le traitement des commentaires des consommateurs est basé sur un autre système plus général *KnowItAll* (Etzioni, et al. 2005). Essentiellement, leur méthode prend tous les syntagmes nominaux (SN) dans un corpus de commentaires et calcule le PMI (point-wise mutual information) entre chaque SN (les candidats) et des phrases comme *of scanner*, *scanner has* générées automatiquement (Etzioni et al. 2005). Le PMI est calculé soit en utilisant le corpus de commentaires, soit en utilisant le Web estimé à l'aide du nombre de documents trouvés par une requête. Les scores PMI sont fournis à un classificateur Naive Bayes qui répond avec la probabilité que le candidat soit un terme. En utilisant un corpus de commentaires (HL), la précision est de 78% (6% de plus que Hu&Liu) avec un rappel de 64% (16% plus bas que Hu&Liu). En ajoutant le PMI du Web, la précision monte à 94% (22% de mieux que Hu&Liu) avec un rappel de 77% (3% plus bas que Hu&Liu). Pour démontrer la stabilité de cette méthodologie, (Popescu & Etzioni 2005) ont fait des évaluations à la main sur deux autres produits : hôtels et numériseurs. Les résultats étaient : une précision de 89% et un rappel de 73%.

Il faut noter qu'il est étonnant que Hu&Liu et Popescu et al. utilisent des analyseurs syntaxiques (NLProcessor et MINIPAR) sur les commentaires pour trouver les NPs. Étant donné l'informalité des textes, il est remarquable que la qualité d'analyse soit suffisante pour obtenir des bons résultats.

Un autre essai d'extraction des caractéristiques de produits a été fait par (Kobayashi, et al. 2004) ; son approche est semi-automatique et exige une aide humaine à chaque itération.

L'autre grande partie du traitement de commentaires de consommateurs et de textes d'avis en général est l'identification et la classification des expressions d'attitude ; les

travaux sur ce problème sont présentés à la section suivante.

2.4 Analyse des émotions (sentiment analysis)

Dans "Calcul Affectif" (Picard 1997), R. Picard explique, en se basant sur des études psychologiques, que les émotions humaines sont très importantes pour la capacité de prendre des décisions. C'est une des raisons pourquoi les machines ne raisonnent pas bien dans certains cas où les humains le font facilement. Donc, pour développer des machines "intelligentes", il est important d'étudier les émotions pour qu'on puisse avoir des systèmes qui "comprennent" les émotions des usagers et incluent même un module d'émotions.

Plusieurs moyens ont été proposés pour découvrir les émotions des usagers : les expressions du visage, les changements physiques, etc. Un problème de base est de comprendre leurs émotions quand ils les expriment via la langue naturelle. Les commentaires des consommateurs sont une ressource où les auteurs expriment leurs opinions et leurs sentiments ouvertement. L'objet de leurs commentaires est un produit connu, mais c'est souvent plus spécifique : des aspects du produit, des modes, la compagnie, etc. Il faut mentionner que souvent, les opinions (*I think X is good*) sont moins émotives que *I was frustrated by X*; néanmoins, ce sont des phrases subjectives où les émotions sont clairement pertinentes.

A cause des évaluations disponibles, les commentaires de consommateurs sont un type de données très populaire dans ce domaine. La plupart de méthodes sont à base de lexique : des mots (plutôt des adjectifs) sont classifiés selon l'attitude (normalement positif/négatif) soit à la main, soit automatiquement, et les commentaires sont classifiés selon l'attitude à base de fréquence de mots d'attitudes différents.

Beaucoup de travaux ont été faits pour classifier des phrases comme subjectives ou objectives (Yu & Hatzivassiloglou 2003), (Dave et al. 2003), (Morinaga et al. 2002), classifier des phrases ou paragraphes comme positives ou négatives (ou neutres) (Yu & Hatzivassiloglou 2003), (Dave et al. 2003), (Gamon 2004), (Beineke et al. 2004), (Morinaga et al. 2002), et estimer la force des émotions dans une phrase (Wilson, et al. 2004). On a aussi utilisé des connaissances générales du monde pour classifier la direction émotive de phrases comme une des six émotions de base (Ekman 1992, {bonheur, tristesse, colère, crainte, dédain, surprise}) (Liu, et al. 2003).

Ce problème est assez bien étudié et les résultats sont suffisants pour le traitement des commentaires de consommateurs (Carenini et al. 2005). Une avenue de recherche

future intéressante est le nuancement de l'attitude, qui a été déjà commencé par (Liu et al. 2003), mais pour l'application de commentaires de consommateurs, la granularité *positive/négative/neutre* est suffisante. Quelques problèmes majeurs qui restent dans le domaine de classification du texte selon l'attitude sont l'ironie et le sarcasme, l'utilisation de négations, utilisation des mots émotifs dans un sens différent.

Dans mon travail, je suppose qu'un classificateur de phrases selon leurs attitudes (positive/négative/neutre) est disponible (ou peut être facilement créé).

2.5 L'interaction coopérative

Une partie de mon travail a été inspiré et est basé sur les travaux de (Benamara 2004), (Benamara & Saint-Dizier 2004). Dans ces articles, ils décrivent leurs études de question-réponse coopérative dans les domaines spécifiques de tourisme et santé. La question principale de leur recherche est comment répondre aux questions des usagers *coopérativement*, comme le font des humains (des agents de voyage où des professionnels de santé). Dans leurs travaux, ils étudient :

- Les types de questions posées normalement par les usagers.
- Les types de coopération effectués par les humains en répondant aux questions - le contenu et la forme de réponses sont étudiés.
- Les connaissances du monde nécessaires pour la coopération et la meilleure façon de les représenter.
- Les méthodes de raisonnement sur les connaissances du monde et les questions nécessaires pour la coopération.

Leur but principal est la recherche automatique de réponses les plus appropriées et précises possibles. Ceci diffère des réponses fournies par la plupart de systèmes modernes qui sont des documents complets à base de mots clés. Dans ce cas, les usagers doivent faire beaucoup de travail : choisir soigneusement les mots clés et trouver l'information qu'ils cherchent dans les documents fournis.

Benamara et al. ont étudié quelques collections de FAQ (Foire Aux Questions), un ensemble de vraies questions posées par des humains et répondues par courrier électronique par d'autres humains. Ceci était leur modèle d'interaction idéale entre l'usager et le système de recherche d'information ; ils l'ont utilisé pour identifier les élements essentiels d'interaction coopérative. La liste des éléments qu'ils ont découverts suit (Benamara & Saint-Dizier 2004). Nous présentons un exemple de question/réponse pour chaque type d'interaction coopérative tiré du même article.

- Si la réponse directe est trop longue, il faut l'organiser où la résumer.
  – *Où sont les gîtes en France avec piscine ?*
  – *En général, les gîtes du sud de la France ont une piscine.*
- Si la question contient des suppositions erronées, il faut les détecter et les accommoder (en expliquant les suppositions erronées et en proposant des alternatives "corrigées" de la question ou requête).
  – *A quelle heure est l'avion de Paris à Albi demain ?*
  – *Il n'y a pas d'aéroport commercial à Albi.*
- Si la question n'a pas d'une réponse, il faut l'indiquer et proposer d'autres informations pertinentes.
  – *Un chalet en Corse pour 15 personnes*
  – *La capacité de chalets est moins de 10 personnes en Corse. Solutions alternatives : deux chalets adjacents en Corse, un chalet pour 15 personnes dans une région proche en Corse, un autre type d'hebergement (hôtel, pension) en Corse.*
- Si la réponse dépend d'une condition, il faut la présenter avec les réponses conséquentes pour chaque possibilité.
  –*Quelles sont les réductions que vous proposez pour les trains ?*
  – *Si vous avez moins de 25 ans, ...*
  Si vous avez plus de 60 ans, ...
  Autrement, ...
- Il faut savoir présenter l'information très pertinente à la question mais pas directement demandée.
  – *Vos chambres ont-elles une vue sur le fleuve ?*
  – *Oui. De chacune de nos chambres vous avez une vue (limitée et non panoramique) sur le fleuve.*

Au chapitre 4, nous présenterons une façon d'adapter la notion d'interaction coopérative au traitement de commentaires de consommateurs. Nous nous sommes concentré sur l'aspect de détection des suppositions erronées qui est particulièrement pertinent pour notre domaine.

2.6 Conclusion

En conclusion, cette revue de littérature a présenté plusieurs travaux pertinents à notre travail décrit dans les chapitres suivants. Ceci comprend des travaux de traitement de commentaires de consommateurs (nos expériences présentées aux chapitres 4, 6 et 7 se

basent sur eux), des travaux de classification de texte selon l'attitude (un module présenté au chapitre 4 s'appuie sur ces méthodes), des travaux sur l'interaction coopérative dans la recherche d'information (un autre module présenté au chapitre 4 utilise ces méthodes), et une introduction à la sémantique lexicale qui sera utilisée au chapitre 7. D'autres travaux précédents pertinents, mais qui n'ont pas été présentés ici, sont ceux qui traitent le résumé automatique de plusieurs textes, l'extraction de terminologie, et l'extraction d'information à base de connaissances.

Chapitre 3

Les corpus de commentaires

Dans ce projet, nous utilisons deux corpus de commentaires des consommateurs : un est petit et annoté, l'autre est grand et brut. Dans ce chapitre, nous en présentons les détails. Nous nous limitons à quatre types de produits électroniques : appareils photo numériques, lecteurs de DVD et MP3 (media digitaux), et téléphones cellulaires.

3.1 Corpus HL

Nous appellerons le petit corpus annoté *corpus HL*, selon les initiales des noms de ses créateurs Hu et Liu.

Produit	**No. de commentaires**	**No. moyen de mots**
Appareil photo numérique (2)	79	≈ 235
Téléphone cellulaire (1)	41	≈ 235
Joueur de MP3 (1)	95	≈ 340
Lecteur de DVD (1)	99	≈ 130
5 produits	314 commentaires	≈ 235

Tab. 3.1 – Corpus HL.

Ce corpus (voir table 3.1) comprend 314 commentaires sur 5 produits. Cet ensemble a été annoté et présenté dans (Hu & Liu 2004a). C'est une source semi-structurée : plus structurée que le texte, et moins qu'une BD. Les commentaires viennent de Amazon.com ; chaque commentaire correspond à un paragraphe. La Table 3.2 décrit les annotations utilisées dans ce corpus.

Annotation	Description
'[t]'	marque les titres des commentaires (il est donc facile de savoir où un commentaire commence et finit)
'xxxx[+|−n]'	xxxx est une caractéristique du produit, +|− correspond à une opinion positive/négative, n est le niveau de positivité/négativité (à cause de la subjectivité, on recommande ne pas utiliser le n)
'xxxx[u]'	marque que le nom de la caractéristique n'est pas dans la phrase, *support[-3][u] apex does n't answer the phone*
'xxxx[p]'	marque que le nom de la caractéristique n'est pas dans la phrase, mais il y a une référence, *player[+2][p] i liked this one enough to buy another*

TAB. 3.2 – Quelques annotations du corpus HL utiles pour nous.

3.2 Discussion de l'annotation

Dans leur annotation, (Hu & Liu 2004a) cherchaient à identifier les caractéristiques des produits qui étaient commentés par des consommateurs. Même si dans plusieurs phrases les caractéristiques sont faciles à identifier, ceci n'est pas toujours le cas. La première question est de savoir si les caractéristiques sont toujours des syntagmes nominaux ou si les verbes et adjectives comptent aussi. La deuxième question est de determiner si les références au produit lui-même doivent être traités comme caractéristiques. Les exemples suivants illustrent ces cas.

– *The phone is small and light.* – Est-ce que *small* et *light* sont des caractéristiques ?
– *The phone looks good.* – *looks good* est-il une caractéristique ?
– *The unit worked perfectly straight out of the box.* – *unit* est-il une caractéristique ?

La réponse dépend d'utilisation de l'annotation ; nous y reviendrons à la section 6.3.

3.3 Corpus SK

Grâce a Shahzad Khan[1] qui a écrit un crawler, nous avons obtenu un corpus de commentaires de *www.epinions.com*. La table suivante décrit le contenu de ce corpus qui comprend à peu près 8094157 mots. Contrairement au corpus HL, les commentaires sont divisés selon le type de produit dont ils parlent, et non par produits concrets. Nous

[1]Etudiant au doctorat à l'Université de Cambridge ; son site web : www.cl.cam.ac.uk/∼sk453/index.html.

appellerons ce corpus *corpus SK*.

Produit	**No. de commentaires**	**No. moyen de mots**
Appareil photo numérique	10605	≈ 340
Téléphone cellulaire	4511	≈ 460
Lecteur de MP3	4042	≈ 370
Lecteur de DVD	3266	≈ 280
Total	22424	≈ 360

TAB. 3.3 – Corpus SK.

3.4 Particularités de ces corpus

Une particularité de ces corpus est que les commentaires sont écrits par le grand public, donc il y a beaucoup d'agrammaticalités et d'erreurs d'orthographe (par exemple, *workin* au lieu de *working*, ou *ringtone* au lieu de *ring tone* ou *ring-tone*).

(Hu & Liu 2004b) utilisent l'extraction automatique des radicaux de mots (stemming) et une correspondance floue (fuzzy mathing) pour résoudre ce problème. Les détails de leur méthode ne sont pas présentés dans leur article ; notre méthode est probablement assez semblable à la leur. Nous comptons ignorer les malapropismes (*two go*), mais on utilise un correcteur orthographique pour corriger les "mots" qui n'existent pas en utilisant GNU Aspell[2] et une correspondance floue basée sur l'algorithme de Levenstein. L'autre problème est celui des abréviations comme *eax*, inconnues même de plusieurs anglophones, qui sont spécifiques au produit ou à son type. Ceux-ci sont combinés dans une liste pour qu'on ne les "corrige" pas.

Pour le corpus HL, on a créé à main une liste des abréviations et compagnies/modèles ; pour le corpus SK, ceci serait prohibitif. Même si on essayait de le faire en utilisant un extracteur d'entités nommées, la grandeur du corpus rend les erreurs négligeables.

3.5 L'utilisation des corpus

Le corpus HL a été utilisé en premier pour se familiariser avec les particularités de ce genre de texte. L'annotation d'attitude est utilisée dans le prototype présenté au prochain

[2]http ://aspell.sourceforge.net/

chapitre. Dans l'expérience d'extraction de terminologie présentée au chapitre 6, le corpus HL était aussi utilisé comme corpus d'entraînement où quelques termes étaient annotés. Le corpus SK a été utilisé pour l'extraction de termes du domaine (chapitre 6) à base de patrons appris.

Chapitre 4

Architecture de l'application

Dans ce chapitre nous proposons une architecture d'un logiciel de résumé de commentaires des consommateurs à partir d'une requête. Nous décrivons le fonctionnement général du système module par module. Même si ce système fonctionne à base de requêtes, nous gardons à l'esprit le but éventuel d'un fonctionnement à base de questions en langue naturelle. Basé sur l'architecture proposée, un prototype a été créé dont les particularités sont présentées à la fin de chaque section.

La figure 4.1 montre la vue générale du système : ses modules, ses ressources principales et les points d'interaction avec l'usager. L'application est lancée quand un usager soumet une requête. Premièrement, on extrait les noms des produits, compagnies, et modèles en utilisant un tableau de noms possibles. Ensuite, on extrait les aspects des produits (termes[1]).

L'information extraite est fournie au module d'analyse coopérative qui cherche des suppositions erronées dans la requête et communique avec l'usager pour les résoudre si nécessaire. La requête finalisée est ensuite passée au module de recherche des phrases pertinentes qui sont classifiées selon leur attitude. En représentant chaque phrase par son attitude et les termes qu'elle contient, on élimine celles qui sont redondantes. Les phrases qui restent sont réordonnées pour créer un résumé à afficher à l'usager.

Dans les sections qui suivent, nous expliquons chaque module en détail. Nous allons utiliser la requête *Nokia speakerphone* comme exemple. Cette requête exprime que l'usager veut voir des commentaires sur le *speakerphone* des téléphones manufacturés par Nokia.

[1]On considère les aspects des produits comme les termes de notre domaine. Quelques exemples sont *volume*, *design*, *menu system*.

