Analyse sémantique d'opinion

Soufiene Katet

MEMOIRE

En vue de l'obtention du

MASTER

Produits de l'Information Spécialisée et Médiation Electronique

Par

Soufiene KATET

ANALYSE SEMANTIQUE D'OPINION

Soutenu le 15 Septembre 2011, devant le jury composé de :

M.	*Rapporteur*
M.	*Examinateur*
M.	*Encadrant*

REMERCIEMENTS

Je tiens à remercier, tout d'abord, Monsieur Ismail TIMIMI, pour son encadrement ainsi que son soutien tout au long de ce travail.

Je remercie tout particulièrement Monsieur Ghalem OUADJED de m'avoir accueilli au sein d'EOWEO, pour sa disponibilité, son encadrement, ses conseils et son soutien inestimable.

Je remercie tout particulièrement Monsieur Isam SAHROUR de m'avoir accueilli au sein d'Euratechnologie Lille.

J'exprime mes remerciements et ma gratitude à tous ceux qui ont apportés l'assistance nécessaire pour finaliser mon travail.

Je remercie vivement les enseignants qui ont bien voulu accepter de faire partie de mon jury.

Je présente mes remerciements les plus sincères à tout le corps enseignant du Master PRISME et tous les membres d'EOWEO et Euratechnologies Lille.

Je remercie chaleureusement mes collègues Vivien MANN, Christophe WILLAERT et Madeleine HUBERT.

Table des Matières

Liste des figures

Introduction

"Qu'est-ce que les autres pensent d'un tel produit ?" ; "De quelle réputation bénéficie une marque" ; "Quelles sont les rumeurs véhiculées sur une société ?"... un ensemble de questions qui demeurent toujours un élément important de l'information pour la plupart des organismes au cours du processus de prise de décision. Bien avant la généralisation du World Wide Web et la prolifération de l'information numérique, beaucoup d'organismes s'appuient sur l'avis d'autrui pour une analyse de situation et une prise de décision (avis des consommateurs, sondages des électeurs...

Mais la socialisation des nouvelles Technologies (web, mobiles, TV connectic...) et l'émergence de nouveaux usages ont actuellement permis de réceptionner et d'analyser de manière plus élargie les opinions et les avis de personnes, souvent externes, voire inconnues des listes de contacts d'un organisme. Aujourd'hui, le web comprend un grand nombre de corpus d'opinion et de sentiments... leurs auteurs expriment aisément leurs avis et recommandations....

Dans la littérature, l'analyse des sentiments est connue sur le nom d'Opinion Mining et elle est récemment devenue un domaine en plein développement en raison de ses nombreuses applications. Mais à part le support du moteur prédicatif nous pouvons citer des nombreuses utilisations comme : la recommandation (par exemple des voitures), l'explication des sondages des suffrages aux élections, la consultation des avis sur les produits, la détection de spam, l'analyse et la surveillance des opinions pour améliorer les produits (matériels ou intellectuels) ou l'étude de marché.

Il est important de mentionner qu'en raison de toutes les applications possibles, il y a un nombre considérable d'organismes administratifs, économiques, politiques... qui exploitent l'analyse de l'opinion et l'analyse des sentiments dans le cadre de leurs missions.

Si du côté sociétal l'intérêt croissant pour les analyses d'opinion et les analyses des sentiments se justifie des ces applications potentielles précitées, du côté scientifique, nous constatons un regain d'intérêt depuis 2002 pour le sujet.

Dans notre travail de recherche envisagé, nous souhaitons concevoir et implémenter une nouvelle méthode pour l'analyse d'opinion dans les corpus en ligne. Il s'agit d'une approche s'appuyant sur des ressources sémantiques externes d'enrichissement.

Ce mémoire est structuré en cinq parties :

Chapitre I : Nous y introduisons les technologies de base et les outils utilisés dans Opinion Mining.

Chapitre II : Dans ce chapitre, nous introduisons les principaux travaux existants pour nous donner des approches exemplaires et des idées.

Chapitre III: Nous présentons le web sémantique est ses apports

Chapitre IV: Nous listons des outils de réalisation des applications de web sémantique.

Chapitre V: Le mémoire se conclut par une synthèse des travaux réalisés, et la présentation des perspectives qui peuvent être envisagées.

CHAPITRE I :

LES TECHNOLOGIES DE BASE

Ce chapitre présente les technologies de base et les outils utilisés par l'opinion Mining, un domaine spécialisé de l'exploitation du Web. Le Web Mining reste à la croisée de recherche de l''information, de l'extraction de l'information et du Data Mining. La recherche d'information (Information Retrieval) et l'extraction d'information (Information Extraction) jouent un rôle important pour localiser et extraire des informations précieuses sur des données non structurées, avant qu'elles ne soient aptes à être traitées par des applications de data mining.

L'exploration de ces techniques est extrêmement nécessaire pour faire face à la quantité de données d'information disponibles. Aussi, avec le fait que le web qui est devenu de plus en plus orienté vers l'importance de la sémantique et l'intégration de l'information, ces domaines d'étude sont devenus très importants pour répondre aux nouvelles tendances du Web.

Ce chapitre est divisé comme suit:

La première partie donne un aperçu des moteurs de recherche et fournit des explications sur ses composantes de base. La deuxième partie présente des outils et des techniques d'extraction de l'information. Enfin, la troisième partie introduit le Web Mining.

1.1 Moteur de recherche web

Information Retrieval (IR) est un domaine d'étude qui concerne la récupération de documents d'une collection d'autres documents (pertinents et non pertinents), généralement basée sur des recherches par mot clé. Avec l'expansion d'Internet, la recherche d'information est d'une grande importance et les moteurs de recherche sont devenus une façon dominante d'accès à l'information sur le Web.

Aujourd'hui, en raison de leur importance, les moteurs de recherche sont devenus l'outil le plus représentatif de la recherche d'information. Cette partie traitera les technologies, les objectifs et les enjeux des développements impliqués dans leurs conceptions.

Une des raisons pour lesquelles certains moteurs de recherche ont autant de succès sur Internet est leur engagement à la qualité des services, en particulier à l'égard de la vitesse de traitement des requêtes des utilisateurs. L'Internet d'aujourd'hui, avec des milliards de pages disponibles et l'absence de mécanismes qui fournissent une réponse dans un court laps de temps incite à quitter les systèmes incompatibles avec les nouvelles normes (plus de données doivent être traitées avec des contraintes encore plus strictes à l'égard de temps).

Les moteurs de recherche (par exemple, Google, Yahoo, Bing, etc) sont capables d'atteindre un haut niveau de service principalement grâce à leur technique d'indexation associé à des infrastructures de haut gamme composé de plusieurs centaines de clusters hautement optimisé pour des emplois exigeants une grande capacité de traitement. Ces moteurs sont très évolutifs et sont capables de fournir des services de haute qualité, même avec des millions d'utilisateurs accédant simultanément à leurs systèmes. Par ailleurs, les algorithmes de classement (par exemple, le PageRank de Google), sont capables de trier les documents les plus importants liés à une recherche. Sans l'aide d'algorithmes de classement, un utilisateur n'aurait aucun indice sur l'endroit où commencer à chercher une information désirée parmi plusieurs autres documents. Un algorithme de classement offre un niveau hiérarchique de plusieurs documents importants, offrant ainsi un premier indice à l'utilisateur sur l'endroit où l'information désirée est plus susceptible de l'être. Lors d'une opération de recherche, les interactions suivantes sont effectuées, comme le montre la figure1 : (1) Une requête est soumise par l'utilisateur. (2) La requête utilisateur est vérifié pour s'assurer qu'elle est prête à être utilisée par le système de récupération. Ceci pourrait être réalisé grâce à des tâches simples telles que la suppression des « mots vides », en réduisant les mots aux racines (radical) et en vérifiant l'orthographe. (3) La requête est vérifiée par rapport aux indices disponibles afin de récupérer les documents qui contiennent certains termes de la requête. Ensuite, un algorithme de classement est appliqué à l'ensemble des documents trouvés qui sont présenté à l'utilisateur (les documents les plus pertinents apparaissent au début de cette liste). (4) L'utilisateur reçoit la réponse et accède aux documents correspondants à partir de la liste de résultats.

Les étapes ci-dessus montrent que le moteur de recherche dans sa phase opérationnelle de recherche desserve directement une requête utilisateur. Cependant, les principales tâches doivent être effectuées à l'avance, le crawling les pages Web, l'indexation et le calcul de classement. Les paragraphes suivants présentent chaque sous-système interne d'un moteur de recherche et leurs tâches respectives.

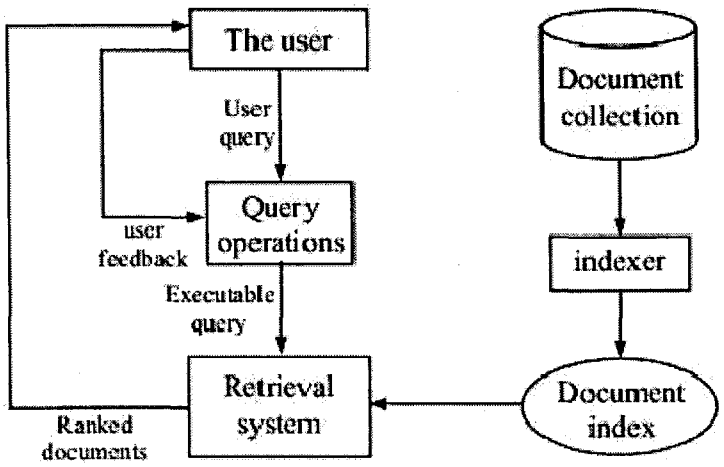

Figure 1:Un système de moteur de recherche lors d'une opération de recherche (1)

1.1.1 Web Crawlers(robots web)

Les moteurs de recherche s'appuient sur des programmes informatiques appelés web crawlers (aussi appelé des robots Web), pour parcourir les pages Web en suivant les hyperliens et stocker les documents web qui sont indexés plus tard pour optimiser le processus de recherche. Un web crawler est probablement la composante la plus importante et la plus complexe d'un moteur de recherche.

Les web crawlers ont deux questions importantes à aborder: La première consiste à utiliser une bonne stratégie de crawler (ce qui inclut l'algorithme pour visiter de nouvelles pages Web) et les mécanismes intelligents pour optimiser le processus de recrawling. Deuxièmement, parce que cette tâche computationnelle est intensive, le système doit être capable de faire face à de nombreux scénarios différents dans des circonstances différentes (panne matérielle, problème de serveur, erreurs lors de l'analyse de documents).

Un système Web Crawler (1), est composé par les éléments suivants, tel que présenté sur la figure 2:

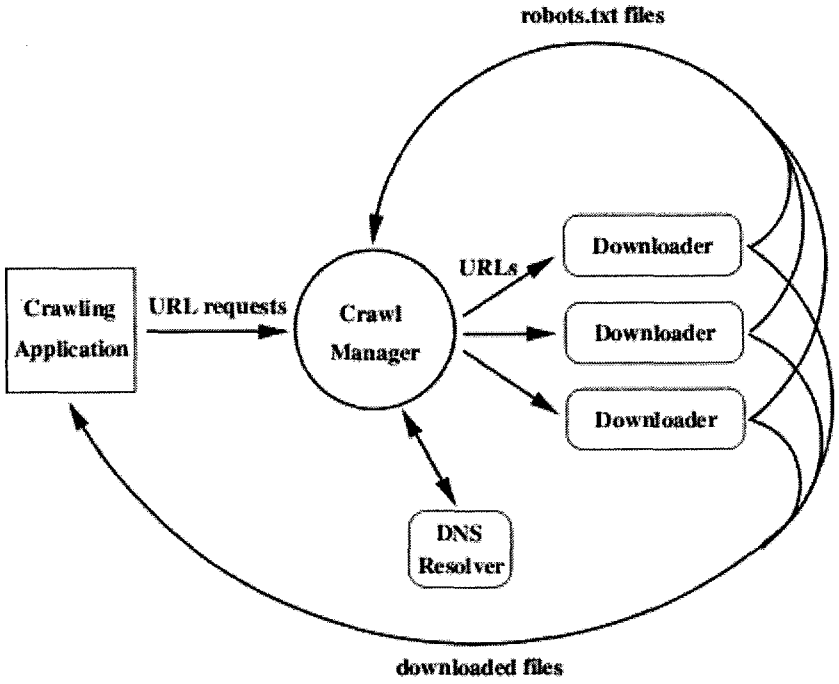

Figure 2:Un système web crawler en détail

Un crawler Manager (1) est chargé de transmettre les demandes d'adresse URL au downloders.

Généralement l'opération de crawling commence avec une liste d'hyperliens, le crawler se connecte sur les pages Web suivant un plan de route, copie leur contenu, et analyse les hyperliens (les URLs) contenus dans les pages copiés et les ajoute à la liste d'URLs à visiter .En outre, cette composante a pour tâche de faire respecter les règles imposées par robots.txt (Robots.txt est un fichier utilisé pour appliquer les règles que les robots Web devraient suivre en explorant les liens d'un site web. Normalement, un web crawler vérifie ce contenu afin de s'assurer qu'il est autorisé à visiter une certaine section d'une page web.), fournies par les administrateurs de serveurs Web.

Les downloaders sont responsables de l'ouverture des connexions avec les différents serveurs web.les moteurs de recherche développés reçoivent des centaines de pages par seconde à travers les downloaders.

Les Méthodes de Crawling

Les Web Crawlers peuvent crawler les pages web de différents façons. Ceci est principalement lié à l'application finale que le système servira. Deux exemples de crawling:

Crawler en largeur (Broad Breadth-First Crawler)

Un Crawler peut sélectionner un petit ensemble de pages Web, et suivre leurs liens en utilisant l'algorithme de parcours en largeur.les Moteurs de recherche emploient une série d'autres techniques pour améliorer l'algorithme de crawling. Avec cette stratégie tous les liens sont suivis et donc il n'ya aucune restriction entre les éléments couverts par le site.

Crawlers topiques (A focused crawler or topical crawler)

Crawlers topique, aussi connu comme les robots ciblés, tentent de crawler des pages spécifiques .ils pourraient être les pages d'un sujet particulier ou dans une langue spécifique, image, mp3 ou des documents de recherches en sciences informatiques. L'objectif de ces robots est de trouver le plus grand nombre possible de pages sans utiliser beaucoup de bande passante.

1.1.2 Lemmatisation, la racinisation (stemming) et mot vide

La lemmatisation simple consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et/ou féminins au masculin singulier avant de leur associer un nombre d'occurrences. Ce processus permet d'amoindrir la malédiction dimensionnelle qui pose de très sérieux problèmes de représentation dans le cas des grandes dimensions. La lemmatisation permet donc de diminuer le nombre de termes qui dénieront les dimensions de l'espace de représentation de termes ou espace vectoriel. D'autres mécanismes de réduction du lexique sont aussi déclenchés. Les mots composés sont repérés automatiquement à l'aide d'un dictionnaire, puis transformés en un terme unique lemmatisé en utilisant des tableaux associatifs.

Pour optimiser le processus de recherche et maximiser la capacité de stockage, les dernières pages web analysées sont prétraitées avant d'être indexé :

L'objet de ces techniques et de ramener un mot à son lemme ou sa racine. Par exemple le mot chantage deviendrait pour le stemming (racinisation) le mot chant. La lemmatisation va moins loin et se contente de ramener les noms, les adjectifs,... au masculin singulier et les verbes à l'indicatif.

La suppression des chaînes de caractères dont le poids sémantique est trop faible (également désignés « mots vides » ou « bruit ») : le, la, les, du, avec, vous, etc., qui jouent rarement un rôle intéressant dans les recherches et risquent de ralentir notablement le processus.

1.1.3 Index inversé

Un système de moteur de recherche pourrait rechercher des milliards de documents. Rechercher tous les termes spécifiques (d'une requête utilisateur donnée), prendrait beaucoup de temps. Afin d'aider les moteurs de recherche a effectué la recherche dans un délai acceptable, le système de récupération utilise les données structurées appelées index. Le meilleur schéma d'indexation et le plus largement utilisé pour les moteurs de recherche sur le Web est l'index inversé. Un index inversé est une structure de données composée d'un terme et tous les documents qui contiennent ce terme. L'index inversé fonctionne exactement comme un « index de livre ».

1.1.4 Algorithmes de classement

Avec la quantité de documents en ligne, il est presque impossible pour un utilisateur de vérifier chaque document pour témoigner sa pertinence. Aussi, les algorithmes de classement aident à vaincre le web spamming, une pratique non seulement nuisible à l'expérience des utilisateurs, mais aussi de recherche d'information en commerce.

Un des algorithmes les plus importants de classement, dans la recherche web, est PageRank de Google, qui utilise généralement le concept de prestige pour trier les documents pertinents. L'idée est que les pages web qui sont référencées par beaucoup d'autres pages web (par hyperliens) sont susceptibles d'être des pages web importantes.

Par conséquent, la page qui a les plus des liens entrants, elle est la plus importante. Cependant, le score de prestige n'est pas seulement limité par le nombre de liens qui pointent vers une page web. L'algorithme prend également en compte le prestige d'une autre page.