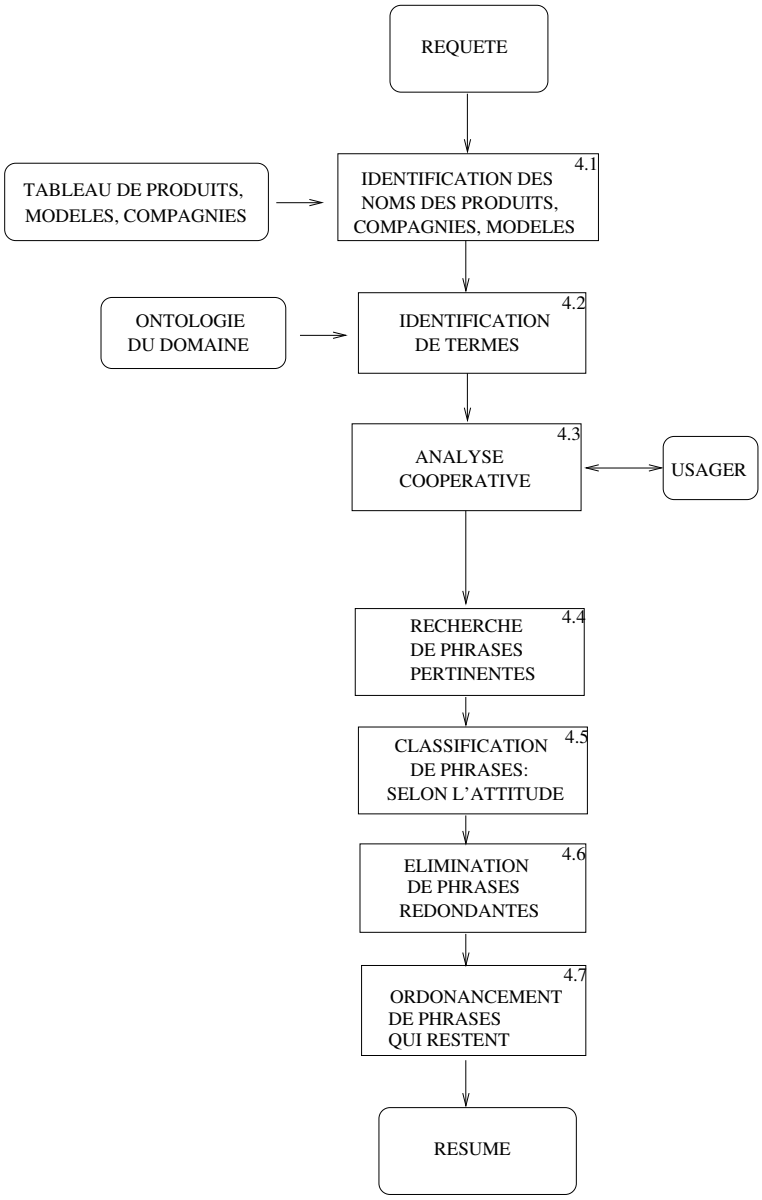

FIG. 4.1 – Survol du système. Les flèches indiquent l'ordre des opérations et l'utilisation des ressources. Les rectangles représentent les modules. Les rectangles aux coins arrondis représentent des ressources et l'interaction avec l'usager. Les sections du chapitre discutant chaque module sont indiquées dans les coins supérieurs droit des boîtes.

4.1 Identification de noms de produits, compagnies et modèles

Ayant reçu une requête, nous identifions en premier les noms de produits, compagnies, et modèles. Pour notre exemple, nous trouvons *Nokia* comme le nom d'une compagnie, et aucune mention ni d'un produit ni d'un modèle (un exemple d'un nom de modèle est *ISO61*). Ceci est fait en utilisant des listes pré-compilées des noms de produits, compagnies et modèles. Ces listes sont organisées dans un tableau de triplets de la forme `compagnie, produit, modèle`. Ce tableau existe probablement déjà chez les propriétaires des sites web des commentaires de consommateurs, mais il peut également être créé en utilisant des étiquettes sur les commentaires de consommateurs. Par exemple, sur le site *epinions.com*, on trouve des étiquettes comme *Panasonic Lumix® DMC-FZ30 Digital Camera*. Le seul défi est d'établir la division de cette étiquette dans compagnie/produit/modèle, qui se fait facilement quand on a plusieurs étiquettes : les noms de produits et compagnies se répètent, et les noms de modèles sont normalement uniques aux produits concrets. Par exemple, *Powershot400* est unique à ce type d'appareils photo - d'autres compagnies n'appellent pas leurs modèles par ce nom, et il très improbable qu'un autre produit ait le même nom de modèle.

Pour accommoder des erreurs d'orthographe mineures, il faut accepter un certain flou : par exemple, *nocia* au lieu de *nokia*. Ceci peut être fait en utilisant la distance de Levenstein (edit distance) (Levenshtein 1966). S'il y a un nom dans la liste qui est moins loin que le seuil (dans le sens de distance Levenstein) d'un mot dans la requête, on suppose que l'usager a fait une erreur de frappe ou d'orthographe. S'il y a eu plusieurs noms qui sont proches au mot, on prend celui qui est le plus proche ; s'il y a plusieurs qui sont également proches, on demande à l'usager de désambiguïser.

Pour ne pas frustrer l'usager, il faut permettre quelques synonymes et hyperonymes. Par exemple, *Nokia phone* devrait être converti dans *Nokia cellphone* automatiquement. Ceci exige une ontologie de produits, où, pour cet exemple, *cellphone* est un sous-noeud de (ou type de) *phone*. Cette information peut être trouvée dans WordNet : le synset de *cellphone*, par exemple, est *cellular telephone, cellular phone, cellphone, cell, mobile phone*. Les hyperonymes directs et indirects à un niveau sont : *radiotelephone, radiophone, wireless telephone* et *telephone, phone, telephone set*. Ayant reçu une requête avec *Nokia phone*, en regardant le triplet `Nokia cellphone 16`, on cherche à comprendre la relation entre *cellphone* et *phone*. Ayant trouvé que le dernier est l'hyperonyme du premier, on suppose que l'usager voulait dire *cellphone*.

Il se peut quand même que WordNet ne contienne pas d'information pour des produits qui viennent d'être créés. Dans ce cas, quelques noeuds peuvent être ajoutés à la main dans l'ontologie. Étant donné une ontologie établie, placer de nouveaux produits devrait être facile. Une avenue de travail futur à explorer est l'apprentissage automatique des hyperonymes parmi des noms de produits. Une possibilité est l'apprentissage de patrons où l'hyperonyme est mentionné explicitement : par exemple, *Of all my stero systems, I like this mp3 player the best.*

Idéalement, il faut aussi être capable d'identifier les produits, compagnies et modèles qui ne sont pas représentés dans la collection de commentaires donnée. Ceci permettrait d'informer l'usager que c'est le cas. Pour créer une telle liste, il faudrait adapter un identificateur des entités nommées à ce type de texte.

Pour le prototype, le module d'extraction des noms des compagnies/produits/modèles a été implanté pour le corpus HL, où le tableau de compagnies/modèles/produits a été créé à la main. En utilisant la distance de Levenstein, un flou est permis pour accommoder des fautes d'orthographe.

4.2 Identification des termes

Prochainement, nous voudrions apprendre les détails qui intéressent l'usager, et nous le ferons en identifiant les aspects des produits dans la requête. Identifier les aspects des produits mentionnés est très difficile sans une liste d'aspects possibles créée à l'avance. Les problèmes incluent :

- des termes complexes de plusieurs mots (*background colour*)
- la présence d'autres mots (*loud volume*)

Le premier est un problème de base avec la recherche par mots clés : le défi est de reconnaître les expressions et de ne pas chercher des phrases (ou documents) contenant juste une partie de l'expression. Par exemple, si l'usager s'intéresse à *background colour*, nous ne voulons pas extraire l'information sur *background* et *colour*, mais seulement sur l'expression entière.

La raison pour laquelle il est important d'enlever tous les mots qui ne sont pas des termes est que l'usager peut donner la caractéristique désirée (*loud volume*), mais pour lui donner une réponse complète, nous avons besoin de sortir les phrases qui caractérisent le volume autrement aussi (*low volume*)[2]. Ce problème est même plus grave si l'usager

[2]Enlever les adjectifs, qui semble être la solution pour l'exemple présenté, ne fonctionne pas à cause de termes qui contienne des adjectifs. Quelques exemples extraits des commentaires sont *navigational*

soumet une question en langue naturelle car ceci augmente le nombre de mots à éliminer.

Nous avons recherché l'extraction automatique des aspects des produits, le Chapitre 6 est dédié à ces expériences. Étant donné cette liste, il suffit de chercher chaque terme possible pour le produit (si le nom du produit a été identifié dans l'étape précédant) dans la requête. Dans notre expérience, le nombre d'aspects maximal était environ 800, ce qui ne pose pas de grands problèmes d'efficacité.

Ce module du prototype utilise l'ontologie dont la création est décrite dans les chapitres suivants. Après avoir cherché chaque terme dans la requête, on élimine les termes trouvés dans la requête qui font partie d'un autre terme aussi trouvé dans la requête. Par exemple, si la requête est *background color*, d'abord on trouve les termes *background*, *color*, et *background color* et ensuite on élimine *background* et *color*.

4.3 Analyse coopérative

Ce module est basé sur les travaux de (Benamara 2004), (Benamara & Saint-Dizier 2004) présentés au chapitre 2. En adaptant leur idée à notre domaine, notre but est de vérifier la consistance de la requête en fonction de produits, compagnies, modèles, et aspects de produits mentionnés. Les usagers peuvent avoir tort au sujet de :

- Quelles compagnies manufacturent (ou vendent) quels produits.
- Quels modèles existent pour chaque produit.
- Quelles compagnies manufacturent (ou vendent) quels modèles.

Des erreurs de ce type sont très probables étant donné la pluralité de produits, compagnies, et modèles sur le marché. L'alternative utilisée normalement sur les sites web est un système de menus à naviguer, qui fonctionne bien mais n'est pas aussi naturel pour les humains que juste donner leur description comme ils le feraient avec un autre humain capable de coopération.

4.3.1 Vérification de consistance

Nous utiliserons le tableau de triplets `compagnie, produit, modèle` que nous avons introduit à la section 4.1. Ce tableau contient l'information de quelles compagnies manufacturent quels produits et quels sont les modèles disponibles.

Étant donné le triplet extrait de la requête, on le compare à tous les triplets dans le tableau. Deux triplets sont équivalents si tous les champs respectifs sont égaux ou vides

system, *alphabetical order*.

(dans le cas où le triplet de la requête manque un élement). Par exemple, une comparaison possible est de `CellPhone, Nokia, 6610` (du tableau) à *CellPhone, NONE, 6610* (d'une requête). Ces triplets sont équivalents car les champs du produit et du modèle sont identiques, et il n'y a aucune mention de la compagnie dans la requête.

Si on trouve un triplet équivalent dans le tableau, on a vérifié que l'usager s'intéresse à un produit qui existe, sinon on suppose que l'usager a une supposition erronée.

triplet de la question	**Problème**	**triplets alternatifs**
CellPhone, Canon, NONE	Canon ne manufacture pas de téléphones cellulaires.	*Camera, Canon, NONE*; *CellPhone, Nokia, NONE*
CellPhone, Nokia, G3	G3 n'est pas un modèle d'un téléphone cellulaire Nokia.	*CellPhone, Nokia, 6610*
CellPhone, Canon, ZenXtra	Rien n'est pas assorti.	*CellPhone, Nokia, 6610*; *Camera, Canon, G3*; *MP3Player, NomandLabs, ZenXtra*

TAB. 4.1 – Trois scénarios des suppositions erronées.

4.3.2 Communication avec l'usager

Si on croit que l'usager a fait une supposition erronée, il faut lui présenter quelques alternatives et lui demander d'en choisir une. On utilise encore le tableau de triplets pour trouver des triplets les plus semblables à celui extrait de la requête. Si un membre du triplet de la requête n'est pas spécifié (*NONE*), nous ne spécifions pas les alternatives. Regardons trois exemples de suppositions erronées dans la table 4.1.

Dans le premier cas, l'usager a tort en pensant que la compagnie *Canon* manufacture des *téléphones cellulaires*. Nous cherchons les alternatives possibles pour le produit et la compagnie : les produits manufacturés par Canon et les compagnies qui manufacturent téléphones cellulaires. Les alternatives trouvées pour notre corpus HL sont présentées dans la troisième colonne.

Dans le deuxième cas, l'usager a tort en pensant qu'il y a un modèle *G3* de téléphone cellulaire Nokia, donc nous cherchons quels modèles existent. Dans le dernier cas, nous

voulons trouver toutes les alternatives pour chaque élément, ce qui génère beaucoup de possibilités.

Ayant obtenu la liste de triplets alternatifs, on communique avec l'usager avec un gabarit de texte prédéfini :

<Canon> does not make <CellPhones>. Alternatives :
Nokia CellPhone volume, *or*
Canon camera volume, *or*
start over.

4.3.3 Utilisation de termes

Cette dernière vérification est basée sur la supposition qu'il est plus probable que l'usager se trompe avec le produit/compagnie/modèle qu'avec les aspects auxquels il s'intéresse.

Dans l'exemple précédent, on a offert *Canon camera* comme alternative après une requête *Canon phone volume*. Ceci n'a pas de sens car *volume* n'est pas un aspect des appareils photo. Pour éviter ce genre de "coopération", on utilise les termes extraits de la requête pour éliminer les produits qui ne les ont pas comme aspects. Étant donné que la liste des aspects des produits est créée automatiquement et qu'on sait qu'elle n'est pas parfaite, au lieu d'éliminer les alternatives invalidées par cette vérification, on peut les mettre après celles qui ont réussi le test.

Pour le prototype, le module coopératif a été implanté sans texte prédéfini d'explications de suppositions erronées et sans utilisation de termes. En plus, on n'a pas implanté l'aspect interactif dans le cas où l'usager change sa requête. Par exemple, si l'usager entre *nokia mp3 player ...*, le prototype retourne tout simplement les alternatives : *nokia cellphone* et *creative labs nomad mp3 player*. Pour un corpus réel, l'accès à une liste de compagnies/produits/modèles est nécessaire dont la création a été discutée à la section 4.1.

4.4 Recherche des phrases pertinentes

Le triplet `compagnie, produit, modèle` et les termes comprennent l'information qu'on extrait de la requête. La prochaine étape est de sélectionner les phrases pertinentes de la collection de commentaires de consommateurs. Parce que nous travaillons avec des

phrases, nous perdons des données à cause des anaphores, mais ceci est un grand problème en TALN que nous n'adressons pas ici.

Premièrement, en utilisant le triplet, on choisit une sous-collection de commentaires (on suppose que les commentaires sont organisés comme dans le corpus HL ; ceci peut être fait automatiquement à la base de triplets extraits de chaque commentaire). Par exemple, s'il s'agit des appareils photo, on ne veut pas regarder les commentaires sur des téléphones cellulaires.

Si on n'a pas extrait de termes, on ne peut plus filtrer les phrases, donc on considère toutes les phrases dans la sous-collection de commentaires pertinents. Si ceci comprend plus qu'un seuil (par exemple, 1000) de mots, on avertit l'usager et lui propose d'ajouter des détails dans sa requête en affichant la liste de termes pour le produit dont il s'agit.

Par contre si nous avons extrait des termes, nous filtrons les phrases de la sous-collection. Dans notre exemple, *Nokia speakerphone*, le filtre est basé sur le mot *speakerphone*. Premièrement, nous créons un ensemble de termes qui comprend les termes de la requête et ses synonymes (détails dans chapitre 7). Puis nous cherchons toutes les phrases contenant au moins un terme de cet ensemble pour obtenir les phrases pertinentes.

La recherche de phrases pertinentes à base de termes extraits de la requête et de chaque phrase dans les commentaires est un processus dont la qualité du résultat dépend entièrement de l'ensemble de termes utilisé. Ayant trouvé ces phrases, on extrait les termes de chacune, y comrpis ceux qui n'étaient pas dans la requête.

Ce module a été implanté sans l'ajout de synonymes dans le prototype, parce que nos résultats de regroupement de termes présentés au chapitre 7 sont plutôt préliminaires.

4.5 Classification de phrases selon l'attitude

Nous supposons que ce module peut être créé à base de méthodes de classification de texte selon l'attitude présentée au chapitre 2. Ceci est une question ouverte de recherche, mais les meilleurs algorithmes peuvent déjà être utiles.

Pour le prototype, nous avons utilisé l'annotation du corpus HL. Dans cette annotation (présentée dans table 3.2), chaque caractéristique mentionnée est donnée un score positif ou négatif. Pour obtenir des étiquettes de phrases complètes, nous avons classifié comme positive/négative les phrases où toutes les caractéristiques avaient des étiquettes positives/négatives. Les phrases avec étiquettes positives et négatives ou sans étiquettes n'ont pas été classifiées.

4.6 Elimination de phrases redondantes

Étant donné l'attitude et les termes de chaque phrase, on peut décider pour chaque paire de phrases si elles sont redondantes. Elles le sont si :

- les attitudes sont les mêmes
- les ensembles de termes sont identiques[3]

ID	Phrase	Représentation ELIM
A	*this phone has a very cool and useful feature - the speakerphone .*	`speakerphone, +`
B	*the speakerphone works better than any speakerphone i 've ever had .*	`speakerphone, +`

TAB. 4.2 – Deux phrases de commentaires de consommateurs sur un téléphone Nokia et leurs représentations ELIM (élimination). L'attitude est représentée par + pour positive.

La Table 4.2 présente un exemple. La phrase B ne contient beaucoup information utile que la phrase A ne contient pas déjà. En appliquant les deux conditions de redondance, on les juge redondantes parce que les attitudes sont les mêmes et l'ensemble de termes de la première phrase contient celui de la deuxième[4].

Pour décider quelle phrase éliminer, on suit les règles suivantes :

- si une phrase a déjà été éliminée (à cause d'une autre phrase redondante), on la laisse éliminée et on garde l'autre phrase ;
- sinon, on garde la phrase plus longue.

Pour notre exemple, on élimine la phrase B parce qu'elle est plus courte.

Ce module a été implanté dans le prototype.

4.7 Organisation de phrases qui restent

Pour ordonner des phrases qui restent, on propose de suivre les trois heuristiques suivantes :

- Débuter avec les phrases redondantes avec le plus grand nombre de phrases éliminées.