Par conséquent, l'importance de la page i (le score PageRank) est déterminée en additionnant les scores de PageRank de toutes les pages qui pointent sur i divisé par le nombre de leurs liens sortants.

1.2 Extraction d'information

Extraction d'Information (IE) est une sous-discipline de l'intelligence artificielle qui vise à extraire des informations précieuses des données non structurées. Un système d'extraction d'information est généralement axé sur l'identification des entités ou des objets (personnes, lieux, entreprises, etc) et des règles d'extraction, mais pas nécessairement de domaine spécifique. Les données non structurées peuvent avoir plusieurs formes différentes, comme des vidéos, images, audio et texte. Les premiers systèmes d'extraction d'information ont été principalement axés sur le texte, et encore

aujourd'hui c'est le type des données le plus exploré par la communauté des chercheurs et des commerciaux. Le but de l'IE est d'identifier les parties utiles de données brutes (données non structurées) et les extraire pour créer plus d'informations précieuses grâce à la classification sémantique. Le résultat peut être adapté à d'autres tâches de traitement de l'information, telles qu'IR et de Data Mining. Il y a une différence entre les objectifs d'IR et IE, mais dans le monde réel, ils doivent être considérés comme des activités complémentaires pour améliorer leur précision et exactitude.

1.2.1 Transformation de la structure de données dans un processus de génération document Web

La génération de documents web peut impliquer différents types de structures de données au long du processus. Dans les documents que l'on appelle web statique, un document HTML tiendra les mêmes informations (contenu), quel que soit le client qui demande la page, ou dans quel contexte cette page est appelée. Toutes les informations sont enfermés entre les balises HTML, dont la fonction principale est de fournir un balisage structurale sémantique du texte (paragraphes, listes, titres, etc.)

Avec des pages web dynamiques, les pages sont générés par un serveur de script, et ils changent habituellement comme une réponse pour différents clients selon différents scénarios. Ce sont des documents générés à la demande, un exemple serait en e-commerce qui montre des produits aux clients en fonction des recherches par mot clé. Différents mots-clés retournent des listes différentes des produits.

Un document web est composé normalement par plusieurs parties, où chacune d'entre elles est étiquetées avec des annotations HTML (, <title> <body>). En raison de cette propriété de l'étiquetage, les différentes parties du document sont en conformité avec les informations qu'elles détiennent. Un document web est un type d'un document semi-structuré (comme il conserve encore une sorte de structure, en comparaison avec un document texte). Une fois les données structurées deviennent une partie d'un document semi-structuré, les propriétés structurales sont perdues et donc pour récupérer les informations souhaitées, des techniques spéciales doivent être utilisées.

Un document web avec un texte entouré par une balise pourrait être traité de deux points de vue différents concernant la granularité de l'information désirée. Le document lui-même est semi-structuré, mais le texte à l'intérieur de la balise div est totalement non structuré.

Les balises fournissent un moyen très efficace pour déterminer les emplacements possibles d'une information cible à l'intérieur du document entier. Il devrait être clair que si on est disposé à reconnaître des entités à l'intérieur de ce texte, des techniques d'extraction spécialisés devraient être utilisé telle que l'exploration par traitement du langage naturel (TAL -NLP).

1.2.2 Web Scraping

Web Scrapping est une technique basé sur des scripts utilisés pour extraire des informations à partir des pages Web. Les pages Web sont des documents écrits en langage de balisage hypertexte (HTML) et plus récemment XHTML qui est basé sur XML. Les documents Web sont représentée par une arborescence structurée appelé le Document Object Model, ou tout simplement l'arbre DOM. L'objectif de HTML est de spécifier le format du texte affiché par les navigateurs Web.

Du point de vue fonctionnement, un Web scrapping ressemble à une opération manuelle de copier et coller. La différence ici est que ce travail est fait d'une manière organisée et automatique par un agent virtuel. Cet agent peut suivre des liens (par l'émission de requêtes HTTP GET) et soumettre des formulaires (par HTTP POST), parcourir de nombreuses différentes pages web.

Après avoir récupérer le document Web cible, l'analyseur suit des chemins spécifiques à intérieur du document pour récupérer les informations souhaitées. Ces chemins sont spécifiés par les sélecteurs CSS ou XPath. Ils utilisent les chemins relatifs ou absolus (basé sur l'arbre DOM) pour pointer l'analyseur à un élément spécifique à l'intérieur d'un document Web. Après avoir localiser l'information désirée, normalement le web scrapping utilise aussi les expressions régulières pour restreindre ou élaguer les informations localisées, afin de récupérer les données avec une granularité spécifique.

Un défaut important de Web scrapping , est la difficulté de généraliser les scripts d'extraction. Le script est généralement attaché au modèle DOM d'une page donnée, donc la dépendance introduite par XPaths ou des sélecteurs CSS, ne le rendent pas facilement réutilisables par différents sites web. Le Web scrapping ne peut être une solution optimale pour récupérer l'information, spécialement lorsqu'il est utilisé en grande échelle ou pour des solutions commerciales. Avoir un document entier lorsque seulement une petite partie de celui-ci est réellement nécessaire, en fait de lui un processus très coûteux du point de vue des performances. Cependant, toujours avec les lacunes mentionnées, le Web scrapping peut être

une technique très puissante, lorsqu'aucune autre option pour récupérer des informations n'est disponible.

1.2.3 Traitement Automatique des Langues

Le Traitement Automatique des Langues (TAL) est un domaine de l'informatique qui étudie les interactions des langages humains avec des ordinateurs. L'objectif principal de TAL est de permettre une efficace communication homme-machine, qui pourrait être soit en tant que forme parlée ou écrite. Ici, seule la forme écrite sera adressée.

Pour de nombreuses applications, il est souhaitable de traiter automatiquement des textes écrits en langage naturel. Les ordinateurs peuvent analyser et générer automatiquement des textes en langage naturel, extraire de la sémantique et identifier les objets du monde réel. En conséquence, de nombreuses nouvelles applications pourraient en bénéficier. Le paragraphe suivant présentera une importante application d'une technique de TAL utilisé dans le text mining appelé Part-of-Speech tagging.

Part-of-Speech tagging(POS)

Une application particulière de traitement du langage naturel est de déterminer chaque mot dans une phrase de chaque partie du discours, connu comme étiquetage grammatical. L'étiquetage grammatical est un processus qui consiste à associer aux mots d'un texte leur fonction grammaticale, grâce à leur définition et leur contexte .L'étiquetage grammatical, sous sa forme la plus simple dite étiquetage morpho-syntaxique consiste à affecter à chaque occurrence d'un corpus un symbole représentant sa catégorie grammaticale (nom, verbe, etc.).

La raison pour laquelle le marquage POS est si important pour l'extraction de l'information est le fait que chaque catégorie joue un rôle spécifique dans une phrase. Les Noms donnent des noms aux objets, des êtres ou des entités de notre monde. Un adjectif qualifie ou décrit des noms.

1.3 Le web mining

La fouille du Web (web mining) est l'application des techniques d'exploration de données en vue de découvrir des constantes, schémas ou modèles, dans les ressources d'internet. Il y a actuellement dans le web mining trois principales directions de recherche : Web Content Mining qui concerne l'analyse du contenu des pages Web, Web Structure Mining qui s'intéresse à l'analyse de la structure des sites Web, Web Usage Mining qui analyse le comportement des utilisateurs des sites Web.

1.3.1 Web Content Mining (WCM)

Le Web content mining a pour objectif d'extraire des connaissances à partir du contenu des pages Web. Ce contenu se présente sous différents types : texte, image, audio, vidéo, métadonnées et hyperliens. Le WCM décrit le processus d'extraction des informations à partir des différentes sources de données dans le Web. Ces sources de données sont structurées, telles que les tables et les bases des données, semi-structurées telles que les pages HTML ou non structurées telles que les textes. Le processus du WCM appliqué aux textes comprend généralement la même succession d'étapes que tout processus d'extraction des connaissances à partir des données. En effet, la première étape est celle du prétraitement des données (nettoyage, structuration...), la deuxième est celle d'application des techniques de data mining pour l'extraction des connaissances et la dernière est celle d'analyse et de validation. Cependant, la phase du prétraitement varie selon le type des données (textes, images, fichiers logs), de même le choix de la méthode de fouille des données varie selon l'objectif de l'analyse.

Le text mining tel qu'il est défini dans (2) est le "processus non trivial d'extraction d'informations implicites, précédemment inconnues, et potentiellement utiles, à partir de données textuelles non structurées dans de grandes collections de textes". Il représente ainsi l'opération d'analyse et de structuration de grands ensembles de documents par l'utilisation de techniques de traitement du langage naturel et des outils de fouille des données. Des exemples de ces techniques sont l'extraction d'information, la catégorisation de textes, la cartographie de textes et les modèles d'apprentissage automatique. Parmi les applications de text mining :

- La classification automatique des documents,
- Le résumé automatique des textes,
- L'alimentation automatique des bases de données,
- La veille sur des corpus documentaires importants,
- L'enrichissement de l'index d'un moteur de recherche pour améliorer la consultation des documents.