[3]Une possibilité intéressante à explorer est changer cette condition à : l'ensemble de termes de l'une contient l'ensemble de termes de l'autre.

[4]Il faut admettre que cette notion de redondance est approximative, et que certains usagers préféraient voir les deux phrases dans le résumé final.

– Grouper les phrases selon leur attitude : on groupe les phrases en positives/négatives/ /inconnues, et on présente d'abord le groupe plus grand parmi positives/négatives, suivi par l'autre groupe parmi positives/négatives, suivi par le groupe de phrase avec attitude inconnue.

Une idée à explorer dans un travail futur est le découpage de phrases. Par exemple, souvent plusieurs aspects sont commentés à la fois : *The buttons, the keypad, the screen were all wonderful.* Pour pouvoir ne donner à l'usager que l'information qu'il cherche, il serait utile de découper cette phrase en *The buttons were wonderful.*, *The key were wonderful.*, et *The screen were wonderful.*. Comme un découpage simple introduit des erreurs de syntaxe, il faudrait les corriger automatiquement.

Pour le prototype, l'ordonnancement de phrases qui restent suit les deux heuristiques proposées ici.

4.8 Exemple de fonctionnement du prototype

Nous présentons maintenant une trace détaillé de l'execution du prototype sur notre exemple, avec commentaires de succès et problèmes à régler. Les sections de ce chapitre correspondantes à chaque étape sont indiquées entre parenthèses.

Requête *nokia speakerphone*

Triplet extrait (4.1) *nokia, NONE, NONE* (aucun mention du produit ni modèle)

Sections du corpus choisis (4.1) `nokia_cellphone_6610` (la seule section de la compagnie Nokia comprend 41 commentaires)

Termes extraits de la requête (4.2) en utilisant la liste de termes extraits automatiquement[5] par la méthodologie décrite dans le chapitre 6 : *speakerphone*, *speaker phone* (le dernier comme variante orthographique parce que la différence est qu'un espace).

Suppositions erronées (4.3) aucune

Phrases pertinentes (4.4) extraites à base de ces deux termes de la section du corpus `nokia_cellphone_6610` sont présentées à la table 4.3. La liste démontre bien la nécessité d'éliminer la redondance : plusieurs phrases n'ajoutent pas de nouvelle

[5]Deux termes extraits incorrectement (*phone* et *feature*) ont été enlevés à la main parce qu'ils créent trop de confusion.

information par rapport aux autres phrases. Elle montre aussi deux erreurs de segmentation en phrases dans le corpus HL : *the speakerphone :* et - *speakerphone* ; ce sont des morceaux d'énumerations faits par les auteurs de commentaires.

Termes extraits de phrases pertinentes (4.4) présentés dans la troisième colonne de la table 4.3 sont extraits de la même façon que de la requête. Les termes qui ne sont pas détectés sont en parenthèses ; des suggestions pour l'alimélioration de l'extracteur de termes sont présentées à la fin du chapitre 6.

Etiquettes d'attitude (4.5) sont présentées dans la table 4.3 dans la quatrième colonne. Phrases 3 et 18 semblent avoir une erreur d'annotation car les phrases sont clairement positives.

Phrases éliminées (4.6) Pour ces 18 phrases, 16 paires ont été trouvées redondantes et 6 phrases ont été éliminées. Les paires redondantes sont présentés à table 4.4. Comme on le voit dans la table 4.3, il y a deux groupes de phrases redondantes : 0 et 1 au sujet de *speaker phone* (dont 1 reste et 0 est éliminiée) ; et 5, 7, 10, 11, 13, 17 qui disent toutes quelque chose bon au sujet de *speakerphone* (dont la phrase 10 reste, et les autres sont éliminées).

Notez que *speakerphone*, *speaker phone*, *speakerphone option* et *speakerphone feature* ont été traités comme caractéristiques différents parce qu'ils sont des termes différents.

Ordonnancement (4.7) La phrase 10 a 5 phrases redondantes, donc elle est la première phrase dans le résumé. Phrase 0, qui a une phrase redondante, suit. Les autres phrases sont dans leur ordre original, en débutant par les positives suivies des négatives, et finalement de celles dont l'attitude n'est pas connue.

Résultat Le résumé final est présenté à la figure 4.2. Il y a plusieurs problèmes en plus de ceux déjà mentionnés (erreurs de segmentation en phrases, erreurs de classification selon attitude, et des termes non détectés). Le premier problème est ce que toutes les phrases sont à la première personne, ce qui est mélangeant. Le deuxième problème est celui des mentions d'autres caractéristiques (p.e. *poly-graphic megatones*, *phone book*) qui motive notre suggestion de découpage de phrases comme projet futur. Malgré ces problèmes, notre résumé est une alternative clairement supérieure à une liste simple de phrases contenant le mot *speakerphone*.

No.	Phrase pertinente	Répresentation ELIM Termes	A
1	**the speaker phone is very functional and i use the phone in the car , very audible even with freeway noise .**	speaker phone	+
0	*my favorite features , although there are many , are the speaker phone , the radio and the infrared .*	speaker phone (radio, infrarer)	+
10	**speakerphone - loud and clear has some nice extra features like currency converter and a stopwatch .**	speakerphone (currency converter, stopwatch)	+
5	*the speakerphone , the radio , all features work perfectly .*	speakerphone (radio)	+
7	*this phone has a very cool and useful feature – the speakerphone .*	speakerphone	+
11	*- speakerphone*	speakerphone	+
13	*the hands-free speakerphone is quite powerful (like the motorola phone , i used to own) .*	speakerphone	+
17	*the speakerphone works better than any speakerphone i 've ever had .*	speakerphone	+
2	**from the speakerphone that can be used up to 15 feet away with clarity , to the downloadable polygraphic megatones that adds a personal touch to this nifty phone .**	clarity, speakerphone (megatones)	+
3	**great speakerphone and great reception ! ! recco-mend !**	speakerphone (reception)	?
4	**the two biggest things is the excellent working speakerphone " unlike the nokia 3650 " and the superb reception nokia is known for in the gsm phones they make ...**	speakerphone, gsm (reception)	+
6	**the speakerphone :**	speakerphone	?
8	**only one complaint about the speakerphone , you can only activate the speakerphone feature once the person you are calling answers the phone , not while the phone is ringing .**	speakerphone feature (ringing)	-
9	**i 've used the speakerphone for almost two hours once and the battery did not even go down one single bar .**	hour, speakerphone (battery)	+
12	**its speakerphone option allows us to do long talks conveniently , lying on bed ; phone lying by your pillowside .**	speakerphone option	+
14	**the phone book is very user-friendly and the speakerphone is excellent .**	phone book, speakerphone	+
15	**the phone has great battery life , fm radio , excellent signal , hands free speakerphone (which i have to say is probably my favorite function) and downloadable java apps .**	battery life, signal, function, hand, speakerphone (java apps)	+
16	**i did a ton of research and settled on this phone because of the small size , speakerphone option , great priced calling plan and access to my corporate email .**	size, email, speakerphone option (calling plan)	+
18	**people i talk to on the speakerphone are shocked when the phone comes out at times that i 'm even using a speakerphone .**	using, speakerphone	?

TAB. 4.3 – Exemple de fonctionnement du prototype : les phrases pertinentes (présentées dans la forme qu'elles ont dans le corpus HL) avec ses termes extraits (les termes ne pas detectés sont en parenthéses) et étiquettes d'attitude (colonne "A") : + pour positive, - pour négative, ? pour inconnu. Les phrases qui restent après le module d'élimination sont en gras, suivies par les phrases éliminées à cause de redondance avec elles. Les phrases sont numérotées selon leur ordre dans le corpus.

Paire redondante		Phrase éliminée
0	1	0
5	7	5
5	10	5
5	11	11
5	13	5
5	17	5
7	10	7
7	11	9
7	13	7
7	17	7
10	11	11
10	13	13
10	17	17
11	13	11
11	17	11
13	17	17

TAB. 4.4 – Les phrases redondantes indiquées par les nombres qui leur ont été assigné à table 4.3. La dernière colonne indique laquelle a été éliminée.

Speakerphone - loud and clear has some nice extra features like currency converter and a stopwatch. My favorite features, although there are many, are the speaker phone, the radio and the infrared. The speaker phone is very functional and I use the phone in the car, very audible even with freeway noise. From the speakerphone that can be used up to 15 feet away with clarity, to the downloadable poly-graphic megatones that adds a personal touch to this nifty phone. The two biggest things is the excellent working speakerphone "unlike the Nokia 3650"and the superb reception Nokia is known for in the gsm phones they make ... I've used the speakerphone for almost two hours once and the battery did not even go down one single bar. Its speakerphone option allows us to do long talks conveniently, lying on bed ; phone lying by your pillowside. The phone book is very user-friendly and the speakerphone is excellent. The phone has great battery life, fm radio, excellent signal, hands free speakerphone (which I have to say is probably my favorite function) and downloadable java apps. I did a ton of research and settled on this phone because of the small size, speakerphone option, great priced calling plan and access to my corporate email. Only one complaint about the speakerphone, you can only activate the speakerphone feature once the person you are calling answers the phone, not while the phone is ringing. Great speakerphone and great reception ! ! reccomend ! The speakerphone : People I talk to on the speakerphone are shocked when the phone comes out at times that I'm even using a speakerphone.

FIG. 4.2 – Résumé final suite à la requête *Nokia speakerphone* (les majuscules et la ponctuation ont été corrigés automatiquement pour lisibilité).

Cet exemple de fonctionnement de notre prototype montre premièrement la nécessité d'éliminer des phrases redondantes en présentant la liste de phrases qui parlent de *speakerphone* sorties de juste 41 commentaires sur un téléphone Nokia. Ensuite, il montre que l'architecture proposée est capable d'en faire une grande partie : un tiers des phrases extraites à base de la requête sont éliminées. Les problèmes et les avenues de travail futurs correspondants ont été mentionnés. On voit donc le grand potentiel de l'architecture proposée à résoudre le problème de résumé de commentaires de consommateurs[6].

4.9 Comparaison avec les travaux précédents

Des travaux précédents de traitement de commentaires de consommateurs ont été présentés au chapitre 2. L'application que nous proposons ici diffère de travaux précédents en quatre façons essentiels :

- Nous donnons à l'usager l'opportunité de choisir l'aspect du produit auquel il s'intéresse via la requête.
- Nous proposons un module coopératif pour analyser des suppositions erronées.
- Nous éliminons les phrases redondantes parmi les phrases pertinentes extraites à la base de la requête.
- Nous présentons à l'usager un résumé en langue naturelle (contrairement à un résumé structuré de Hu&Liu).

Pour le dernier point, notre supposition est que si les usagers voulaient voir des évaluations numériques pour les aspects des produits, c'est tout ce qu'on demanderait des consommateurs qui écrivent les commentaires. Parce que ce n'est pas le cas, nous supposons que les usagers qui consultent les commentaires préfèrent voir des textes. Il serait intéressant de vérifier ceci par un sondage des utilisateurs des commentaires de consommateurs.

Ceci conclut la présentation de l'application visée. Dans le reste de ce mémoire, nous nous concentrons sur la création semi-automatique de l'ontologie du domaine, la ressource principale nécessaire pour le fonctionnement d'une telle application.

[6]Certaines personnes croient que pour le système existant, présenter toutes les phrases comme au Table 4.3 est plus utile pour l'usager que le résumé présenté à la Figure 4.2. Ceci améliore l'usabilité du système dans son état actuel, mais notre but générale est d'être capable de résumer des ensembles des textes, donc c'est la qualité de résumés finales qui nous intéresse le plus.

Chapitre 5

Ontologie du domaine

5.1 Motivation

Au chapitre 1, nous avons mentionné le domaine de représentation de connaissances qui traite des problèmes de création d'une représentation cohérente, consistante, et utilisable en raisonnement. Une ontologie est un type de représentation de connaissances populaire. Il y a plusieurs types et même plusieurs définitions différentes d'une ontologie. Une ontologie peut être définie comme "les spécifications explicites et formelles de termes d'un domaine et des relations entre ces termes (Gruber 1993)" (Noy & McGuinness 2001). Essentiellement, c'est un arbre de concepts avec des propriétés, où les enfants d'un noeud héritent de ses propriétés. Par exemple, le noeud `Animal` peut avoir des sous-noeuds `Mammifère`, `Oiseaux`, etc, et la propriété `peut bouger` qui est héritée par les sous-noeuds ; le noeud `Oiseaux` peut avoir la propriété `peut voler` qui sera héritée par ses sous-noeuds. L'ensemble des concepts et propriétés forment la description du domaine en question.

(Noy & McGuinness 2001) décrivent l'essence de ce qu'est une ontologie, ses avantages et comment les construire manuellement[1]. Il existe plusieurs outils qui assistent les ingénieurs d'ontologie dans leur travail en donnant un environnement graphique pour la construction et plusieurs aides - par exemple, la vérification de la consistance d'ontologies.

La construction (semi-)automatique des ontologies est couramment un domaine de recherche actif pour deux raisons principales :

– La vision du Web Sémantique est basée sur l'existence d'ontologies pour plusieurs domaines.

[1]Il est important de ne pas les confondre avec des ressources lexicales. Les noeuds des ontologies sont des concepts ; les noeuds de ressources lexicales sont des mots. Des fois les deux sont identique mais souvent ils ne le sont pas : voir (Hirst 2004) pour une discussion détaillée.

– La construction manuelle des ontologies est très exigeante et lente.

Le Web Sémantique est une initiative pour rendre les documents plus faciles à traiter automatiquement en ajoutant des marqueurs sémantiques aux textes. Le développeur principal de cette vision est Tim Berners-Lee du *World Wide Web Consortium* et l'ensemble des textes visé est le Web (Antoniou & van Harmelen 2004). La vision du Web Sémantique est basée sur un type particulier d'ontologies qui s'appelle *OWL* (Web Ontology Language). Ce type est différent des autres types d'ontologies à cause de quelques restrictions sur les types de relations permises. Nous avons respecté ces restrictions dans la création de notre ontologie du domaine.

5.2 Partie manuelle

A cause de notre domaine très spécifique, plusieurs éléments de notre projet ne portent trop à confusion : produits, consommateurs, compagnies, services, etc. Nous avons créé cette partie d'ontologie manuellement en utilisant un outil de création d'ontologies *Protégé*[2]. Dans cette section, nous présentons les parties manuelles ; l'appendice A présente l'ensemble de ces parties. Dans la prochaine section nous discuterons les parties que devraient être créées automatiquement.

Dans notre ontologie, nous avons mis les six noeuds directement sous la racine *Thing (Chose)* pour représenter ce qui est important dans notre domaine :

`Human`

`Mental`

`Company`

`CompanyAspect`

`Product`

`ProductAspect`

L'arbre sous le noeud `Mental` inclut les `Opinions` et les `Sentiments`. Comme nous l'avons mentionné au chapitre 2, pour le moment les classifications selon l'attitude sont normalement en fonction de positif/négatif/neutre. Avec le développement de cet aspect, cette partie peut être agrandie. Pour le moment, ses sous-noeuds sont :

`Excellent`

`Average`

[2] "Protégé is a free, open source ontology editor and knowledge-base framework" (http ://protege.stanford.edu/)

```
Poor
```

pour `Opinion`, et pour `Sentiment` :

```
Content
Neutrality
Discontent
```

L'arbre sous `Company` décrit les deux types de compagnies pertinentes au domaine : celles qui fabriquent des produits (`Manufacturer`) et celles qui les vendent (`Distributor`) ; les deux ne sont pas nécessairement différentes (ceci est spécifié dans l'ontologie en indiquant que ces concepts ne sont pas disjoints).

L'arbre sous `CompanyAspect` contient deux aspects simples des compagnies : `CustomerService` et `Delivery`. Les noeuds sous `Human` sont séparés en deux types : `Customer` et `Employee`.

La partie sous `Product` liste les produits pour notre agent : pour le moment, ceci inclut seulement des produits électroniques pertinents étant donné notre corpus de commentaires. Les commentaires au sujet des téléphones est un cas spécial parce que le service de la compagnie est normalement évalué aussi. Nous n'avons pas inclu ce service explicitement dans notre ontologie parce qu'il n'y a pas d'ensembles de commentaires séparés au sujet du service, mais certains aspects (par exemple `Long distance plan`) seront mis comme aspects du téléphone[3].

```
Product
    ElectronicProduct
        Camera
        CellPhone
        DVDPlayer
        MP3Player
```

Jusqu'ici, nous avons présenté l'ontologie sauf les sous-arbres de `ProductAspect`. Ceci devrait contenir une description de produits en termes de leurs caractéristiques. Premièrement, les caractéristiques diffèrent selon le type de produit, même s'il y a des caractéristiques communes à tous les produits :

```
ProductAspect
    CameraAspect
```

[3]Pour l'inclure comme un concept diffèrent, il faudrait avoir un concept de `ce qu'on peut acheter`, qui aurait `service` et `produit` comme enfants.

Human hasMental Mental	Mental isOf Human
Product boughtBy Consumer	Consumer hasBought Product
Product isManufacturedBy ManufacturerCompany	ManufacturerCompany manufactures Product
Product isSoldBy SellerCompany	SellerCompany sells Product
Employee isEmployedBy Company	Company employs Employee
Mental pertainsTo Product OR ProductAspect OR CompanyAspect	
Product hasAspect AllProductsAspect	AllProductsAspect isAspectOf Product
Camera hasAspect CameraAspect	CameraAspect isAspectOf Camera
CellPhone hasAspect CellPhoneAspect	CellPhoneAspect isAspectOf CellPhone
DVDPlayer hasAspect DVDPlayerAspect	DVDPlayerAspect isAspectOf DVDPlayer
MP3Player hasAspect MP3PlayerAspect	MP3PlayerAspect isAspectOf MP3Player

TAB. 5.1 – Les propriétés lient les concepts dans l'ontologie.

CellPhoneAspect

DVDPlayerAspect

MP3PlayerAspect

AllProductsAspect

Les propriétés définissent les relations entre les concepts ; nous en avons défini onze qui sont présentées dans la table 5.1.

En plus de ces concepts et propriétés, les ontologies contiennent des *instances* : ceux-ci sont des références aux objets concrets dans le monde (au moins les plus concrets dans l'ontologie donnée). Dans notre cas, les instances incluent les compagnies de notre corpus (avec l'information sur les produit qu'elles fabriquent ou vendent) et les produits (qui sont identifiés par les modèles).

5.3 Partie automatique

Nous avons découvert que compiler une liste de caractéristiques même pour seulement quatre types de produits est un processus exigeant. Nous avons essayé d'utiliser l'annotation dans le corpus HL et les descriptions de produits sur les sites web de compagnies qui les ont produits pour nous guider, mais ce n'était pas très productif. Essentiellement, à moins de lire tous les commentaires, on ne peut pas être certain que la liste est complète. Ceci est prohibitif, en particulier si on a plusieurs produits. Nous avons donc pensé à créer les listes de caractéristiques automatiquement. Le prochain chapitre est dédié à la description de notre méthodologie d'extraction automatique de caractéristiques ; le chapitre suivant présente notre essai de regrouper les termes extraits.

Chapitre 6

Extraction de terminologie

Au chapitre 4, nous avons présenté trois raisons expliquant qu'une liste de caractéristiques (aspects) des produits est essentielle pour résumer des commentaires de consommateurs. On les rappelle ici :

- l'importance de préparer une liste d'aspects des produits a priori pour l'analyse de requêtes.
- la représentation de phrases par l'attitude et un ensemble de termes mentionnés.
- le regroupement de phrases pertinentes selon les termes qu'elles contiennent.

Comment obtenir les listes de caractéristiques ? Même si la liste des aspects de chaque produit électronique n'est pas très longue, à cause de l'xutilisation des synonymes (y compris phrases synonymes) et de la quantité des produits différents, il n'est pas possible de créer à la main une liste complète des noms de caractéristiques pour chaque produit auquel on s'intéresse[1]. Il faut donc penser à les extraire automatiquement. Ce problème est un cas particulier d'extraction de terminologie du domaine.

Nous avons décidé de procéder en deux étapes : premièrement extraire les termes, puis les grouper par synonymie (par exemple, *scroll button* et *scroll wheel*). Ce chapitre est dédié à la première étape et le chapitre suivant traite la deuxième.

Quelques travaux présentés au chapitre 2 ont déjà essayé d'extraire et organiser les caractéristiques des produits électroniques. Ici, nous présentons notre nouvelle méthodologie.

[1]Nous l'avons vérifié en essayant les créer à la main pour ensuite abandonner assez frustrés. Nous avons aussi essayé d'utiliser les listed fournies par les sites web des compagnies qui ont créé le produit, mais ces listes de caractéristiques varient trop dans le niveau de détail accordé et dans la forme d'expression (*Ringing tones composed by award winning composer Ryuichi Sakamoto* vs. *Device-to-device synchronization*). Il faudrait donc traiter ces listes automatiquement pour extraire les caractéristiques pures, et il nous semblait plus intéressant de traiter les commentaires eux-mêmes parce qu'ils contiennet tous les synonymes et termes que les consommateurs utilisent.

6.1 Patrons des étiquettes

Notre approche se base sur le fait qu'il y a des patrons dans la façon dont les consommateurs mentionnent les noms des caractéristiques. Par exemple, "the <caractéristique> is great(<description>)" ou "I liked the <caractéristique>" sont deux types de phrases fréquentes. Parce que créer une liste de patrons comme celles-ci à la main serait fastidieux ou même prohibitif, il faut penser à les apprendre automatiquement. Notre approche exige donc un ensemble d'entraînement, ce qui veut dire que pour un produit, il faut identifier manuellement les caractéristiques dans quelques commentaires car d'autres caractéristiques que celles annotées pourront par la suit être identifiées.

Ceci vaut la peine parce qu'il nous permet d'extraire automatiquement les caractéristiques d'autres produits. Dans notre expérimentation, nous montrons comment notre méthode extrait les caractéristiques d'autres produits électroniques. Nous supposons que la méthode peut être utilisée avec d'autres types de produits (par exemple, des livres) sans changement de l'ensemble d'entraînement. Ceci est basé sur l'observation que quelque soit le produit, les façons d'exprimer l'avis sur un objet acheté sont les mêmes. Par exemple, dans un commentaire sur un livre, on pourrait voir la phrase *The plot is delightfully twisted.*, qui suit le patron "the <caractéristique> is <description>".

L'avantage de notre approche en comparaison avec celles que nous venons de présenter est sa simplicité :

- nous n'utilisons qu'une analyse superficielle du texte ;
- nous utilisons un corpus plus grand que HL mais mieux défini que tout le Web ;
- notre approche est plus intuitive que la technique de Web PMI et "association mining" introduits aux sections 2.3.3 et 2.3.1.

6.2 L'extracteur de terminologie

Nous avons utilisé un extracteur de terminologie pour notre expérimentation. C'est un extracteur de terminologie créé par (Patry & Langlais 2005) qui fonctionne à base de patrons des parties du discours. Ceci veut dire qu'avant de chercher des patrons, le texte est converti en une liste d'étiquettes grammaticales. Un avantage est que même si l'étiqueteur ne fonctionne pas toujours bien (assigne des fois des mauvaise étiquettes), il est consistant, donc la recherche de patrons peut réussir malgré ses fautes.

En bref, cet extracteur prend un corpus de commentaires et une liste des termes créés à la main qui enseigne le programme ce que nous considérons être des termes. Tous les

deux (la liste de termes et le corpus) sont lemmatisés et étiquetés (indépendamment). Toutes les occurrences des termes dans le texte sont utilisées[2]. Un modèle de langue est ensuite entraîné, où les mots sont les étiquettes. Ce modèle est utilisé pour extraire les candidats de termes d'un corpus différent (que celui d'entraînement). Puis un classificateur entraîné par AdaBoost (Schapire 1999) à la base de plusieurs caractéristiques courantes dans le domaine de l'extraction de terminologie (longueur, fréquence dans le corpus, etc) est utilisé pour trier les candidats selon la probabilité qu'ils soient des vrais termes. Chaque terme extrait reçoit donc deux scores principaux : un du modèle de langue et un d'AdaBoost.

Nous supposons que les patrons d'utilisation des noms des caractéristiques sont les mêmes pour tous les produits électroniques et nous créons un seul modèle ("de langue") pour les produits dont nous nous occupons.

6.3 L'ensemble d'entraînement

Pour créer le modèle de langue, on a besoin de fournir un ensemble des termes et un texte où ils se trouvent. Originalement, nous pensions utiliser l'annotation dans le corpus de Hu&Liu, mais après un examen manuel, nous avons réalisé que leur annotation comportait trop de bruit pour nos besoins.

Voilà quelques exemples de bruit (selon la notation de la Table 3.2). Même si les erreurs d'annotation sont inévitables, parce que notre ensemble d'entraînement est petit et les termes annotés sont recherchés partout dans le corpus, elles causent facilement du bruit dans les résultats.

- *this thing[+3]##other than that this thing is great ...* est une erreur de ne pas avoir ajouté *[u]* pour indiquer qu'une résolution de référence est nécessaire
- *##the batteries would not charge .*, où le terme n'est pas identifié
- *manual[-3]##also , the instruction manual is very bad .*, où le terme est identifié mais n'est pas complet (*instruction manual* est le terme complet)

En plus, (Hu & Liu 2004a) n'ont pas annoté que les caractéristiques qui n'étaient pas commentés dans la même phrase. Des phrases comme *##case - the nomad comes with a leather holding case .* étaient ignorées. Ceci est malheureux pour nous car ces phrases peuvent suivre des patrons utiles (par exemple, *comes with a <caractéristique>*.

[2]Ceci n'est pas toujours opportun pour nous car nos termes sont des mots assez communs (en comparaison à la terminologie botanique ou technique), et il se peut que dans quelques contextes, ils ont un sens différent que le terme envisagé.

Pour résoudre ces problèmes, nous avons extrait à la main un ensemble de termes en utilisant le corpus HL, la section du lecteur des mp3s. Nous avons regardé 777 phrases et nous avons extrait 221 termes (présentés dans l'appendice B). Nous avons comparé cette liste (designée "termes FL") avec celle produite par l'extraction de l'annotation de cette section du corpus (désignée "termes HL"). Termes HL comprend 181 termes extraits de 1811 phrases annotées. Il est donc clair que notre extraction est plus détaillée. Le chevauchement entre les deux listes (HL et FL) est de 68 termes.

Comme nous l'avons mentionné au chapitre précédent (3.2), il y a trois complications dans l'annotation de termes pour ce corpus. Nous avons décidé de les régler comme ceci : ni les références au produit lui-même, ni les verbes, ni les adjectifs ne sont des termes. Seuls les syntagmes nominaux sont cherchés. Nous avons pris cette décision parce que (a) les références au produit lui-même apparaissent dans plusieurs contextes et empêchent la recherche des patrons (b) nous avions une autre idée sur la façon d'identifier les verbes et adjectifs pertinents à la base des syntagmes nominaux (elle sera présentée à la section 6.7).

6.4 Ajouter le contexte

On peut utiliser l'extracteur de terminologie de deux façons : en entraînant directement sur les termes (*battery life*), ou en incluant leur contexte (quelques mots autour du terme : *the battery life is*, *battery life is amazing*). Nous avons expérimenté avec les deux, et nous avons trouvé que notre intuition de départ était correcte : c'est le contexte des termes qui aide à les extraire. Le contexte peut prendre plusieurs formes, mais l'idée de base est d'inclure des mots et la ponctuation autour des termes.

Un avantage du contexte spécifique à l'extracteur de terminologie que nous avons utilisé est qu'il permet d'améliorer les étiquettes des termes. Par exemple, le terme *use* tout seul sera annoté comme un verbe, mais avec du contexte (par exemple, *ease of use*), il est annoté correctement comme un nom. Un travail futur intéressant serait de changer l'extracteur de terminologie pour qu'il étiquette les termes en utilisant leurs contextes dans le corpus et ne pas utiliser le contexte dans l'extraction de termes. Ceci permettrait comprendre si c'est l'amélioration de l'étiquetage ou la complexité élevée des patrons qui améliore la précision finale.

Si on fournit à l'extracteur des termes avec leurs contextes, la sortie de l'extracteur de terminologie doit être nettoyée : le contexte doit être enlevé. Par exemple, si on a utilisé les deux mots suivant chaque terme comme contexte (*battery life is great* de la

Exemples d'entraînement fournis à ET (*terme* avec contexte)	Sortie d'ET		Sortie d'ET nettoyée	
	Terme	Score	Terme	Score
battery life is great	*bluetooth* is active	x	*bluetooth*	$\frac{x+y}{2}$
buttons don't work	*bluetooth* doesn't work	y		

TAB. 6.1 – Exemple démontrant les étapes de l'extraction de terminologie. L'extracteur de terminologie de (Patry & Langlais 2005) (ET) fonctionne à la base des étiquettes, donc *great* et *active*, et *don't work* et *doesn't work* sont identiques pour lui. Ses sorties sont nettoyés : le contexte est enlevé, et les scores sont combinés.

phrase *The battery life is great, you can listen to music all day.*), l'extracteur va aussi sortir des termes avec contexte (*bluetooth set is active*), où nous avons enlevé le contexte (*is active*). La liste nettoyée comme ceci contient des doublons (par exemple, *bluetooth doesn't work* donne aussi *bluetooth* comme terme après nettoyage) qui sont ensuite enlevés. En éliminant les doublons, la moyenne des scores originaux était prise comme le score du nouveau terme unique. La table 6.1 clarifie cet exemple.

Le type de contexte préférable était le sujet de notre expérimentation. Nous avons expérimenté avec l'utilisation de deux mots directement avant et après des termes (MOT TERME MOT[3]), deux mots directement après des termes (TERME MOT MOT), et la combinaison (MOT TERME MOT MOT). La forme de contexte qui fonctionnait le mieux était TERME MOT MOT ; les autres n'étaient pas assez sélectifs (MOT TERME MOT) ou trop sélectif (MOT TERME MOT MOT). Cette solution est probablement spécifique à notre domaine.

6.5 Expériences

Pour les expériences, notre ensemble d'entraînement comprenait :

– une liste de termes pour les lecteurs de mp3 (HL, 181 termes, et FL, 221 termes)

– la section du corpus HL qui contient les commentaires sur les lecteurs de mp3.

La précision rapportée est basée sur l'évaluation manuelle d'un juge (l'auteur), qui regardait manuellement tous les termes extraits et les marquait corrects ou incorrects. Il n'y avait qu'un juge, donc la précision est approximative, mais elle permet quand même de choisir la meilleure méthode d'extraction des caractéristiques des produits.

Pour valider la nécessité de la création d'ensemble d'entraînement FL, nous avons fait quelques expériences avec l'ensemble HL. La table suivante résume notre expérience avec

[3]Où MOT peut être aussi ponctuation.

les deux ensembles de termes.

Ensemble d'entraînement (contexte)	**# de termes extraits**	**Précision approximée**
HL (aucun)	environ 12,000	30% (dans les premiers 685 termes)
HL (MOT TERME MOT)	environ 430	55%
HL (TERME MOT MOT)	environ 810	65% (dans les premiers 60 termes), 53% (dans le premiers 200 termes)
HL (MOT TERME MOT MOT)	53	85%
FL (MOT TERME MOT)	environ 1200	beaucoup de bruit
FL (TERME MOT MOT)	**environ 800**	**80%**
FL (MOT TERME MOT MOT)	26	88%

TAB. 6.2 – Les expériences d'extraction de terminologie.

La première section de la table rapporte notre expérience avec l'ensemble d'entraînement HL. En entraînant sans contexte, nous avons reçu une grande quantité des termes avec beaucoup de bruit. La précision dans les premiers 685 termes était de seulement 30% (nous avons regardé 685 termes parce que les scores ont changé sensiblement à ce point-là). En ajoutant le contexte, la précision a été améliorée, mais pas suffisamment : 55% avec MOT TERME MOT, 65% avec TERME MOT MOT[4].

La deuxième section de la table présente notre expérience avec l'ensemble d'entraînement FL. Ayant déjà vu que l'ajout de contexte améliore bien les résultats, nous n'avons pas expérimenté sans contexte. Avec le contexte MOT TERME MOT, nous avons obtenu 1200 termes. En examinant à la main les patrons appris, nous les avons trouvé trop généraux : les patrons du genre *the <caractéristique> of/has/for* dominaient. Un examen informel de la liste des termes extraits a confirmé qu'elle contenait beaucoup de bruit.

Avec les autres types de contexte, les résultats étaient bien meilleurs. Avec TERME MOT MOT, nous avons obtenu environ 800 termes avec une précision de 80%. Cette précision était stable : elle était à peu près la même pour n'importe quel nombre de premiers termes. Avec le type de contexte MOT TERME MOT MOT, nous avons obtenu la meilleure précision (parmi nos expériences) de 88%, mais le rappel était très bas : seulement 26 termes ont été extraits.

Nous avons conclu que la méthode qui fonctionne le mieux est le contexte TERME MOT MOT avec les termes FL. Ceci nous a donné 802 termes, qui est un nombre assez

[4]Ayant vue que la précision diminue de 65% dans les premiers 60 termes à 53% dans les premiers 200 termes, nous avons décidé qu'il ne vaut pas la peine procéder avec l'évaluation - ce niveau de précision est loin de ce qui est suffisant.

grand, mais qui est expliqué par la présence des termes dans différents contextes locaux. Par exemple, *plastic case*, *silver case* et *case* sont tous dans la liste. En regardant les patrons appris informellement, nous avons vu que les patrons du genre TERME *is/has* ADJ dominaient.

6.6 Les scores

Nous avons examiné l'utilité des scores en faisant des graphiques qui montrent la relation entre le rang d'un terme (axe horizontal) et son score (axe vertical). Nous avons trouvé que les courbes sont lisses, sans sauts intéressants. Nous avons conclu que les scores ne peuvent pas être utilisés pour filtrer la liste de termes sortie pas notre algorithme.