1.3.2 Web Structure Mining

Web Structure Mining s'intéresse à l'analyse des liens afin d'exploiter l'information véhiculée par ses liens et par le voisinage des documents Web. Par définition, la propagation de pertinence consiste à propager des scores attribués à des pages à travers la structure du Web.

Cependant, la plupart des algorithmes de propagation de pertinence utilisent des paramètres fixes de propagation qui dépendent des requêtes exécutées et de la collection de documents utilisée. De plus, ces techniques ne distinguent pas entre les pages répondant totalement ou partiellement à la requête utilisateur et ne tiennent pas compte des différentes thématiques abordées dans les pages web.

Les techniques d'analyse de liens ont été développées, premièrement, pour améliorer les performances de la recherche d'information sur le Web en calculant une valeur de pertinence d'un document en fonction non pas de son contenu seul mais également en fonction de son voisinage (documents reliés par des liens hypertextes), ainsi que de la structure globale du graphe. Deuxièmement, ces techniques nous permettent, dans une certaine mesure, et parmi d'autres techniques, d'atteindre et d'indexer des documents non visibles à l'utilisateur tels que les documents protégés, les bases de données, les documents multimedia (images, vidéos, etc).

1.3.3 Web Usage Mining (WUM)

La fouille de données d'usage du Web (Web Usage Mining (WUM), en anglais) est définie comme étant l'application du processus d'Extraction des Connaissances à partir de bases de Données (ECD) aux données issues des fichiers Logs afin d'extraire des modèles comportementaux d'accès au Web en vue de répondre aux besoins des visiteurs de manière spécifique et adaptée et faciliter la navigation (3) Comme les analyses se font à partir des fichiers logs de serveurs Web, on parle également de Web Log Mining.

Le WUM consiste en "l'application des techniques de fouille des données pour découvrir des patrons d'utilisation à partir des données du Web dans le but de mieux comprendre et servir les besoins des applications Web" (4).

La première étape dans le processus de WUM , une fois les données collectées, est le prétraitement des fichiers Logs qui consiste à nettoyer et transformer les données. La deuxième étape est la fouille des données permettant de découvrir des règles d'association, un enchainement de pages Web apparaissant souvent dans les visites et des " clusters " d'utilisateurs ayant des comportements similaires en terme de contenu visité. L'étape d'analyse et d'interprétation clôt le processus du WUM. Elle nécessite le recours à un ensemble d'outils pour ne garder que les résultats les plus pertinents.

CHAPITRE II :
OPINION MINING

Beaucoup de recherches dans l'opinion mining ont été faites pour l'identification des caractéristiques des produits et trouver l'opinion sentiment / orientation. Dans ce chapitre, les travaux effectués par (5) et (6) vont être exposés avec plus de détails que d'autres, avec plus d'attention à la dernière. La raison pour laquelle ces travaux ont été choisies parmi d'autres c'est leur solution d'identification automatique des caractéristiques et l'analyse des sentiments à un niveau optimale de granularité.

Aussi, les deux définissent des problèmes qui ressemblent, spécialement pour faire face aux opinions dans un contexte de e-commerce. Enfin, un argument important favorise l'étude de (6) avec plus de détails. Dans (5), le sentiment est analysé au niveau de la phrase, alors que cette approche fonctionne raisonnablement, elle peut cacher beaucoup de détails importants. Dans (6) ce problème est résolu grâce à une analyse très fine des sentiments faite au niveau des caractéristiques.

2.1 La définition des composantes d'opinion dans un contexte Opinion Mining

Les définitions utilisées dans cette partie ont été proposées dans (6), et ils résument les éléments importants qui composent une opinion. Certaines de ces définitions sont juste une observation naturelle des éléments présents dans les opinions, tandis que d'autres se réfèrent aux problèmes abordés dans (6). Pour cette raison, certains de ces définitions peuvent ne pas s'appliquer à d'autres travaux, car ils peuvent avoir des objectifs différents ainsi que d'autres stratégies qu'ils emploient pour les réaliser.

Définition du modèle d'objet

L'objectif principal de l'avis est de mettre en évidence les points forts et les faiblesses possibles sur les objets en cours de discussion (OuD). Les objets peuvent représenter une variété de choses dans le monde réel, comme les produits, organisations et personnes.

Un OuD est définie comme un arbre et l'utilisation d'une partie de relation pour décomposer un objet en différents éléments (qui à son tour peut être décomposé en sous-composants) .Un objet est associé au paire O: (T, A) , où T est une taxonomie des éléments (ou parties d'un objet) et éventuellement des sous-composantes, et A est un ensemble d'attributs de O. Comme dans une arborescence, les composants peuvent également avoir leur propre ensemble organisé.

Par exemple, un appareil photo représente le nœud racine et les opinions peuvent mettre en évidence les aspects à propos d'un attribut de l'appareil ainsi que des attributs d'une partie de l'appareil (les composants).

Dans la phrase « Cet appareil a un super design » par un exemple, le design est un attribut de la caméra (le nœud racine). D'autre part la phrase "La vie de la batterie est trop courte" parle de la batterie, qui est une composante de la caméra et la vie qui est un attribut de la batterie (autonomie). Une opinion ne doit pas nécessairement mettre en évidence que les attributs d'objets ou de composants, ils peuvent également se référer à l'objet lui-même.

Les parties suivantes vont utiliser ce modèle pour faire référence à des opinions ainsi que des objets cibles. Dans ce chapitre, l'accent sera mis sur l'exemple des produits qui représentent un exemple concret de modèle d'objet discuté ci-dessus. Ainsi, le mot ' fonction (caractéristique)' correspond aux composants et aux attributs, ce qui permettra également de simplifier le modèle en omettant la hiérarchie.

Caractéristiques Explicites et implicites

Quand une caractéristique f est rapidement disponibles dans un commentaire R, f est appelée une caractéristique explicite. Il ya des cas où une caractéristique f n'est pas disponible rapidement, dans R, donc elle est considéré comme une caractéristique implicite.

Exemple 1:

I. La vie de la batterie de cet appareil est trop courte

II. Cet appareil photo est trop grand

Dans la première phrase, la vie de la batterie est une caractéristique explicite, tandis que dans la seconde, la taille est une caractéristique implicite. La taille n'est pas mentionnée dans cette phrase, mais il est facile de comprendre que « grand » indique une caractéristique négative de l'attribut taille.

Opinion explicite et implicite

Une opinion explicite sur une caractéristique f est celle qui exprime directement les aspects positifs ou négatifs d'une caractéristique f. Une opinion implicite sur une caractéristique f est une phrase objective qui implique une opinion.

Exemple 2:

I. La qualité d'image de cette caméra est incroyable.

II. Cette écouteur s'est brisé en deux jours.

L'exemple ci-dessus tiré de (6) montre que dans la première phrase est claire et explicit et que l'opinion sur la qualité d'image est positive. Dans le second cas, l'opinion sur l'écouteur n'est pas explicite, mais on peut supposer qu'elle est négative, basé sur le contexte de la phrase.

2.2 Architecture d'un système d'Opinion Mining

Un système d'Opinion Mining proposé par (6) et (5) est composé par les éléments suivants, comme illustré dans la figure 3.