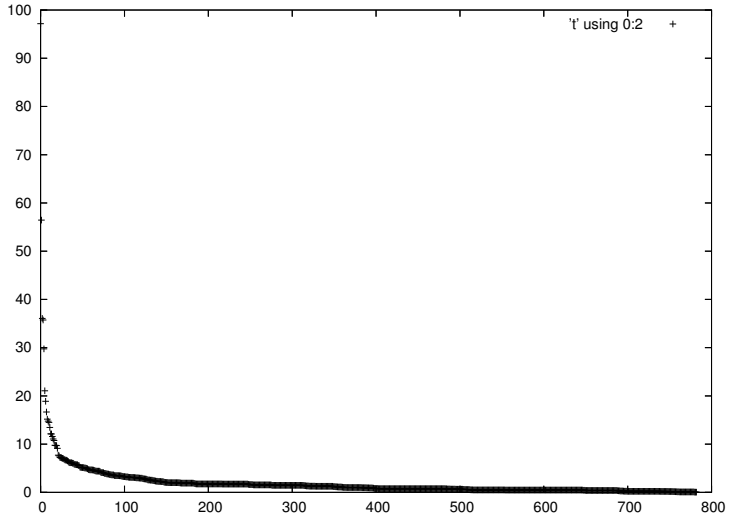

FIG. 6.1 – Rang des termes (X) vs. leur score AdaBoost (Y).

6.7 Mentions implicites de caractéristiques

Une fois la liste de caractéristiques des produits obtenue, nous l'utilisons dans l'application (chapitre 4) pour chercher ces termes dans le texte de commentaires et dans les requêtes. Mais souvent les caractéristiques sont mentionnées implicitement (des exemples ont été présentés dans section 3.2). Ceci est un grand problème à cause de la multiplicité des façons avec lesquelles on peut référer à une caractéristique. Ici nous présentons une partie de la solution.

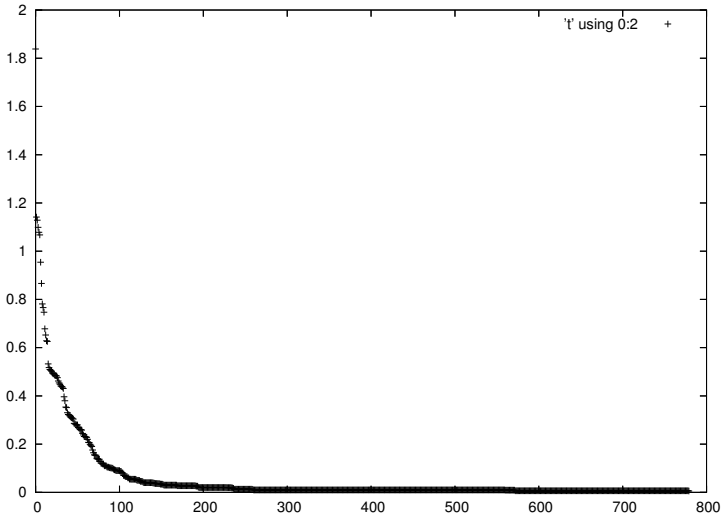

FIG. 6.2 – Rang des termes (X) vs. leur score Automaton (Y).

Parce que les caractéristiques sont souvent mentionnées indirectement, il est désirable de savoir quels adjectifs sont pertinents pour chaque terme. Cette information se trouve dans les ressources lexicales. Nous avons exploré l'utilité de WordNet pour ce problème. Dans WordNet, les adjectifs sont parfois liés au nom via une relation `attribute`. Les synonymes de ces adjectifs peuvent être trouvés via la relation `similar to` (cette relation est spécifique aux adjectifs).

La liste d'adjectifs pertinents aux noms dans les termes de téléphones cellulaires est présentée à la Table 6.3. Les sens d'adjectifs ne sont pas présentés mais les relations sont définies entre des sens concrets des noms et adjectifs. Il faut mentionner le problème de désambiguïsation entre, par exemple, le nom `light` et l'adjectif `light` ; nous laissons les détails d'utilisation de ces adjectifs pertinents aux travaux futurs.

6.8 Conclusion

Ceci conclut la présentation de notre expérience avec l'extraction de termes des commentaires de consommateurs. Ces termes forment la partie principale de l'ontologie du domaine en décrivant les aspects important des produits en question. Nous avons montré une méthode à base de patrons qui donne une précision d'environ 80%. Le temps de calcul était moins qu'une demi-heure pour l'entraînement et l'extraction. Il reste à tester cette méthode sur d'autres produits ; un test sur un type de produit très différent

Nom	Adjectifs (`attributes`)
use	functional, nonfunctional
function	functional, nonfunctional
volume	loud, soft
quality	good, bad, positive, negative ; superior, inferior
size	large, small
color	colored, uncolored ; colorful, colorless
alarm	alarming, unalarming
clarity	clear, unclear
strength	delicate, rugged, strong, weak ; potent, impotent
speed	fast, slow
weight	heavy, light
connection	connected, unconnected
light	bright, dull ; light, dark

TAB. 6.3 – Les noms de termes pour téléphones cellulaires avec ses attributs. Les point-virgules séparent les adjectifs pertinents aux sens différents du nom en question. Les attributs de *size*, *weight*, et *volume* sont particulièrement utiles.

que l'électronique serait plus convaincant. Pour faire cette expérience, il ne faut qu'un ensemble de commentaires et une personne pour évaluer le résultat.

L'autre question importante est si 80% est suffisant pour que le système de résumé au complét fonctionne bien. Nous avons présenté un exemple de fonctionnement du système au complet au chapitre 4, mais la qualité générale de résumés produits reste à evaluer. Si ce niveau de qualité de l'extracteur des caractéristiques n'est pas suffisant, il y d'autres types de patrons à tester. Par exemple, souvent les auteurs de commentaires énumèrent les aspects qu'ils ont aimés (*The speakerphone, the radio, all features work perfectly.*) ; ceci peut être utile dans une deuxième itération avec les patrons comprenant les termes déjà découverts (par exemple, si de la première iteration on sait que *speakerphone* est un terme, on peut déduire que *radio* l'est aussi).

Chapitre 7

Regroupement de termes

Au chapitre précédent, nous avons présenté une méthode d'extraction automatique de caractéristiques de produits. Le nombre de termes extraits a varié selon le produit entre 450 et 800 termes. Dans ce chapitre, nous présentons un essai d'organisation de cette masse de termes pour les rendre plus utilisables. Il y a trois raisons pour regrouper les termes extraits, deux spécifiques à notre application et une autre plus générale :

- On veut connaître les termes semblables pour répondre aux requêtes des usagers. Par exemple, si l'usager s'intéresse à la caractéristique *screen*, on voudrait aussi trouver les phrases traitant de *display*.
- On veut connaître les relations plus générales entre les termes pour ordonner les phrases extraites. Par exemple, les phrases traitant de *messaging* seraient mieux placées à côté de celles au sujet de *email* plutôt que *screen*.
- Le but plus général est le développement d'une méthodologie pour la création d'une description d'un objet comme un élément de connaissances du monde qui peut être utilisé dans plusieurs applications. C'est un cas d'apprentissage de descriptions des humains : parce que la machine n'a pas les sens (sens tactile, ouïe, etc) nécessaires pour apprendre ce qu'est un téléphone cellulaire, elle apprend de textes qui le décrivent.

7.1 Base de travaux précédents

Dans chapitre 2, nous avons présenté les travaux de (Hu & Liu 2004a) et (Carenini et al. 2005). (Hu & Liu 2004b) utilisent WordNet très conservativement (seulement l'information de synsets) pour grouper quelques termes. (Carenini et al. 2005) projettent un ensemble de termes sur une taxonomie prédéfinie après avoir annoté les sens des mots à

la main. Notre but était d'améliorer la méthodologie de regroupement de termes de trois façons :

- Eliminer l'utilisation de taxonomies créées manuellement, parce qu'elles exigent beaucoup d'effort manuel et qu'elles n'existent encore pas pour tous produits.
- Utiliser WordNet moins conservativement que Hu&Liu pour profiter de connaissances disponibles dans cette ressource lexicale.
- Automatiser la désambiguïsation de sens (WSD).

Nos expériences comprennent le regroupement de termes par mots clés du domaine (première étape), le regroupement de groupes et l'ajout d'autres termes dans les groupes en utilisant des mesures de similarité sémantique (deuxième étape), et la désambiguïsation automatique de sens de termes (qui assiste la deuxième étape). Dans notre utilisation des mesures de similarité sémantique, nous nous basons sur les expériences de Carenini et al. Le résultat est un regroupement raisonnable, facile à corriger (pour un humain), et utilisable dans l'application proposée au chapitre 4. Comme au chapitre précédent, nous présentons les résultats pour les termes se rapportant aux téléphones cellulaires.

7.2 Regroupement par mots clés du domaine

??

Premièrement, nous voulions profiter du fait que plusieurs termes avaient des mots en commun. Néanmoins, tous les mots communs ne sont pas une bonne base de regroupement. Nous voulions donc identifier les mots clés pertinents au domaine qui peuvent être utilisés pour un premier regroupement de termes. Pour décrire ce qu'on considère comme mot clé, regardons deux paires de termes :

- *signal quality* et *signal strength* - on voudrait grouper ces termes parce qu'ils traitent d'un aspect important : *signal*.
- *signal quality* et *image quality* - on ne voudrait pas grouper ces termes parce qu'ils parlent d'aspects très différents (donc *quality* n'est pas un mot clé du domaine).

Pour découvrir cet ensemble de mots, nous avons utilisé la différence entre les fréquences de termes dans un corpus général (Hansard) et dans notre corpus de commentaires (FL). Hansard est une grande collection de débats parlementaires ($6,660,123$ phrases)[1].

A chaque mot dans le corpus FL a été attribué un score (d'ici, "score hansard") égal à $\frac{frequencedansFL}{frequencedansHansard}$. On a jugé tous les mots avec un score supérieur à un seuil (0.5)

[1]C'est un corpus souvent utilisé en traduction automatique parce qu'il y a une version française parallèle.

déterminé empiriquement comme représentatifs du domaine. Il y avait 242 mots avec un score supérieur au seuil, et 759 avec un score inférieur.

Parmi les 242 mots avec un score élevé, il y en a eu 100 qui apparaissaient dans au moins un terme. Ceci est comparable au nombre de noeuds dans les taxonomies utilisées par Carenini et al. Le nombre total de mots différents dans les termes est 344, donc environ 29% ont été choisis et le reste était éliminé.

Parmi les 100 mots clés choisis, 73 sont bons (selon une évaluation manuelle effectuée par une personne), mais quelques-uns sont problématiques. Trois groupes de mots clés problématiques méritent un commentaire :

- Les noms de modèles et compagnies (5) - ceux-ci peuvent être éliminés avec une liste de compagnies et de modèles du domaine électronique (discuté au chapitre 4).
- Les adjectifs (5) - ceux-ci sont les mots qui ne sont que adjectifs ; ils peuvent être éliminés très facilement (par exemple, en utilisant WordNet).
- Les mots pertinents mais très vagues : par exemple, *feature*, *function*, *device*[2]. Ceux-ci sont quand même faciles à identifier (et à éliminer) pour un humain.

7.2.1 Regroupement de termes de téléphone cellulaire

Pour le regroupement, on a créé 100 groupes, chacun avec les termes contenant le mot clé correspondant. Chaque terme pouvait être mis dans plusieurs groupes (s'il est complexe). Nous avons traité les pluriels dans les mots clés et les termes.

434 (ou environ 60%) de 720 termes de téléphones cellulaires ont été groupés. Tous les groupes sont présentés dans l'appendice C. Voilà deux exemples pour démontrer les bons et mauvais groupes :

- Un bon groupe est un groupe qui décrit un seul aspect du produit, par exemple : *charger (mot clé), extra charger, desktop charger, charger connector, car charger, top charger, charger device, base charger, charger input, travel charger*
- Un mauvais groupe est celui qui ne le fait pas, par exemple : *silver (mot clé), silver plastic, silver design*

La variation du seuil pour la sélection de mots clés a comme effet un changement du nombre de groupes créés (plus haut est le seuil, plus petit est le nombre de groupes) et la qualité de mots clés (plus haut est le seuil, le plus grande est la quantité de bons mots

[2]Notre essai d'éliminer les mots à base de leur fréquence dans le corpus de tous les commentaires vs. dans le corpus du produit en question n'étaient pas positive parce qu'il y avait d'autres mots pertinents avec les patrons de fréquences pareils : par exemple, *screen* (un aspect de la plupart de produits électroniques).

clés). Le seuil optimal peut varier pour différents domaines et même différents produits. Choisir le seuil optimal automatiquement est une avenue à explorer dans les travaux futurs.

7.3 Co-occurrences des termes

Clairement, les groupes de base décrits à la section précédente ne sont pas suffisants même si ceux à la base de faux mots clés seraient éliminés, parce que ces groupes ne présentent pas une description cohérente de caractéristiques du produit. Deux problèmes essentiels sont :

- Plusieurs groupes reliés ne sont pas liés (par exemple, ceux de *screen* et *display*).
- Plusieurs termes ne sont pas dans aucun groupe même s'il y a un groupe logique pour eux (par exemple, *net access* n'est pas dans le groupe de *internet*).

Il faut donc d'autres sources d'information pour améliorer ces groupes.

Deux sources que nous avons considérées sont les co-occurrences de termes dans le corpus et WordNet. Les co-occurrences de termes sont, dans la forme la plus simple, une liste de mots qui apparaissent le plus souvent à côté du terme en question. Souvent les co-occurrences portent beaucoup d'information sur la similarité de termes (Dagan 2000), mais dans le cas de notre corpus, nous avons trouvé que cela n'est pas le cas.

Dans nos expériences préliminaires, nous avons utilisé les algorithmes de clustering de WEKA[3]. Le termes étaient représentés par un vecteur des nombres de co-occurences avec chaque mot qui est une co-occurrence fréquente d'un terme. Ici, fréquente signifie parmi les dix co-occurrences les plus fréquentes. Les clusters résultants n'étaient pas du tout prometteurs, nous avons donc abandonné les co-occurrences comme source d'information.

Cet échec n'était pas étonnant : les commentaires de consommateurs contiennent beaucoup de références aux caractéristiques de produits, le vocabulaire utilisé n'est pas très riche, donc le fait que les termes co-occurrent tous avec à peu près les mêmes mots n'est pas une grande surprise. Il valait quand même la peine de faire quelques expériences préliminaires pour valider cette intuition.

Un autre argument contre l'utilisation de co-occurrences dans le traitement de commentaires de consommateurs est présenté par (Carenini et al. 2005) qui rappelle qu'il est difficile de compiler un corpus suffisamment grand pour que les co-occurrences soient utiles.

[3]WEKA est un logiciel de plusieurs algorithmes d'apprentissage machine. Nous avons expérimenté avec les algorithmes suivants : EM, K-means, et Cobweb.

7.4 Similarité sémantique via WordNet

A priori, il n'est pas clair que WordNet peut être utile car c'est une ressource générale et non spécifique à notre domaine. Néanmoins, comme il n'y pas de ressources spécifiques à notre domaine, et parce que nous préférons des méthodologies portables à d'autres domaines, nous voulions enquêter sur l'utilité de WordNet. En plus, (Carenini et al. 2005) ont utilisé des mesures de similarité à base de WordNet avec un certain succès.

Le premier pas était de vérifier à la main que pour la majorité de mots de notre domaine, le sens correct est dans WordNet. En fait, WordNet contient une quantité étonnante de termes spécifiques à notre domaine, par exemple *pda* (personal digital assistant). Quelques cas problématiques sont causés par les abréviations : *tones* qui est une contraction de *ringtone* ; quelques autres ne le sont pas : *ringtone* n'est pas dans WordNet non plus. Néanmoins, nous avons conclu que la couverture de nos termes pour les téléphones cellulaires par WordNet est suffisante pour procéder à des expériences.

7.4.1 Désambiguïsation de sens (WSD)

Contrairement à (Carenini et al. 2005) qui a annoté les sens de mots dans ses termes à la main, nous avons automatisé ce processus. A cause de notre expérience avec les co-occurrences, nous n'avons pas suivi (McCarthy, et al. 2004) qui présente une méthode de désambiguïsation pour domaines spécifiques à base de co-occurrences (présenté dans le chapitre 2).

Algorithme

Notre méthode de désambiguïsation est simpliste, mais fonctionne quand même assez bien. Nous avons utilisé un ensemble de mots qui définissent, d'une certaine façon, le domaine de chaque produit. C'est-à-dire qu'étant donné cet ensemble de mots, un humain dirait sans hésitation qu'il s'agit d'un tel produit (par exemple, téléphone cellulaire). Encore une fois nous avons utilisé les scores hansard ; cette fois en prenant 100 mots avec les scores le plus hauts (100 étant un chiffre arbitraire qui nous semblait raisonnable), et en choisissant ceux qui n'ont qu'un sens dans WordNet. Il y en a eu 12 qui nous semblent bien représentatifs :

keypad
speakerphone
pda (personal digital assistant)
phonebook
headset

handset
faceplate
voicemail
email
earpiece
messaging
modem

Nous avons supposé que tous les mots dans les termes sont des noms (ce qui est certainement une simplification). L'algorithme de désambiguïsation est le suivant :

```
Pour chaque mot M avec plusieurs sens:
    Pour chaque sens S de M:
        Pour chaque mot W parmi les 12 mots représentatifs:
            score(S) = score(S) + sem_rel(S,W);
Retourner S avec le score(S) maximal.
```

Évaluation

Pour évaluer cet algorithme, nous avons désambiguïsé à la main les 125 mots clés qui ont un score hansard (défini à la section **??**) supérieur à 0.2 (ceci a été effectué par une personne). Il faut noter que la désambiguïsation n'est pas toujours évidente même pour un humain parce que comme on en a discuté au chapitre 2, les sens sont parfois très proches ou même se chevauchent. Quand plusieurs sens semblaient raisonnables, on a choisi celui qui était le plus fréquent. Le nombre de sens pour cet ensemble de mots varie entre 2 et 12.

Avant de présenter les résultats d'évaluation, voilà une liste de mots pour lesquels il n'y avait pas de sens approprié dans WordNet : *scroll, lightweight, ringer, navigation, tone, pad*. Les deux premiers sont problématiques seulement à cause de notre supposition que tous les mots sont des noms. Les deux derniers sont des abréviations (pour *ringtone* et *keypad*).

La précision de cet algorithme est comparable à celles rapportées dans (McCarthy et al. 2004) ; la Table 7.1 présente les résultats. Pour les expériences qui suivent, nous avons utilisé les sens pour ces 125 mots et la désambiguïsation automatique pour tous les autres mots.

7.4.2 Algorithmes de regroupement et mapping

Rappelons notre but principal dans cette deuxième étape de regroupement :

Mesure de similarité	Précision de WSD
path	51%
res	47%
lin	55%

TAB. 7.1 – La précision de notre algorithme pour la désambiguïsation de sens de mots basé à mesures de similarité sémantique différents sur 125 mots annotés manuellement.

- Trouver des mots très proches sémantiquement parmi les mots clés et combiner leurs groupes (par exemple, ceux de *screen* et *display*).
- Projeter les termes non groupés sur un groupe en utilisant ses similarités aux autres termes et aux mots clés (par exemple, *net access* dans le groupe de *internet*).

Nous avons expérimenté avec trois mesures de similarité sémantique (introduits dans chapitre 2) : **res** et **lin** parce qu'elles avaient été trouvées les plus utiles par (Carenini et al. 2005), et **path** parce que c'est la mesure qui utilise l'information dans WordNet le plus directement. Pour chaque mesure, nous avons calculé la similarité sémantique pour chaque paire de termes et chaque paire {terme, mot clé}. En suivant Carenini et al., pour les termes complexes, nous avons pris comme score de similarité la moyenne des scores de toutes les paires de mots dans les deux termes. Contrairement à Carenini et al., nous avons vérifié si le terme complexe existe comme tel dans WordNet (par exemple, *operating system* est là) ; dans ce cas le terme peut être traité comme un terme simple.

Nous avons trouvé que la similarité est assez haute pour beaucoup de paires. Ceci est raisonnable car les termes viennent tous d'un domaine spécifique commun. A cause de ceci, le seuil de similarité à utiliser pour des regroupements devrait être très haut. Par exemple pour **path** (la mesure de distance sémantique égale à la distance entre les noeuds dans le graphe de relations), le seuil est 0.4 qui correspond à la distance de 2.5 liens entre les noeuds (où les noeuds sont les synsets de mots dont la similarité est mesurée). Pour un seuil un peu plus bas (trois liens), la précision diminue dramatiquement. Pour éviter cette diminution sans perdre des paires intéressantes, nous avons expérimenté avec la combinaison de deux mesures qui prennent des approches différentes : **res** et **path**.

Il faut porter une attention spéciale au traitement de paires de termes identiques : on veut leur donner un score haut, mais on ne veut pas que des termes complexes soient groupés parce qu'ils partagent un mot non pertinent (on a évité ceci dans la première étape en utilisant les scores hansard). Pour **res** et **lin**, nous avons pris les scores donnés par la mesure ; ils varient parce que le IC (Information Content, 2.1.3) est différent pour chaque noeud dans WordNet. Pour **path**, qui donne toujours le même score haut aux paires des mots identiques, on n'a pas traité des paires ou un terme est une sous-chaîne

de l'autre (donc effectivement leur score était zero).

Ayant calculé la similarité pour toutes les paires, l'algorithme de regroupement consiste des règles suivantes (dans cet ordre ; où S est un seuil déterminé empiriquement) :

- Si deux mots clés ont une similarité supérieure à S, joindre les groupes de ces mots clés.
- Si un terme non groupé a une similarité supérieure à S avec un mot clé, l'ajouter dans le groupe de ce mot clé.
- Si un terme non groupé a une similarité supérieure à S avec un terme groupé, l'ajouter dans les groupes de ce terme groupé.
- Si deux termes non groupés ont une similarité supérieure à S, créer un nouveau groupe avec ces deux termes.

Plusieurs itérations sont nécessaires pour maximiser le regroupement. L'algorithme arrête après une itération où aucun terme n'a changé. Le temps de calcul, y compris le calcul de distances sémantiques qui est la partie lente, est moins que 3 heures.

7.4.3 Résultats

Nous présentons ici les regroupements faits à la base de scores de similarité sémantique en utilisant **res**, **lin**, **path**, et **path** et **res** en combinaison. L'utilité de chaque regroupement proposé est un peu subjective. Par exemple, faut-il regrouper *email* et *messaging*? Ils sont semblables mais ce sont des fonctions séparées dans les téléphones. Aussi, pour les faux positifs (mots mal-identifiés comme mots clés), il est difficile de juger les regroupements proposés parce qu'on n'a pas un seul groupe possible en esprit. Nous présentons donc tous les regroupements proposés et notre jugement d'utilité.

7.5 Discussion

Les résultats présentés aux Tables 7.2 - 7.5 montrent que notre méthode propose des regroupements prometteurs et utiles. Ils ne sont pas suffisants pour la construction automatique d'une ontologie, mais ils peuvent offrir un bon début à un ingénieur humain. Quelques regroupements remarquables sont ceux de *software* et *operating system*, *screen* et *display* et *lcd*, *keypad* et *keyboard*, etc. Les problèmes sont causés dans la plupart des cas par des mots trop généraux comme *device*, *list*, *use*, etc.

On pourrait envisager l'utilisation d'un algorithme de clustering plus sophistiqué que la comparaison de scores avec un seuil. L'information de similarité entre les termes, même

ceux plus éloignés, devrait être utile. Par exemple, le fait que *device* est proche de plusieurs termes pourrait signaler que c'est un mot trop général. Ceci est une avenue de travail futur qui exige une mesure de similarité sémantique assez précise, donc l'autre avenue de travail importante est le développement de mesures appropriées - à base de WordNet et d'autres ressources.

TAB. 7.2 – Le traitement de groupes de base proposé par notre algorithme en utilisant la mesure de similarité **res**. Le jugement est soit 1 (bonne idée), soit 0 (mauvaise idée). Sur les 20 paires proposés, 12 (60%) nous semblent bonnes.

Type de regroupement	Paires proposées			
	Bonnes paires		Mauvaises paires	
Joindre groupes	microphone	headset	switch	dial
	keypad	keyboard	button	dial
	lcd	screen	cell	speakerphone
	lcd	display	phone	speakerphone
	switch	button		
	screen	display		
	web	internet		
	cell	phone		
Ajout d'un terme (à la gauche) dans un groupe à base de similarité avec un mot clé (au droit)	hand	indicator	set	wireless
	caller id	display	sound	tone
	phone book	phonebook	long distance	voicemail
	memory	store	call	voicemail

TAB. 7.3 – Le traitement de groupes de base proposé par notre algorithme en utilisant la mesure de similarité **lin**. Le jugement est soit 1 (bonne idée), soit 0 (mauvaise idée). Sur les 13 paires proposés, 9 (69%) nous semblent bonnes.

Type de regroupement	Paires proposées			
	Bonnes paires		Mauvaises paires	
Joindre groupes	switch	button	wireless	reception
	screen	display		
Création de nouveaux groupes	purchase	purchasing		
	listing	list		
Ajout d'un terme (à la gauche) dans un groupe à base de similarité avec un mot clé (au droit)	choices	favorite	address	software
	user	customer	call	reception
	phone book	phonebook		
	memory	store		
	use	usage		
Ajout d'un terme (à la gauche) dans un groupe à base de similarité avec un terme (au droit) dans le groupe			long distance	call

TAB. 7.4 – [Le traitement de groupes de base proposé par notre algorithme en utilisant la mesure de similarité **path**. Le jugement est soit 1 (bonne idée), soit 0 (mauvaise idée). Sur les 55 paires proposés, 37 (67%) nous semblent bonnes.

Type de regroupement	Paires proposées			
	Bonnes paires		Mauvaises paires	
Joindre groupes	keypad	keyboard	device	charger
	switch	button	device	alarm
			device	indicator
			device	adapter
			device	keyboard
Création de nouveaux groupes	purchase	purchasing		
	address	address book		
	clarity sound	sound clarity		
	interior panel	inside panel		
	conversation clarity	talk clarity		
	capabilities	picture capability		
	long distance	call		
Ajout d'un terme (à la gauche) dans un groupe à base de similarité avec un mot clé (au droit)	use quality	usage	radio feature	wireless
	accessory list	accessories	listing	calendar
	net access	internet	listing	menu
	caller id	display	list	calendar
	battery use	usage	list	menu
	car adaptor	adapter	radio kit	wireless
	choices	favorite	charger device	adapter
	internet use	usage	charger device	keyboard
	phone use	usage	device	adapter
	voice mail	voicemail	device	keyboard
	operating system	software	radio reception	wireless
	web use	usage		
	power adaptor	adapter		
	phone book	phonebook		
	memory	store		
	mobile phone	cell		
	memory space	store		
	memory card	store		
	use	usage		
	accessory attachment	accessories		
	digital display	lcd		
	speakerphone capability	capabilities		
Ajout d'un terme dans un groupe à base de similarité avec un terme	message system	voice message	video quality	voice quality
	contact list	listing	input system	message system
	number list	listing		
	audio clarity	audio quality		
	name list	listing		
	audio quality	voice quality		

TAB. 7.5 – Le traitement de groupes de base proposé par notre algorithme en utilisant les mesures de similarité **path** et **res**. Le jugement est soit 1 (bonne idée), soit 0 (mauvaise idée). Sur les 37 paires proposés, 23 (62%) nous semblent bonnes.

Type de regroupement	Paires proposées			
	Bonnes paires		Mauvaises paires	
Joindre groupes	email	messaging	battery	antenna
	keypad	keyboard	battery	plug
	lcd	display	software	interface
	switch	button	calendar	menu
	screen	display	switch	dial
	web	internet	antenna	plug
	cell	phone	phone	speakerphone
			lock	clip
Création de nouveaux groupes	purchase	purchasing	mail support	message
	worth	true worth		
	charge	charge time		
Ajout d'un terme (à la gauche) dans un groupe à base de similarité avec un mot clé (au droit)	caller id	display	switch	dial
	battery use	usage	call	voicemail
	use	usage	list	menu
	internet use	usage		
	user	customer		
	phone use	usage		
	voice mail	voicemail		
	web use	usage		
	phone book	phonebook		
	memory	store		
	memory card	store		
Ajout d'un terme (à la gauche) dans un groupe à base de similarité avec un terme (au droit) dans le groupe	operating system	software	address	software
	card	memory card	long distance	call

Chapitre 8

Des requêtes aux questions

Dans le système décrit au chapitre 4, l'usager soumet ses demandes d'information sous la forme de requêtes. Cette forme est utilisable mais n'est pas la plus naturelle pour les humains ; en particulier, elle n'est pas aussi intuitive que poser des questions en langue naturelle. Les questions sont quand même beaucoup plus difficiles à analyser que des requêtes. Dans ce chapitre, nous décrivons notre travail qui vise à avancer la méthodologie d'analyse des questions.

8.1 Les défis d'analyse de questions

Pour un humain qui veut obtenir de l'information, la façon la plus naturelle d'interagir avec la source d'information est de poser des questions. Une question exige une analyse beaucoup plus élaborée qu'une requête parce qu'elle contient plus d'information sur les besoins de l'usager. L'avantage est donc la possibilité de lui donner une réponse plus adéquate à ses besoins que ce qui est possible avec des mots clés.

L'analyse d'une question est un grand problème de recherche à cause de deux raisons principales :

- Les humains ne savent pas préciser ce qu'ils veulent savoir. Par exemple, pour répondre à *Who makes better cameras, Sony or Panasonic ?*, il faudrait déterminer que c'est l'information sur les aspects des caméras qui sont les plus importants pour cet usager. Il faut donc désambiguïser et approfondir les demandes, idéalement via un dialogue avec l'usager, semblable à celui qu'on aurait avec un bibliothécaire.
- Souvent les humains ne posent pas juste une question : ils donnent une description de la situation, puis ils demandent de l'aide. La requête d'information peut même ne pas contenir une question propre. Par exemple,

I've tentatively settled on getting a Powershot s3 because I realized the biggest class of pictures that I can't take involves subjects that are too small or distant. But I've been thinking more about the sensitivity issue. I don't really need 5 or 6 MP if I could have, say 3 or 4 with higher sensitivity. And I mean usable ISO, not just the fact that the camera takes a picture at a certain ISO. People complain about the noise in the Powershot s2, and the s3 is supposedly better, but are there cameras in the 12x-zoom class that were designed more for sensitivity and less for high MP numbers ?

Pour le moment, les méthodes d'analyse de questions ne sont pas suffisamment avancées pour traiter ce genre de questions. La plupart des études travaillent avec des questions factuelles (*Where do turtles live ?*). Souvent les méthodes comprennent beaucoup d'effort manuel, par exemple la création de patrons pour classifier les questions en fonction du type de réponse. Quelques classes possibles sont *Personne* (*Who was the first to fly to the moon ?*), *Lieu* (*Where is Montreal ?*), etc. Connaître le type de réponse facilite la recherche de la réponse dans un texte source (p.e. le web). Pour les questions factuelles, la recherche de réponses est basée sur les techniques d'extraction et classification d'entités nommées.

8.2 Classification de questions

Un but intermédiaire dans le domaine de QR est de classifier les questions automatiquement, sans patrons créés à la main. Nous avons étudié la performance d'algorithmes d'apprentissage machine qui ont démontré leur efficacité dans d'autres domaines : machines avec vecteurs de support, réseaux de neurones et boosting. Nous avons utilisé les n-grammes pour caractériser les questions, une méthode simple et portable à plusieurs autres langues. Notre ensemble de données était 2000 questions factuelles que nous avons annotées à la main en notant pour chaque question son type de réponse correct parmi les huit classes possibles[1]. Pour vérifier la portabilité de la méthode, nous avons expérimenté aussi avec leur traduction en français. Nous avons utilisé la validation croisée pour les tests. La précision obtenue était d'environ 80% pour les deux langues. Ceci est pire que les patrons manuels (90%) ou que certaines méthodes semi-automatiques, mais impressionnant étant donné la simplicité des n-grammes.

Dans l'article (Feiguina & Kégl 2005) (en anglais) donné en appendice D, nous avons décrit notre expérience, des travaux précédents, et des travaux futurs.

[1]Les huit classes sont : *explanation, living entity, numeric, location, person definition, specification, definition, synonym.*

Chapitre 9

Travaux futurs et conclusion

9.1 Travaux futurs

Les expériences présentées dans ce mémoire suggèrent plusieurs explorations futures, ce qui comprend des évaluations humaines et des expériences additionnelles. Ici, nous énumérons quelques avenues de travail intéressantes, en commençant avec celles qui exigent le moins d'effort étant donné les outils implantés au cours de ce mémoire.

9.1.1 À court terme

Premièrement, il serait intéressant d'explorer la portabilité des méthodes que nous avons développées pour l'extraction de terminologie à des domaines moins spécifiques. On pourrait commencer avec un ensemble de produits plus grand et varié, tel que tous les produits du site web *epinions.com*. Cette expérience n'exige qu'un nouvel ensemble de commentaires (facile à créer en utilisant le crawler) et quelqu'un pour évaluer les termes et regroupements sortis.

Deuxièmement, parce que notre méthode n'utilise qu'une analyse syntaxique superficielle, il serait aussi assez facile d'explorer la portabilité de méthodes que nous avons développées pour l'extraction de terminologie à d'autres langues : des collections de commentaires de consommateurs existent aussi sur des sites web francophones, etc. La seule addition exigée est l'intégration d'un autre étiquetteur (français, etc).

Finalement, il serait intéressant d'évaluer (à la main) la technique d'élimination de phrases redondantes proposée au chapitre 4. Ceci exige un classificateur de phrases selon l'attitude qui pourrait peut-être être obtenu de chercheurs qui y travaillent.

9.1.2 À moyen terme

Ensuite il serait intéressant de faire une évaluation humaine du système complet : créer des résumés et demander aux humains de trouver les bouts d'information dans les commentaires originaux qui ne sont pas dans le résumé mais qui devraient y être, et les bouts d'information pas vraiment utiles dans le résumé.

Tester notre méthodologie de classification de questions factuelles sur questions plus complexes est une expérience qui prendrait plus d'effort surtout parce que ceci exige la création d'une collection de questions/requêtes d'autres types.

9.1.3 À long terme

Les grandes avenues de travail futur dans le traitement de textes exprimant des opinions sont le traitement de texte informel (la plupart d'opinions sont exprimées dans des blogs, etc, où aucun niveau de formalité n'est pas exigé) et le traitement plus approfondi de l'information dans les commentaires.

Le traitement de texte informel est un problème en soi car le texte informel comprend toujours des fautes d'orthographe et de grammaire. Plusieurs travaux se sont concentrés sur les premières, mais peu de travail a été fait pour la gestion des erreurs de grammaire. D'un autre côté, au lieu de corriger les erreurs, on peut viser à développer des méthodes qui sont robustes à ces erreurs.