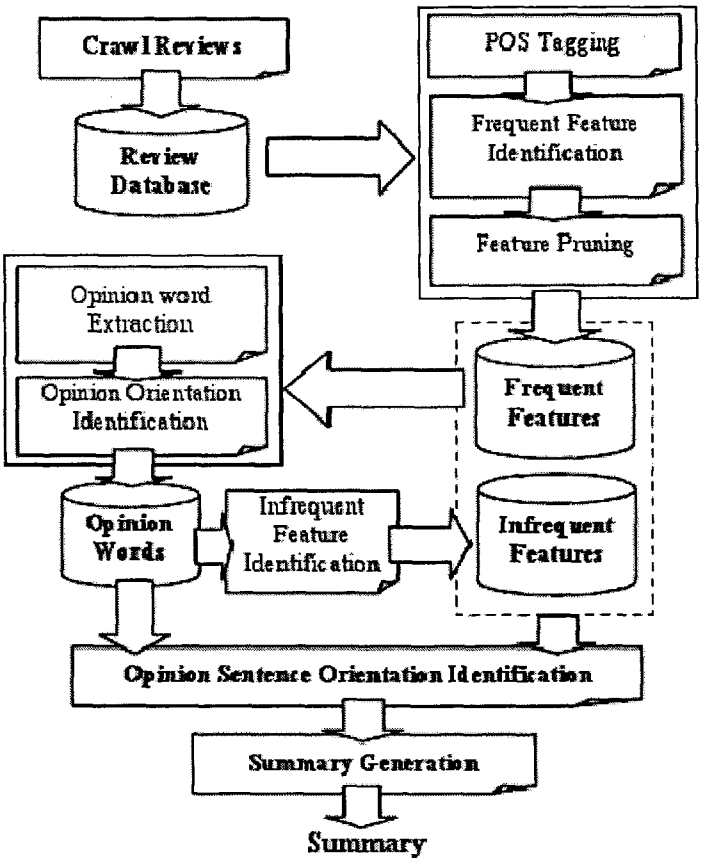

Figure 3:Architecture d'un système d'extraction d'opinion (5)

Le système compte un robot, qui télécharge tous les commentaires et les stocke dans la base de données. Après que POS Tagger tague toutes les critiques qui travaillent comme des crochets pour la partie responsable de l'exploitation des caractéristiques fréquentes .Cette étape est ignorée par certains systèmes d'annotation qui la font manuellement comme dans (7) où les ontologies sont utilisées pour annoter les caractéristiques des films manuellement. Ensuite, avec les phrases marquées et les caractéristiques identifiées, les mots d'opinion sont extraits et leurs orientations sémantiques sont identifiées à l'aide de WordNet. Maintenant, avec les mots d'opinion identifiées et extraites, le système identifie les caractéristiques rares. Dans la dernière partie du processus l'orientation de chaque phrase est identifiée, et un résumé est généré. Les parties suivantes discutent de certaines de ces étapes en détail.

2.2.1 Part-of-Speech Tagging

Dans (5) et (6) un tagueur (POS tagger) a été utilisé pour produire pour chaque mot une part-of-speech (diviser les opinions en phrases), comme indiqué dans le chapitre précédent. La raison pour laquelle les avis sont partagés en phrases est essentiellement de parvenir à la granularité la plus fine autant d'aspects discutés qui peuvent résider dans les différentes phrases qui composent l'ensemble du texte. Plus tard, il sera discuté le niveau de granularité optimale pour analyser des opinions.

Les phrases marqués produite par le NLProcessor dans cette étape, jouera un rôle très important pour le reste du système. Dans l'identification des caractéristiques, un système d'extraction des données dépendra des nom ou phrases nominales (deux à trois noms voisins dans une phrase) générée dans cette étape pour produire un certain nombre de caractéristiques fréquentes. En outre, la classification du sentiment dépendra des mots classifiés à la fois comme des adjectifs et des adverbes dans cette étape pour produire un ensemble de mots d'opinion possible.

Mot et phrase d'opinion

Un mot d'opinion est un terme utilisé par (5) et (6) pour faire référence à un mot qui est normalement qualifié comme un objet ou un attribut de cet objet. Ils sont généralement les adjectifs et les adverbes, mais ils peuvent aussi être des noms et des verbes. Une phrase d'opinion est une phrase qui détient au moins une référence à l'objet (qui pourrait être l'objet lui-même ou tout autre attribut de l'objet) et comprend également un ou plusieurs mots d'opinion. Les phrases «J'ai acheté cette caméra l'année dernière. Depuis lors, j'ai été très heureux avec sa qualité d'image.". Ici, la première phrase sera rejetée et ne sera pas encore analysée puisque aucun mot d'opinion n'a été trouvé. La seconde phrase satisfait la définition d'une phrase d'opinion puisque *heureux* est un mot d'opinion et la qualité d'image est une caractéristique de l'appareil photo.

2.3 Identification des caractéristiques

L'Identification des caractéristiques est le processus utilisé pour déduire les caractéristiques possibles des produits en dehors des textes marqués générés par la dernière étape. Les deux (5) et (6) utilisent des heuristiques pour les mots qui sont les plus susceptibles d'être une caractéristique dans une phrase. Normalement, le part-of-speech est le responsable de donner des noms aux entités du monde réel qui sont des noms, dans ce cas un nom donne un nom au produit et à ses caractéristiques (zoom, la vie de la batterie, qualité d'image,

etc.) Dans ces travaux, ils définissent deux catégories de caractéristiques, des caractéristiques fréquentes et des caractéristiques non fréquentes.

Dans (7)une approche basée sur l'ontologie a été utilisée pour extraire les caractéristiques d'opinions. Dans leur travail, ils l'ont expérimenté avec des critiques des films, où ils identifient des phrases contenant les terminologies d'ontologie.

Ici, il est important de différentier entre les deux approches avec leurs avantages et inconvénients. Dans (5) et (6), l'identification des caractéristiques est effectuée automatiquement. Le grand avantage de cette méthode est d'effectuer l'ensemble du processus automatiquement, avec une intervention humaine minime. Le plus grand inconvénient est que la sortie (les caractéristiques fréquentes) dépendra beaucoup du nombre d'avis en cours d'analyse.

En outre, il n'ya aucune garantie qu'une caractéristique fréquente trouvées par le système est en fait une caractéristique réelle. Dans (7) et d'autres travaux où les caractéristiques ont été annotées manuellement, l'avantage est que le système peut toujours identifier les caractéristiques réelles, étant fréquentes ou non. Cela dépendra juste de l'exactitude de l'annotation faite précédemment. Cependant, L'inconvénient majeur est qu'un grand nombre d'annotations doit être fait. Ils ne peuvent pas être seulement spécifiques à des catégories (comme les caméras numériques, jeux vidéo, téléphones cellulaire), mais ils pourraient être encore plus spécifiques tels que des modèles d'une marque spécifique (Nikon P90, Nikod D5000, etc.) Cela rendrait l'annotation des caractéristiques un travail très dur. Aussi, les gens peuvent commenter le manque de caractéristiques d'un produit donné, ou ils peuvent utiliser différents mots pour désigner la même caractéristique pour laquelle un système avec une annotation manuelle de caractéristiques va échouer à la reconnaître. Compte tenu de la brève comparaison entre les différentes approches ci-dessus, les méthodes explorées dans (5) et (6) ont besoin d'une intervention humaine minimale pour accomplir leurs tâches ce qui pourrait être amélioré plus tard par d'autres méthodes

2.3.1 Identification de caractéristiques fréquentes

Dans (5) , et (6), les systèmes proposés font l'extraction uniquement des noms ou des syntagmes nominaux (caractéristiques explicites possibles) à partir du texte. Dans cette étape, les noms extraits sont appelés caractéristiques des candidats.

Puis un algorithme d'exploration d'association trouvera les objets fréquents, qui sont l'ensemble des caractéristiques fréquentes (ceux dont nombreux utilisateurs en discutent). L'idée derrière cette technique est que les caractéristiques qui apparaissent dans des

nombreuses opinions ont plus de chance d'être pertinentes, et par conséquent, plus susceptibles d'être effectivement une caractéristique du produit réel. L'algorithme (8) a été utilisé pour générer l'ensemble des éléments fréquents. Toutefois, pour cette tâche il n'y avait pas besoin des règles pour trouver d'association entre les objets.

2.3.2 Identification de caractéristiques non fréquentes

Une heuristique très simple a été utilisée dans (5) pour découvrir les caractéristiques possibles non fréquentes (ceux référencées par un petit nombre de personnes).

Exemple 3:

I. Les photos sont absolument incroyables.

II. Le logiciel qui vient avec, il est incroyable.

Dans l'exemple ci-dessus, les deux phrases ont un mot commun d'opinion : incroyable. Parce que le mot d'opinion peut être utilisé pour décrire plus d'un objet, ces mots d'opinion sont utilisés pour chercher des caractéristiques qui n'ont pas pu être trouvées dans l'étape décrite avant. Les caractéristiques non fréquentes sont extraites comme illustré dans la figure 4 .

```
for each sentence in the review database
  if (it contains no frequent feature but one or more opinion
      words)
    { find the nearest noun/noun phrase around the opinion
      word. The noun/noun phrase is stored in the feature
      set as an infrequent feature. }
```

Figure 4:L'extraction de caractéristiques peu fréquentes

2.3.3 Analyse des sentiments des opinions

La classification des sentiments ou de l'analyse des sentiments est un domaine d'étude qui vise à classer les sentiments codées par des textes comme le montre l'exemple suivant:

Exemple 4:

I. La fille est en colère -> négative

II. Le soleil est absolument magnifique aujourd'hui -> positive

Le mot sentiment est synonyme de polarité et les deux sont largement utilisés pour décrire l'orientation des textes, des phrases et des mots comme dans l'exemple 4. Le travail portera sur le la classification des sentiments de textes des avis des utilisateurs, d'où le nom sentiment d'opinion.

Le travail effectué par (9) et (10) classifient chacun des opinions des utilisateurs dans son ensemble. Dans (5) et (11) la classification de sentiment se fait au niveau de la phrase.

Dans (6), chaque caractéristique au sein d'une opinion a un sentiment associé. La raison pour laquelle cette dernière approche est préférable aux autres, est facile à réaliser grâce à une simple observation. Pour l'illustrer, pensez à un site web spécialisé pour les caméras, où les clients peuvent écrire leurs opinions sur un certain produit, comme illustré dans la figure . Le titulaire d'un avis pourrait attirer l'attention tant pour les aspects positifs et négatifs d'un certain produit, le tout dans le même texte (avis). En outre, l'approche de fractionnement de l'avis en phrases et en trouver le sentiment de chacun d'eux peut toujours ne pas être suffisant. Par exemple la phrase: «J'aime mon appareil photo et le zoom 24x, mais je pense que la vie de la batterie est trop courte ". Ici, il est facile de comprendre que la phrase est « plus positif que négatif», mais qui cachent encore un aspect négatif de la caméra en cours de discussion. Cela peut représenter un élément très important de l'information, qui peut être masqué en classant la phrase entière comme positive. Pour cette raison, la méthode explorée par (6) permet d'atteindre un niveau de granularité optimale car elle traite chaque attribut de l'OuD avec les détails nécessaires. La partie présente une méthode explorée par (6) et (5) pour trouver l'orientation des mots d'opinion.

2.3.3.1 Identification des mots de sentiment

Les mots d'opinion codent un état émotionnel, qui peut être désirable ou indésirable.les mots d'opinion qui codent les états souhaitables (beau, gentil, heureux, génial) ont une orientation positive, tandis que ceux qui codent les états indésirable (mauvais, terrible, décevant) ont une orientation négative. Comme déjà discuté, les mots d'opinion peuvent appartenir à plusieurs groupes syntaxiques, mais ils sont généralement les adjectifs et les adverbes. Dans (5), et (6) une solution simple et efficace a été proposé pour trouver l'orientation des mots d'opinion. Les auteurs ont utilisé une liste avec certains adjectifs et leurs orientations respectives annotées. L'idée est d'utiliser Wordnet (un système de référence lexicale en ligne qui organis des mots dans des ensembles synonyme, appelés synsets), pour rechercher des mots trouvés dans les opinions, et enrichir la liste avec les nouveaux mots trouvés. Dans WordNet, les adjectifs sont organisés comme des grappes bipolaires, comme illustré dans la figure 5. Le cluster fast/slow est constitué de deux moitiés de cluster .fast et son antonyme slow sont appelés head synsets .Chaque head synset a un *satellite synsets* a qui lui sont associés, qui sont des significations pour le head synset correspondant. En outre, la flèche en pointillés dans la figure représente l'association de fast avec son antonyme slow.

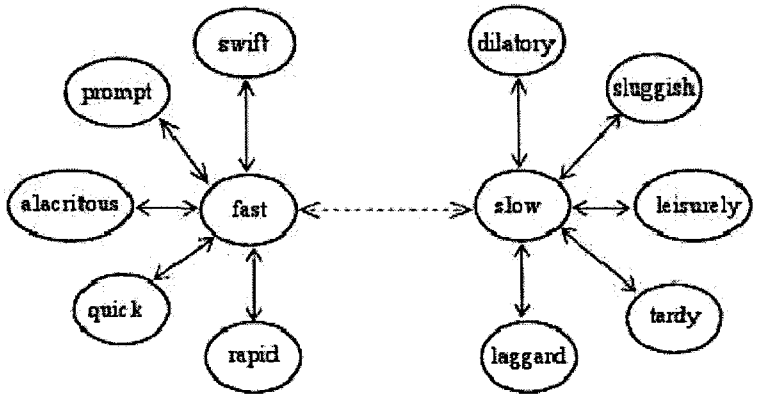

Figure 5:Structure Bipolaire des adjectifs (5)

Pour chaque nouveau mot trouvé (qui n'est pourtant pas dans la liste), le système recherche dans WordNet pour des synonymes possibles. Si tout synonyme trouve une correspondance dans la liste et parce que les synonymes sont des mots différents avec la même signification, le système comprend ce mot dans la liste en lui donnant la même orientation que le synonyme dans la liste. Si aucun synonyme n'est trouvé, alors le système recherche un antonyme. Si un antonyme existe et il a une correspondance dans la liste, le nouveau mot trouvé est inclu. Toutefois, en raison des antonymes qui ont une signification opposée, la même règle est appliquée à l'orientation qui sera également l'orientation opposée comme la correspondance trouvée dans la liste. Pendant ce processus, la liste va croître.les mots avec aucun correspondance peuvent tomber dans l'un des deux cas :

(1) Ils n'ont aucune correspondance dans la liste, et par conséquent ils devraient être annotés manuellement plus tard. (2) Le mot dépend du contexte, donc la partie du système qui gère les mots dépendants du contexte va décider de son orientation.

Ce processus peut être mieux remarqué dans la figure7. Dans leur travail, la liste a débuté avec 30 adjectifs, des adjectifs positifs (grand, fantastique,) et les adjectifs négatifs (mauvais, ternes).

2.3.3.2 Déterminer le sentiment d'opinions au niveau des phrases

Une opinion peut être analysée à différents niveaux de granularité. La figure 6 présente une pseuco-code qui vise à trouver le sentiment d'opinions au niveau de la phrase. La partie suivante, va analyser le sentiment d'une opinion au niveau de caractéristique, telle que proposée par (6), ce qui a la plus grande importance pour ce travail.

Règles de négation

Un mot de négation comme non, pas et n'a jamais et aussi quelques mots qui suivent des modèles changent l'l'orientation des mots l'opinion de la façon suivante:

I. Négation Négatif ->Positif

II. Négation Positif ->Négatif

III. Négation Neutre ->Négative

Quelques exemples (en anglais) pour chaque règle de la négation définie ci-dessus:

I. "no problem"

II. "not good"

III. "does not work"

Règles avec *TOO(en anglais)*

Le mot "too" est habituellement utilisé pour donner une connotation négative s'il est trouvé avant les adjectifs. L'idée est d'appliquer cette règle à des mots qui ont des orientations dépendantes du contexte.

I. "The battery life lasts long"

II. "The initialization time takes too long."

Donc, chaque fois qu'un mot *too* se trouve avant un mot d'opinion, l'orientation de l'avis devrait être négative.

2.3.3.3 Déterminer le Sentiment de l'opinion au niveau des caractéristiques

Dans (6), après avoir identifié tous les mots d'opinion pour une caractéristique donnée, le système calcule l'orientation d'opinion pour chaque caractéristique en utilisant l'équation suivante:

$$Score(f) = \sum_{w_i : w_i \in S \wedge w_i \in V} \frac{w_i.SO}{dis(w_i, f)}$$

S est une phrase avec un ensemble de caractéristiques

w_i est un mot d'opinion

V est l'ensemble des mots d'opinion (y compris les expressions idiomatiques)

w_i.SO est l'orientation sémantique du mot d'opinion w_i

dis (w_i, f) est la distance entre la caractéristique f et le mot d'opinion w_i dans la phrase S

Pour une phrase S qui contient un ensemble de caractéristiques et pour chaque caractéristique f le score d'orientation ci-dessus est calculé. Un mot positif est attribué une orientation +1 et un négatif est attribué -1. Dans l'équation ci-dessus w_i est un mot d'opinion, V est l'ensemble de tous les mots l'opinion et s est la phrase qui contient la caractéristique f, dis (w_i, f) est la distance entre f et le mot d'opinion w_i dans la phrase. Le pseudo code de la figure 7 a été utilisé pour trouver l'orientation de l'opinion au niveau caractéristique.

```
1.    Procedure SentenceOrietation()
2.    begin
3.      for each opinion sentence s_i
4.      begin
5.        orientation = 0;
6.        for each opinion word op in s_i
7.          orientation += wordOrientation(op, s_i);
8.          /*Positive = 1, Negative = -1, Neutral = 0*/
9.        if (orientation > 0) s_i's orientation = Positive;
10.       else if (orientation < 0) s_i's orientation = Negative;
11.           else {
12.            for each feature f in s_i
13.             orientation +=
14.             wordOrientation(f's effective opinion, s_i);
15.           if (orientation > 0)
16.             s_i's orientation = Positive;
17.           else if (orientation < 0)
18.                 s_i's orientation = Negative;
19.               else s_i's orientation = s_{i-1}'s orientation;
20.         }
21.     endfor;
22.   end

1.    Procedure wordOrientation(word, sentence)
2.    begin
3.      orientation = orientation of word in seed_list;
4.      If (there is NEGATION_WORD appears closely
          around word in sentence)
5.        orientation = Opposite(orientation);
6.    end
```