Pour une analyse plus approfondie des commentaires de consommateurs, on peut se concentrer sur le type de question le plus exigeant qui est la comparaison. Il y a deux sources d'information qu'on pourrait utiliser. Les comparaisons faites directement dans le texte sont assez rares, au moins dans le cas des commentaires de consommateurs. Il faut donc que le système soit capable de chercher l'information au sujet des choses à comparer et produise une comparaison automatiquement. Ceci est une avenue très intéressante parce qu'elle correspond aux besoins réels des usagers qui cherchent à comparer des produits pour *choisir un* produit parmi eux.

9.2 Conclusion

Dans ce mémoire, nous avons présenté notre expérience de développement d'un système de résumé automatique de textes exprimant des opinions. En nous basant sur un type de texte spécifique, les commentaires de consommateurs, et un domaine restreint, l'électronique, nous avons proposé l'architecture d'une application de résumé automatique. Nous

avons implanté un prototype qui utilisait une ontologie du domaine construite semi-automatiquement. Ce processus incluait l'extraction automatique de caractéristiques de produits et comportait des expériences avec le regroupement par similarité sémantique de ces caractéristiques. Nous avons aussi exploré la classification automatique de questions en fonction du type de réponse et proposé quelques avenues de travail futur.

Les contributions de ce mémoire sont les suivantes : la proposition d'un premier système de résumé des commentaires de consommateurs en langue naturelle, une nouvelle méthode d'extraction automatique des caractéristiques de produits (via des patrons lexicaux) avec une bonne précision d'environ 80%, un premier essai de regroupement de ces caractéristiques sans taxonomie prédéfinie en utilisant la ressource lexicale de WordNet, et des expériences en classification automatique des questions factuelles en fonction de type de réponse avec une précision de 80% tant en anglais et qu'en français.

Nous croyons que ce travail pourrait déboucher sur un système de résumés de commentaires de consommateurs plus utiles et accessibles. Ceci constitue un premier pas vers un système général de résumé de plusieurs textes exprimant des opinions sur un sujet.

Bibliographie

G. Antoniou & F. van Harmelen (2004). *A Semantic Web Primer*. MIT Press.

P. Beineke, et al. (2004). 'The Sentimental Factor : Improving Review Classification Via Human-Provided Information'. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pp. 263–270. Association for Computational Linguistics.

F. Benamara (2004). 'Cooperative Question Answering in Restricted Domains : the WEBCOOP Experiment'. In D. M. Aliod & J. L. Vicedo (eds.), *ACL 2004 : Question Answering in Restricted Domains*, pp. 31–38, Barcelona, Spain. Association for Computational Linguistics.

F. Benamara & P. Saint-Dizier (2004). 'Construction de réponses coopératives : du corpus à la modélisation informatique'. In *Revue Québéquoise de linguistique*, no. 3 in 18, pp. 34–59.

G. Carenini, et al. (2005). 'Extracting knowledge from evaluative text'. In *Proceedings of the 3rd international conference on Knowledge capture (K-CAP 2005)*, pp. 11–18, New York, NY, USA. ACM Press.

I. Dagan (2000). *Contextual Word Similarity*, chap. 19. Marcel Dekker Inc.

K. Dave, et al. (2003). 'Mining the peanut gallery : opinion extraction and semantic classification of product reviews'. In *Proceedings of the 12th international conference on World Wide Web (WWW 2003)*, pp. 519–528, New York, NY, USA. ACM Press.

O. Etzioni, et al. (2005). 'Unsupervised named-entity extraction from the web : an experimental study'. *Artificial Intelligence* **165**(1) :91–134.

O. Feiguina & B. Kégl (2005). 'Learning to Classify Questions'. In *Proceedings of Computational Linguistics in the North-East (CLiNE). www.crtl.ca/cline*05/*cline*05_*papers*/*FeiguinaKegl.pdf*.

M. Gamon (2004). 'Sentiment classification on customer feedback data : noisy data, large feature vectors, and the role of linguistic analysis'. In *Proceedings of Coling 2004*, pp. 841–847, Geneva, Switzerland. COLING.

G. Hirst (2004). 'Ontology and the Lexicon'. In S. Staab & R. Studer (eds.), *Handbook on Ontologies*, pp. 209–229. Springer, Berlin.

G. Hirst & D. St-Onge (1998). *WordNet : An electronic lexical database*, chap. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. Christiane Fellbaum (editor), Cambridge, MA : The MIT Press.

M. Hu & B. Liu (2004a). 'Mining and Summarizing Customer Reviews'. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2004)*, pp. 168–177, New York, NY, USA. ACM Press.

M. Hu & B. Liu (2004b). 'Mining Opinion Features in Customer Reviews'. In *Proceedings of the 19th national conference of the American Association for Artificial Intelligence*, pp. 755–760, San Jose, USA.

N. Kobayashi, et al. (2004). 'Collecting Evaluative Expressions for Opinion Extraction'. In *Proceedings of Natural Language Processing – IJCNLP 2004, First International Joint Conference*, pp. 596–605, Hainan Island, China.

V. I. Levenshtein (1966). 'Binary codes capable of correcting deletions, insertions, and reversals'. *Soviet Physics Doklady* **10**(8) :707–710.

B. Liu, et al. (2005). 'Opinion observer : analyzing and comparing opinions on the Web'. In *Proceedings of the 14th international conference on World Wide Web (WWW 2005)*, pp. 342–351, New York, NY, USA. ACM Press.

H. Liu, et al. (2003). 'A model of textual affect sensing using real-world knowledge.'. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pp. 125–132, Miami, FL. ACM Press.

D. McCarthy, et al. (2004). 'Finding predominant senses in untagged text.'. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pp. 280–288, Barcelona, Spain.

S. Morinaga, et al. (2002). 'Mining product reputations on the Web'. In *Proceedings of the Eighth ACM SIGKDD Internatinal Conference on Knowledge Discover and Data Mining*, pp. 341–349, Edmonton, Alberta, Canada.

N. Noy & D. McGuinness (2001). 'Ontology development 101 : A guide to creating your first ontology.'. Tech. Rep. KSL-01-05, Stanford Knowledge Systems Laboratory.

A. Patry & P. Langlais (2005). 'Corpus-Based Terminology Extraction'. In *Proceedings of the 7th International Conference on Terminology and Knowledge Engineering*, pp. 313–321, Copenhagen, Denmark.

S. Patwardhan, et al. (2003). 'Using measures of semantic relatedness for word sense disambiguation.'. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 241–257, Mexico City, Mexico.

Pedersen, et al. (2004). 'WordNet : :Similarity - Measuring the Relatedness of Concepts'. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pp. 1024–1025, San Jose, CA, USA.

R. Picard (1997). *Affective Computing*. MIT Press.

A.-M. Popescu & O. Etzioni (2005). 'Extracting Product Features and Opinions from Reviews'. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 339–346, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

R. E. Schapire (1999). 'A brief introduction to boosting'. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*.

P. Turney (2002). 'Thumbs Up or Thumbs Down ? Semantic Orientation Applied to Unsupervised Classification of Reviews'. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia, Pennsylvania, USA*, pp. 417–424. Association for Computational Linguistics.

J. Wiebe, et al. (2005). 'Annotating expressions of opinions and emotions in language.'. *Language Resources and Evaluation (formerly Computers and the Humanities)* **39**(2-3) :165–210.

T. Wilson, et al. (2004). 'Just how mad are you ? Finding strong and weak opinion clauses.'. In *Proceedings of 19th National Conference on Artificial Intelligence (AAAI-2004)*, San Jose, CA, USA.

H. Yu & V. Hatzivassiloglou (2003). 'Towards answering opinion questions : separating facts from opinions and identifying the polarity of opinion sentences'. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 129–136, Morristown, NJ, USA. Association for Computational Linguistics.

Annexe A

Ontologie du domaine : partie manuelle

```
# owl:Thing

    * Company
          o Distributor  (2 instances)
          o Manufacturer  (5 instances)

    * CompanyAspect
          o CustomerService
          o Delivery

    * Human
          o Consumer
          o Employee

    * Mental
          o Sentiment
                + Pleasure
                + Neutrality
                + Displeasure
          o Opinion
                + Excellent
                + Average
                + Poor
```

```
* Product
      o ElectronicProduct
            + Camera  (2 instances)
            + CellPhone  (1 instance)
            + DVDPlayer  (1 instance)
            + MP3Player  (1 instance)

* ProductAspect
      o AspectOfCamera
            + ...
      o AspectOfCellphone
            + ...
      o AspectOfDVDPlayer
            + ...
      o AspectOfElectronicProduct
            + ...
      o AspectOfMP3Player
            + ...
```

Les propriétés sont présentées dans Table 5.1 dans chapitre 5.

Annexe B

221 caractéristiques manuelles

221 caractéristiques de lecteurs de mp3 extraites à la main de 777 phrases du corpus HL :

accessibility to online music services
accessories
accessory
adapter unit
affordability
album order
alphabetical order
audio aspects
audio quality
backlight
backlit screen
battery
battery adapters
battery life
battery lifespan
best bang for the buck
black-lit screen
blue light
blue-screen display
break
buttons
capacity
carrying cases
case
casing
cd-ripping program
changeable battery
construction
controls
cover
cumbersome
customer rep.
customer support
customer support website
customization
customization options
dent
dial
disk space
documentation
download
drivers
drop
dropped it
earbuds
earphone plug
earphones
ease of use
ease of use
eax
eax environmental audio
eax mode
eax settings
eax support
environmental audio
equalizer
equipment
explorer program
eye candy
face-plate
features
filecount capacity
file errors
file limits
filenames
file transfer
filing system
finding

firewire
firewire plug
firmware
firmware update
fits in
fly wheel
fm receiver
fm transmitter
folder
folder structure
fragile
freeze
front plate
gadgets
games
getting files
gigs of media files
gigs of music
hd size
headphone
headphone earbuds
headphones
heavy
high resolution screen
holding case
id3 tags
id tags
install
installation instructions
installing
instruction booklet
instruction manual
instructions
interface
interfacing software
jog dial
lcd
lcd screen
leather case
leather pouch
line out jack
lithium batteries
loading
location of various buttons
locking
lock up
looks cool
looks nice
mechanics
mediasource
mediasource software
menus
mp3s
music and data storage
navigating options
navigation
navigational system
navigation system
navigator
nomad explorer
operating system
option to clear the previous playlist
organization
pc compatibility
pdf instructions
playback quality
playing list
playlist
play list
playlists
playlists
play options
presets
price
program
random
realiability
recharge
rechargeable batteries
recharger
refuse to work
remote
replaceable battery
replacement battery
reset button
resilience
ripping
screen
screen options
screen saver
scroll
scroll bar
scroll button
scroll tab
scrollwheel
scroll wheel
search function
set of cans
setup
setup process
shipping
shuffle
singal to noise ratio
size
sized
software
software downloads
sort
sorting the files
sound

sound
sound options
sound presets
sound quality
sound-quality
spare battery
standard notebook drive
storage
storage capacity
storage space
switch
tags
tech support
tech. support
tech support operators
title of the song
toggle switch
tranferring
transfer
transfer files
transfering
transfer of data
transfer process
transfer rate
transferring
updated driver
upgrade
upload
uploading
usb 2.0
using
value for the price
value per gb
viewing hole
visuals
volume
volume range
warrant
warranty
weighs
weight
window
wma files

Annexe C

Regroupement de termes : première étape

Téléphone cellulaire, 100 mots clés avec ses groupes :

Mot clé	**Groupe**
email	email, email feature, short emails, email client, writing emails
battery	battery door, battery time, battery power, battery capacity, life battery, battery consumption, battery cover, battery use, battery indicator, rechargeable battery, lithium battery, battery line, battery meter, actual battery, polymer battery, big battery, battery life, battery quality, battery compartment, standard battery, battery level
id	id display, caller id, picture id, id screen, id, id feature
favorite	personal favorite
microphone	external microphone, microphone combo, microphone gain, microphone quality
ericsson	ericsson
charger	extra charger, desktop charger, charger connector, car charger, top charger, charger device, base charger, charger input, travel charger
charging	charging
software	software, software design, software flaw, impressive software, recognition software, software support
keypad	circular keypad, numeric keypad, keypad lock, phone keypad, keypad, keypad shape, keypad area, keypad style, style keypad
mode	night mode, analog mode, mode feature, mode
calendar	calendar function, calendar manager, calendar feature, active calendar, calendar

bluetooth	bluetooth function, bluetooth, bluetooth headset
scroll	scroll button, scroll wheel, scroll key
flip	flip feature, flip, flip door, flip phone
lcd	lcd screen, extra lcd
info	info feature
silver	silver plastic, silver design
ear	ear piece, ear, ear volume
switch	switch
screen	screen visibility, screen display, color screen, internal screen, screen feature, matrix screen, interior screen, exterior screen, outer screen, large screen, external screen, screen size, colour screen, big screen, display screen, screen resolution, digital screen, video screen, orange screen, screen quality, id screen, lcd screen, small screen, backlit screen, inside screen, resolution screen, blue screen, primary screen, main screen, monochrome screen
ring	ring option, ring volume
antenna	antenna, antenna technology, internal antenna, antenna setup, antenna design, antenna supplyed, stub antenna, rubber antenna, prodruding antenna, short antenna, retractable antenna
download	download speed, download
color	color screen, background color, boy color, color, color definition, color feature, color display
headset	special headset, headset access, free headset, wireless headset, headset, bluetooth headset, headset mic
lightweight	lightweight phone
polyphonic	polyphonic ringer
messaging	picture messaging, messaging option, text messaging, messaging, messaging feature
pad	phone pad, key pad, arrow pad, numeric pad, navigation pad, tiny pad
incoming	incoming sound, incoming call
bright	bright display
samsung	samsung, samsung phone
nokia	nokia model
leather	leather case
palm	palm, palm portion
button	navigation button, power button, button layout, scroll button, end button, talk button, directional button, tall button, side button, command button, menu button

digital	digital signal, digital zoom, digital operation, digital screen, digital camera, digital reception, digital coverage, digital service, digital display
dial	voice dial, dial, speed dial
web	web experience, web connection, web site, wireless web, web feature, web, web service, web browser, web access, web use, web video, web usage, mobile web
text	menu text, text system, text feature, predictive text, text entry, text interface, text messaging, text, text input, text recognition
alarm	alarm function, alarm clock, alarm
camera	camera function, camera feature, camera quality, pixel camera, digital camera, camera phone, camera lens, camera
device	memo device, charger device, separate device, device
wireless	wireless web, wireless headset, wireless productivity, wireless service, wireless
plug	piece plug
browser	browser, web browser
interface	cradle interface, text interface, phonebook interface, link interface, user interface, menu interface
analog	analog network, analog mode, analog coverage, analog reception
infrared	infrared port
provider	service provider
reception	school reception, reception strength, overall reception, phone reception, signal reception, digital reception, gsm reception, radio reception, analog reception, outdoor reception
ringer	ringer quality, ringer, ringer volume, polyphonic ringer, ringer selection
display	bright display, screen display, monochromatic display, inside display, id display, inner display, main display, display area, scale display, front display, display screen, outer display, secondary display, monochrome display, internal display, color display, call display, exterior display, external display, display backlight, display quality, digital display
sprint	sprint, sprint service, sprint vision

feature	flip feature, feature list, radio feature, great feature, dialing feature, camera feature, web feature, info feature, clock feature, wallet feature, feature set, sound feature, screen feature, command feature, message feature, recording feature, history feature, datebook feature, joystick feature, text feature, recognition feature, voicemail feature, mode feature,stopwatch feature, stick feature, email feature, vibration feature, speakerphone feature, color feature, phone feature, mail feature, cancellation feature, calendar feature, messaging feature, memo feature, log feature, id feature, screening feature
internet	internet use, internet homepage, internet
durable	durable phone
verizon	verizon
cell	cell phone
phone	world phone, phone pad, right phone, phone volume, admirable phone, overall phone, phone, phone design, phone breakup, lightweight phone, phone use, phone package, phone traffic, phone security, phone book, phone reception, phone clarity, phone usability, phone integration, mobile phone, particular phone, camera phone, phone performance, phone feature, entire phone, little phone, phone usage, cellular phone, samsung phone, speaker phone, phone keypad, flip phone, phone operation, phone size, black phone, durable phone, cell phone, phone aspect, phone coverage
navigation	navigation button, menu navigation, navigation mouse, thumb navigation, navigation key, navigation pad
clock	clock feature, important clock, alarm clock, clock
manual	instruction manual, manual
holster	plastic holster, holster design, clip holster
gsm	gsm, gsm reception
alert	silent alert, alerts, vibration alert
outlook	outlook sync
signal	digital signal, signal strength, strong signal, quality signal, signal pickup, weak signal, signal quality, tower signal, signal reception, signal, service signal
indicator	battery indicator
pocket	pocket
lock	lock feature, keypad lock, lock
phonebook	phonebook interface, phonebook capacity, phonebook
calculator	calculator

volume	phone volume, ring volume, volume control, ringer volume, tone volume, speaker volume, ear volume, volume, call volume
menu	menu access, menu navigation, menu text, menu structure, menu control, menu system, main menu, function menu, menu interface, menu button
caller	caller id, caller
tone	tone, tone volume, tone function
faceplate	standard faceplate
mobile	mobile thing, mobile phone, mobile web
clip	belt clip, clip, clip holster
sync	outlook sync, sync cable
voice	voice message, voice dial, voice recording, voice dialer, voice clarity, voice command, voice dialing, voice mail, voice quality, voice memo, voice channel, voice recognizer, voice recognition
voicemail	voicemail feature
backlight	exterior backlight, green backlight, blue backlight, display backlight
functionality	search functionality, overall functionality, tooth functionality, card functionality
store	store
dialing	speed dialing, dialing feature, dialing function, voice dialing, recognition dialing, dialing system, dialing
customer	customer support, customer service
kit	radio kit, office kit, car kit
tiny	tiny thing, tiny pad
speakerphone	speakerphone option, speakerphone quality, course speakerphone, speakerphone feature, speakerphone capability, speakerphone
accessories	accessories
plastic	silver plastic, plastic case, plastic casing, plastic holster
cellular	cellular service, cellular network, cellular phone, cellular company
ringtone	ringtone composer, ringtones
standby	standby, standby time, standby claim
adapter	power adapter
keyboard	thumb keyboard
usage	phone usage, web usage

Annexe D

Learning To Classify Questions (Feiguina & Kegl 2005)

Learning to Classify Questions

Olga Feiguina and Balázs Kégl
Département d'informatique et de recherche opérationnelle
Université de Montréal
{feiguino,kegl}@iro.umontreal.ca

Abstract

An automatic classifier of questions in terms of their expected answer type is a desirable component of many question-answering systems. It eliminates the manual labour and the lack of portability of classifying them semi-automatically. We explore the performance of several learning algorithms (SVM, neural networks, boosting) based on two purely lexical feature sets on a dataset of almost 2000 TREC questions. We compare the performance of these methods on the questions in English and their translation in French.