Figure 6:Pseudopode de l'orientation d'opinion s'une phrase (5)

```
1.  Algorithm OpinionOrietation()
2.    for each sentence s_i that contains a set of features do
3.      features = features contained in s_i;
4.      for each feature f_j in features do
5.        orientation = 0;
6.        if feature f_j is in the "but" clause then
7.          orientation = apply the "but" clause rule
8.        else remove "but" clause from s_i if it exists;
9.          for each unmarked opinion word ow in s_i do
10.               // ow can be a TOO word or Negation word as well
11.             orientation +=wordOrientation(ow, f_j, s_i);
12.         endfor
13.       endif
14.       if orientation > 0 then
15.         f_j's orientation in s_i = 1
16.       else  if orientation < 0 then
17.               f_j's orientation in s_i = -1
18.           else
19.             f_j's orientation in s_i = 0
20.           endif
21.       endif
22.       if f_j is an adjective then
23.         (f_j).orientation += f_j's orientation in s_i;
24.       else  let o_ij is the nearest adjective word to f_j, in s_i,
25.             (f_j, o_ij).orientation += f_j's orientation in s_i;
26.       endif
27.     endfor
28.   endfor;
      // Context dependent opinion words handling
29.   for each f_j with orientation = 0 in sentence s_i do
30.     if f_j is an adjective then
31.       f_j's orientation in s_i = (f_j).orientation
32.     else  // synonym and antonym rule should be applied too
33.         let o_ij is the nearest opinion word to f_j, in s_i,
34.         if (f_j, o_ij) exists then
35.           f_j's orientation in s_i = (f_j, o_ij).orientation
36.         endif
37.     endif
38.     if f_j's orientation in s_i = 0 then
39.         f_j's orientation in s_i = apply inter-sentence
                conjunction rule
40.     endif
41.   endfor
```

Figure 7:pseudo code de l'orientation d'opinions des caractéristiques du produit (6)

CHAPITRE III :

LE WEB SEMANTIQUE

Le Web sémantique (12) est une évolution continue du web en une infrastructure plus puissante et plus réutilisable pour le partage de l'information et de la gestion des connaissances. L'étape pour aller à partir du Web vers le Web sémantique est l'étape du passage du traitement manuel des informations au traitement automatique de l'information. Ainsi, le Web sémantique peut être vu comme un facteur clé de l'ère du savoir (13).

Le Web actuel est une plateforme de publication qui nous permet de se connecter avec des sources d'information arbitraires. Cependant, le Web est simplement une infrastructure d'édition de documents et de liens avec très peu de considération accordée au contenu ou la signification des documents ou à la signification de ces liens.

En conséquence, le Web sert comme un excellent géant référentiel de documents qui permet la fourniture de services en ligne, mais la réutilisation des connaissances est limitée car aucune norme uniforme n'est disponible pour exprimer le sens ou l'utilisation prévue des éléments de l'information.

Le Web sémantique est un web d'information qui est plus compréhensible et plus utilisable par les machines que le Web actuel. Ce n'est pas un web distinct, mais c'est une couche invisible (pour les utilisateurs) sous le web actuel dans lequel les informations et services ont de sens, ce qui permet aux agents logiciels de comprendre les demandes des personnes permettant aux ordinateurs et aux personnes de travailler en coopération. Le Web sémantique est activé par des technologies et des normes qui sont non seulement la représentation syntaxique de documents (comme le HTML), mais aussi leur contenu sémantique.

Le Web sémantique peut être considéré comme semblable à une grande base de données en ligne contenant des informations structurées qui peuvent être interrogées. Cependant, contrairement aux bases de données traditionnelles (14):

- Les informations peuvent être hétérogènes: il n'a pas besoin de se conformer à un schéma unique.
- Les informations peuvent être contradictoires, incohérentes et incomplètes.
- Les ressources ont des identificateurs globaux permettant de les lier pour former un « web sémantique » global.

L'épine dorsale du Web sémantique est l'ontologie (15). Les ontologies sont passées de la philosophie à l'intelligence artificielle, qui vise à faciliter le partage et la réutilisation des connaissances entre les hommes et les machines. En tant que structure de connaissances abstraites, les ontologies doivent être concrètement représentées pour les machines pour se

comprendre mutuellement. Dans le Web sémantique, cette interopérabilité est facilitée par les récentes normalisations W3C, notamment RDF, RDFS, OWL et SPARQL.

3.1 les Ontologies

En philosophie, l'ontologie est la branche de la métaphysique concernés par ce qui existe et des catégories de l'être (16) (17), (18).

Gruber identifie que, dans la gestion des connaissances, ce qui « existe » est celui qui peut être décrit. La définition la plus acceptée de l'ontologie comme un terme technique en informatique, redéfinie par Studer (19) et plus tard par Sure et Studer (20) , cité dans la définition suivante :

Définition : Une ontologie est une explicite, spécification formelle d'une conceptualisation partagée d'un domaine d'intérêt.

En termes pratiques, les ontologies sont des structures formelles qui permettent une compréhension partagée de certains domaines qui peuvent être communiquées entre les gens et les machines.

Guarino présente une étude exhaustive des définitions alternatives ainsi que des discussions (18). Une ontologie sert des objectifs différents qu'une base de connaissances: une ontologie partagée doit seulement décrire un vocabulaire pour décrire un domaine, alors qu'une base de connaissances peut inclure les connaissances nécessaires pour résoudre un problème ou répondre à des requêtes arbitraires sur un domaine (17). Suret Studer élabore sur chaque aspect de la définition de l'ontologie (20):

- « Formelle » : l'ontologie doit être lisible par la machine.
- « Partagée» c'est à dire qu'elle n'est pas privée à quelques individus, mais acceptée par un groupe.
- La référence au « domaine d'intérêt » indique que pour les ontologies de domaine on n'est pas intéressé à la modélisation du monde entier, mais plutôt la simple modélisation des concepts qui sont pertinents pour la tâche à accomplir.

Les ontologies se composent de plusieurs composants principaux (21) :

1. Instances
2. Classes
3. Attributs
4. Relations

Instances

Ces sont les fondamentaux, les composantes « de bas niveau » d'une ontologie. Les instances dans une ontologie peuvent inclure des choses concrètes comme Sarah, Nicolas, le Nokia N80, la Toyota Corolla grise stationnée à l'extérieur, et la planète Mars, ainsi que des choses abstraites comme l'événement de big bang, le contexte dans lequel ces mots ont été écrits et le moment de leurs déroulements dans le temps. Strictement parlant, une ontologie ne doit pas inclure toutes les instances, mais l'un des objectifs généraux d'une ontologie est de fournir un moyen de classer les instances, même si ces instances ne font pas explicitement partie de l'ontologie.

Classes

Définis comme des groupes abstraits, ensembles, ou des collections d'instances. Les classes peuvent classer les instances, d'autres classes, ou une combinaison des classes. Quelques exemples de classes sont: « Personne», la classe de toutes les personnes; « événement », la classe de tous les événements;« véhicule », la classe de tous les véhicules; « voiture », la classe de toutes les voitures; « classe », représentant la classe de toutes les classes, et « Thing», représentant la classe de toutes choses. Surtout, une classe peut subsumer ou être subsumé par d'autres classes, une classe subsumée par une autre est appelée une sous-classe de la superclasse.

Par exemple, « véhicule »subsume « voiture », puisque (forcément) tout ce qui est un membre de la dernière classe est un membre de l'ancienne. La relation de subsomption est utilisée pour créer une hiérarchie de classes, généralement avec une classe générique au maximum comme `Thing" en haut, et une très spécifique classes comme « Toyota Corolla VVTi modèle 2003 "en bas.

L'importante conséquence de la relation de subsomption est l'héritage des propriétés de la superclasse par la classe. Ainsi, tout ce qui est nécessairement vrai d'une superclasse est aussi nécessairement vrai de toutes ses sous-classes.

Dans certaines ontologies, une classe est autorisée à avoir seulement une superclasse (héritage simple), mais dans la plupart des ontologies, les classes sont autorisées à avoir un certain nombre de superclasses (héritage multiple). Dans ce dernier cas toutes les propriétés nécessaires de chaque superclasse sont héritées par les sous-classes. Ainsi, une classe particulière d'animaux – « Chat de maison » - peut être une sous-classe de la classe «chat »et aussi une sous-classe de la classe « animaux ».