1 Introduction

Many question-answering (QA) systems include a component that classifies natural language questions in terms of the type of answer expected. In most (but not all, see Related Work) systems, this is done using hand-crafted lexico-syntactic patterns. It is desirable to use learning algorithms to perform this classification to decrease the required manual effort and increase language portability, ideally maintaining the accuracy. Since no standard test set of questions nor even a standard set of question categories exist, it is hard to perform comparisons. A rough estimate of the coverage achieved by semi-automatic methods on TREC questions is 90% [3].

2 Data

We took a set of 1893 TREC questions in English and in French, the latter a translation of the former done at NRC by professional translators. The questions are largely factual, requiring a brief answer, e.g. "Where was George Washington born?" or "What are animals without a backbone called?". We manually classified them into eight classes listed in Table 1. Since no standard set of categories exists in the domain, we chose a rather coarse set that seemed appropriate to us. The proportion of each type of questions in the corpus is presented in the table as well; the uneven distribution of the data over classes complicates the classification problem. It's worth noting that these classes are not necessarily mutually exclusive, but we assume each question belongs to only one of them to simplify the annotation and learning procedures.

3 Features

Given a set of questions, there are three main ways of characterizing them: lexically, syntactically, semantically. Extraction of syntactic and semantic information usually requires language-specific tools. Moreover, many tools, such as parsers, are not well suited for processing questions since they are trained on corpora that don't contain many questions. For example, the Abney chunker labeled "Name" as a Noun in "Name an animal...". The most portable and accessible characterization is lexical.

As a simple version of lexical patterns, we considered the uni-, bi-, and tri-grams in the whole corpus. Using uni- and bi-grams resulted in slightly worse performance (of the decision trees) than using them together with tri-grams. Using uni-grams alone resulted in inferior performance to combining them with other n-grams.

Our only pre-processing step was the replacement of all numbers by *NUM*, which can be done easily for any language. Upon examination, some n-grams looked very useful (e.g. "another word for"), while others looked bland (e.g. "of the"). We experimented with using various sets of n-grams based on their frequency, and found the performance (of the SVM) wasn't affected: the important n-grams must be those that have medium range frequency.

We decided to procede with two sets: one with all n-grams but those that occur only once, the other limited to medium range n-grams (excluding top 5 and bottom 15/5/5 for uni/bi/tri-grams). The size of the former was 2835 for English and 3303 for French, and of the latter - 275 for English and 362 for French. We noticed that the number of n-grams for the French set was always higher, probably because French words get inflected more than English ones. Since our feature vectors were quite sparse, we applied Principal Component Analysis (95%) to most of the data: 2835 dimensions for the large English set were reduced to 891; 275 dimensions for the small English set - to 154, and 362 for the small French set - to 187 (we did not apply PCA to the large French set because we saw that it brought no improvements on the large English set, and the performances on the smaller sets were similar between the two languages).

4 Learning

4.1 Baselines & Setup

The lowest baseline of random guessing for an eight-class classification task is 12.5%. However, we devised a baseline by

Table 1: Question classes' meanings and proportions.

Code	Name	Description	Proportion in the corpus
0	Explanation	E.g. "How does X affect Y?"	2.9%
1	Living entity	People, animals, Gods, organizations, etc.	19.8%
2	Numeric	Cardinality, measurements, time, etc.	25.6%
3	Specialization	Asking for a term given its description	19.8%
4	Location	E.g. "Where can I find X?"	17.6%
5	Persondef	E.g. "Why is X famous?"	16.9%
6	Definition	E.g. "What is an X?"	10.7%
7	Synonym	E.g. "What is another word for X?"	18.5%

classifying the questions using basic rules a human can come up with without thoroughly examining a set of questions, such as "where" implies class "location". This simplistic classification method gave us $\approx 55\%$ accuracy. This was only performed for the English dataset.

As a third-level baseline, we used the K-Nearest-Neighbours algorithm (using WEKA[1]); the optimal K value was determined via cross-validation, and the 9-fold cross-validation accuracy was: **73% for English and 72% for French** using the *small feature set*, and **73% for English and 76% for French** using the *large feature set*. These are well above the 55% baseline even with this simple algorithm. Notably, the accuracy is about the same for the two languages. Using post-PCA data resulted in accuracies lower by several percent.

In our experiments, the validation set size was 200-250 samples (or $\approx 10\%$ of the corpus). We ensured that all classes were represented in the validation set used to set the parameters. Since the dataset is small, 9-fold cross-validation was used for final tests with optimized parameters.

4.2 SVM

Our first experiment was carried out using SVMTorch[2]. It does one vs. all multi-class classification and is designed to handle large datasets. Using a simple validation set, we found that the radial basis function kernel was best.

With the *small feature set*, this method attained an accuracy of **77.3% for English** and **79.2% for French**. Again, the accuracy is similar across the two languages. It is also 22% above our hand-crafted baseline and 5-7% above the accuracy achieved by KNN.

With the *big feature set*, it attained an accuracy of **80.6% in English**, and **80.8% for French**. This is 26% above our hand-crafted baseline and 5-8% above the accuracy achieved by KNN on this feature set.

Again, the accuracy is similar across the two languages for both sets. Using post-PCA data resulted in accuracies lower by several percent. Note that although the larger feature set gave superior performance, it is ten times bigger than the small feature set, and the improvements are only 2-4%. With a bigger dataset, this difference would likely be even smaller and possibly negligible.

[1] http://www.cs.waikato.ac.nz/ ml/weka/

[2] http://www.idiap.ch/ bengio/projects/SVMTorch.html

4.3 Neural Networks

Our next experiment involved using a two-layer sigmoid backpropagation-trained network (available in WEKA) with as many input nodes as there are features (i.e. 154 for English, 187 for French) and as many output nodes as there are classes (i.e. 8). Because of training times, only the post-PCA smaller feature set was used in this experiment.

The optimized parameters were found using a simple validation set: 100 nodes in the only hidden layer, 0.3 learning rate, 0.2 momentum, weight decay. This network setup resulted in **75.6% accuracy for English**, and **74.7% for French**. Yet again, the accuracies are approximately the same across the two languages. We are able to achieve good accuracy using rather small networks. Further experimentation with the networks was attractive but darkened by the training times in light of having to optimize parameters.

4.4 Boosting

We experimented with AdaBoost.M1 ([5], available in WEKA) using reweighting (as opposed to resampling). AdaBoost.M1 is the simplest generalization of AdaBoost to the multi-class case: it uses base classifiers that are capable of multi-class classification.

Since even KNN gives accuracy above 70%, we had trouble finding a weak algorithm to use as the base classifier. Decision stumps achieved only 30-40% on either feature set, so boosting couldn't be based on them. Boosting KNN resulted in a marginal decrease of accuracy.

We also attempted boosting decision trees despite the fact that on their own, they resulted in 79.3% accuracy with the large feature set (for English), and 76.2% with the small feature set. A boosted C4 decision tree resulted in **80.1%** with the large set, and **78.3%** with the small set, bringing these improvements after five iterations. **For French**, using the large feature set resulted in **81.7%** accuracy, and the small one - **77.2%**.

We see that M1 boosting can improve the accuracy of even fairly strong algorithms, but only marginally. Using post-PCA data resulted in slightly lower accuracies, but reduced the training time.

Table 2: Confusion matrix for English (En) and French (Fr).

Class Name	Code	0		1		2		3		4		5		6		7		Accuracy%	
		En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr
Exp.	**0**	22	25	-	2	4	-	19	20	2	3	3	2	4	2	-	-	41	46
Liv.ent.	**1**	1	-	295	297	3	3	45	52	11	7	12	10	1	1	7	5	79	78
Num.	**2**	4	2	2	3	435	434	32	30	4	13	-	-	7	2	1	1	90	90
Spec.	**3**	10	3	35	43	16	21	252	274	27	23	1	1	29	7	5	3	67	73
Loc.	**4**	2	2	5	9	5	5	36	26	280	291	-	-	4	1	2	-	84	87
P-def.	**5**	2	1	-	1	-	-	-	2	-	-	30	28	-	-	-	-	94	88
Def.	**6**	2	3	1	-	3	-	10	17	2	2	-	-	185	181	-	-	91	89
Syn.	**7**	1	1	6	7	-	-	10	9	-	-	-	-	1	1	17	17	49	49

5 Error Analysis

To analyze the errors, we looked at the confusion matrix of boosting a C4 decision tree based on 9-fold cross validation because it gave us one of the highest accuracies of 80.1% for English and 81.7% for French, based on the large feature set. We're interested in which type of questions were hardest to classify and which got confused most often. We also want to see if there is a relationship between the number of training samples available per class and the error rate for it. The matrix is presented in Table 2. The columns correspond to what the questions were classified as automatically, whereas the first entry in every row represents the true classification.

What stands out most is the amount of questions that were incorrectly classified as class 3 (Specialization). Likely, this can be explained by the fact that specialization questions have the most lexical variability - they are descriptions of various terms, so most of their n-grams are uninformative of the question class. Moreover, they usually use the question word "what", the least informative question word in terms of the answer type.

Second, we notice that class 0 (Explanation) and 7 (Synonym) were classified most poorly. For the former, this can likely be explained by under-representation: 2.9% of the corpus. For the latter, it is not clear. Most other classes were handled fairly well, especially considering that some misclassifications can hardly be considered mistakes. For example, if an instance of class 5 (Persondef) is classified as an Explanation question, this is understandable because questions such as "'Why is X famous?" can be easily viewed as Explanation questions as well. This goes back to our previous comment about assigning one category per question being a simplification.

Looking at the confusion matrix for French, we see a very similar picture. The minor differences indicate different uniformity of expressions used in certain types of questions in the two languages, but are overall marginal. It's worth noting that the corresponding matrices for other algorithms displayed similar tendencies.

Finally, seeing that the Specialization class was so problematic, we experimented with taking it out of the dataset. That left us with 1518 questions and 7 classes. We ran the C4 decision tree classifier on this data using the small feature set and got an accuracy of 86%, 10% higher than on the whole dataset. This leads to several ideas. First, perhaps some classes can be discovered using lexical features alone, and only the more complex ones need more sophisticated processing. Second, perhaps the Specialization class isn't a good idea in general: for example, it could be replaced by more specific classes.

We considered another form of evaluation - by comparing the automatically generated classification rules to the manual patterns. The algorithm whose rules were easiest to extract is the decision tree. Comparing them turned out to be difficult for a curious reason: the automatically created patterns relied a lot of the absence of n-grams from a given question, whereas the manually created patterns focused on the presence of things. Although the manual patterns do sometimes make use of absences, and the automatic pattens certainly make use of presences, the difference in focus seems to us big enough that a comparison is not possible.

6 Related Work

The problem of classifying TREC questions using learning algorithms has been pursued since 2002. Radev [4] used Ripper to classify questions into 17 classes and achieved an accuracy of about 70%. The next attempt was made by Li&Roth [2], who used SNoW and a large number of features (lexical, syntactic and semantic), some semantic ones constructed *semi-automatically*, and achieved near-perfect accuracy using a dataset of 5,500 questions.

Zhang&Lee [9] tested a variety of machine learning algorithms using bags of words and bags of n-grams as features. They also tried using smoothed bi-grams after generalizing the corpus syntactically (e.g. replacing clauses by CLAUSE), semantically (e.g. replacing animal names by ANIMAL) and lexically (e.g. replacing all numbers by NUMBER). They found that SVM was the most fitting algorithm. They achieved low 80s accuracy on coarse categories using a dataset of a size similar to ours, and high 80s using datasets of up to almost 6000 questions. They also experimented with adding syntactic information, which improved the accuracy to 90% for the coarse division. For fine-grained categories (almost 50), the top accuracy achieved was about 80%. [9] also includes the learning curves of every algorithm they tested. Although in most cases, the performance keeps improving as the set is augmented, the curves become less steep. It would be interesting to see these curves separately for different question categories.

Suzuki et al. [8] attempted classifying questions in Japanese using their HDAG kernel. The highest accuracy of about 95%

Table 3: Summary of our findings: accuracies in percent (baseline 55%).

KNN				SVM				PCA+Neural Nets				AdaBoost.M1			
En		Fr		En		Fr		En		Fr		En		Fr	
Sm	Lg	Sm	Lg	Sm	Lg	Sm	Lg	Sm	Lg	Sm	Lg	Sm	Lg	Sm	Lg
73	73	72	76	77.3	80.6	79.2	80.6	75.1	-	74.4	-	78.3	80.1	77.2	81.7

was achieved using a combination of bag-of-words, named entity labels and other semantic information.

Solorio et al. [7] focused on language-independent question classification. In addition to bags of words and word prefixes, they used Internet search engines by comparing the number of returned documents for queries such as "president is a person", "president is a date". They use SVM to perform the classification, and achieve low 80s results in English, Italian and Spanish using a small dataset of 450 questions.

7 Future Work

Automatic question classification has not been studied extensively. The comparison of various approaches is difficult due to the lack of a standard data set and a standard set of question categories. Establishing those would be beneficial both in terms of allowing comparisons and reducing the manual labour of annotation.

What's obvious by now is that very accurate automatic classification of factual questions is possible using a combination of lexical, syntactic and semantic information. The challenge lies in developing language-independent methods, which may include using language-independent named entity recognition and automatically acquired semantic information. It would also be interesting to see how automatic classification performs on more complex (longer, non-factual) questions. As mentioned above, allowing multiple categories per question would probably improve the accuracy and provide useful information as well.

One relaxation of the language-independence constraint that would be interesting to explore is the use of part-of-speech labels. It would be especially interesting to employ mixed n-grams that contain both words and POS labels, since manually created patterns are mixed in this sense.

In terms of experiments with boosting, our results with AdaBoost.M1 and decision trees suggests exploring the alternating decision trees algorithm proposed by Freud&Mason in [1]. The AdaBoost.MH [6] approach is attractive because of the easy extension to multi-label tasks, but our preliminary single-label experiments resulted in disappointing baseline-level accuracies.

Another avenue to explore in future work would be the use of large unlabelled corpora which can help get around the lack of linguistic processing; it could be used in conjunction with active learning.

8 Conclusion

In this paper, we described our comparison of various learning algorithms with respect to question classification based on word n-grams. We experimented with two feature sets, one of about 300 features, the other about 10 times bigger, and found that using the latter brings consistent but small accuracy improvements. The best accuracies attained were 80-82% for English and French, using SVM and boosting decision trees. The performance was consistently similar for English and French. We summarize our findings in Table 3.

Acknowledgments

Guy Lapalme for guidance in the setup of the classification task. Norman Casagrande for discussions of multi-class boosting.

References

[1] Y. Freund and L. Mason. The alternating decision tree learning algorithm. In *Proceedings of the 16th International Conference on Machine Learning*, pages 124–133.

[2] X. Li and D. Roth. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, 2002.

[3] L. Plamondon and L. Kosseim. Le web et la question-réponse : transformer une question en réponse. In *Journées francophones de la toile (JFT 2003)*, pages 225–234, Tours, France, jul 2003.

[4] D. R. Radev, W. Fan, H. Qi, H. Wu, and A. Grewal. Probabilistic question answering from the web. In *Proceedings of the 11th International World Wide Web Conference*, 2002.

[5] R. E. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 1999*.

[6] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[7] T. Solorio, M. Pérez-Coutiño, M. M. y Gómez, L. Villaseñor-Pineda, and A. López-López. A language independent method for question classification. In *Proceedings of the 20th International Conference on Computational Linguistics, COLING-04, volume II*, pages 1374–1380, 2004.

[8] J. Suzuki, H. Taira, Y. Sasaki, and E. Maeda. Question classification using hdag kernel. In *Workshop on Multilingual Summarization and Question Answering 2003, (post-conference workshop in conjunction with ACL-2003)*, pages 61–68, 2003.

[9] D. Zhang and W. S. Lee. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 26–32, New York, NY, USA, 2003. ACM Press.