Attributs

Les instances et les classes dans une ontologie peuvent être décrites en les rapportant à d'autres instances, des classes ou localement à des valeurs de données. Depuis que les valeurs données représentent des aspects ou propriétés confiées à l'instance ou la classe décrite ils sont considérés comme des attributs internes de cette instance ou de classe.

Relations

Les instances et les classes dans une ontologie peuvent aussi être décrites en les rapportant à d'autres instances et classes. Dans ce cas, les relations qui connectent des composants discrets externes les uns aux autres et ne sont donc pas considérés comme des attributs locaux confiés au composante. Typiquement une relation elle-même est d'une classe particulière qui spécifie dans quel sens l'élément est lié à l'autre composante de l'ontologie.

Une grande partie de la puissance des ontologies vient de la capacité à décrire les relations. L'ensemble des classes de relations utilisées et leur hiérarchie de subsomption déterminent la puissance expressive de l'ontologie. Le type le plus important de la relation est la relation de subsomption. Ceci définit quels sont les objets qui sont classifiés par tel classe. Par exemple, nous avons déjà vu que la classe « Toyota Corolla » est une sous-classe de « voiture »', qui à son tour est une sous-classe de « véhicules ».

Un autre type courant de la relation est la relation « méronymie », qui représente la manière dont les objets se combinent pour former des objets composites. Par exemple, si l'ontologie a été étendue pour inclure des concepts tels que « Volant », nous dirions que le « volant » est une partie d'une « Toyota Corolla». Si nous introduisons les relations « méronymie », à notre ontologie, nous trouvons que cette arborescence simple devient rapidement complexe et significativement plus difficile à l'interpréter manuellement.

C'est la raison de permettre aux machines d'interpréter les connaissances automatiquement et faciliter les tâches pour l'utilisateur. Les conceptualisations partagées de domaines d'intérêt doivent premièrement, être formellement spécifiées dans un modèle de données explicite adapté au traitement de la machine. Le W3C envisage une suite de technologie pour la réalisation d'un tel modèle de données explicite pour le Web sémantique, comme l'illustre la figure 8.

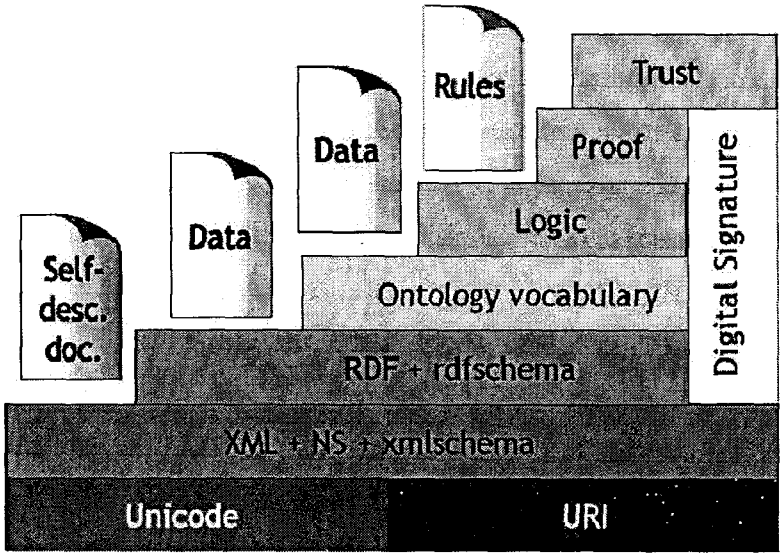

Figure 8: Architecture du web sémantique

Comme on le voit, XML ainsi que XML schéma forment la deuxième couche au-dessus des URIs et l'Unicode dans la pile de technologie. La troisième couche est RDF et RDFS. Les vocabulaires de l'ontologie sont construits sur cette base. Au sommet des vocabulaires, il y a un besoin d'une ontologie pour exprimer la logique, de sorte que l'information peut être déduite ainsi qu'une meilleure mise en relation. Une fois qu'il ya une logique, il est logique de s'en servir pour prouver des choses. La couche de la preuve permet à chacun d'écrire des déclarations logiques, et un agent peut suivre ces liens sémantiques pour construire des preuves, de sorte que la validité d'une déclaration, en particulier une déclaration déduite, peut être vérifiée.

3.2 RDF

Le modèle de données fondamental du Web sémantique est le Resource Description Framework (RDF) (22). Les ressources peuvent être soit des instances ou des classes dans la terminologie de l'ontologie, et donc « ressource» peut être considérée comme synonyme de la superclasse commune de toutes les instances et les classes. RDF est alors un modèle de données sémantique pour affirmer une déclaration arbitraire des ressources identifiables dans une tentative pour répondre aux limites sémantiques du langage XML.

Ce modèle de données est constitué de noeuds reliés par des arcs étiquetés, où les nœuds représentent des ressources (des instances ou des classes dans la terminologie de l'ontologie) et les arcs représentent les propriétés de ces ressources (attributs ou les relations dans l'ontologie).L'utilisation de Uniform

Resource identificateurs (URIs) permet aux différentes sources de se relier, formant finalement un graphe d'états.

Une déclaration typique de RDF est constituée de trois éléments synonymes du sujet, prédicat et objet dans une phrase en langage naturel simple. Une telle déclaration est considérée comme un triplet: une ressource (le sujet) liée à une autre ressource (l'objet) par un arc étiqueté avec une troisième ressource (le prédicat). Par conséquent, un triplet unique est une déclaration selon laquelle un sujet (par exemple une photographie, une personne) a une relation (par exemple " a un créateur ", " a un nom") à un objet (par exemple, une personne, la valeur Sam).

Les ressources peuvent être identifiées par un URI ou peuvent être non identifié, dans ce cas elles sont considérés comme des "nœuds vides ". Un URI peut identifier une ressource réelle comme un livre imprimé, mais peut aussi être le Uniform Resource Locator (URL) d'un document numérique ou même d'un élément dans un document, dans ce cas l'URI se compose de l'élément URN (Uniform Resource Name) apparaitrait à la fin de l'URL du document. Seules les ressources (identifiées ou non identifiées) peuvent être l'objet d'une déclaration; seule les ressources identifiées peuvent être un prédicat d'une déclaration et toutes les ressources ou de littéraux (valeurs de données) peuvent être l'objet d'une déclaration.

Un exemple pour un triplet unique est la déclaration suivante: « photo2011.jpg a un créateur Sarah "Cet exemple de déclaration RDF est illustré par la figure 9 où il peut être vu que la ressource est photo2011.jpg identifié par un URI qui est l'URL. Indiquant son emplacement, alors que la ressource Sarah est identifiée par l'URI qui se compose de l'URN soh qui identifie le nœud de Sarah dans le document RDF situé à l'URL http://sfn.fr/foaf.rdf. Le symbole # est utilisé pour ajouter des URN à des URL de cette manière. Le prédicat est identifié avec l'URI http://bbk.ne/dc/terms/creator.

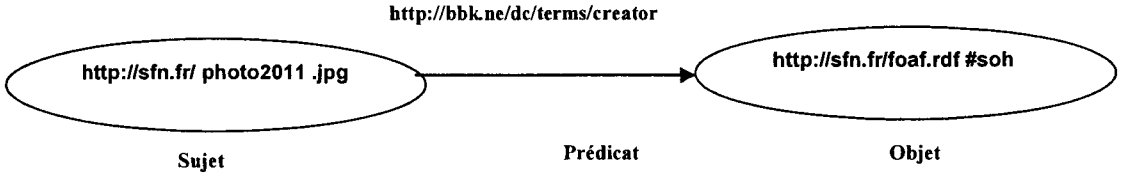

Figure 9: Triplet RDF

RDF est un modèle abstrait de données et peut être sérialisé dans plusieurs formats tels que RDF / XML (23) , N3 (24) ou N-Triples (25) (dont seulement RDF /XML est officiel approuvé par le W3C).

```
<rdf:RDF
xmlns:rdf =" http: // www.w3.org /1999/02/22 - rdf -syntax -ns#"
xmlns:dcterms =" http://purl.org/dc/ terms /"
xmlns:foaf =" http: // xmlns .com/ foaf /0.1/ ">
<foaf:Image rdf:about =" http://sfn.fr/ photo2011 .jpg ">
< dcterms:created >20011 -08 -01 T21:01:21Z </ dcterms:created >
< dcterms:creator >
<foaf:Person rdf:about =" http://sfn.fr/foaf.rdf #soh ">
<foaf:name >Sarah </ foaf:name >
</ foaf:Person >
</ dcterms:creator >
</ foaf:Image >
</ rdf:RDF >
```

Figure 10:Exemple RDF/XML

Les Déclarations RDF sont relativement simples et explicites: il ya peu de sémantique implicite. Oren Déclare que « RDF peut être considéré comme un alphabet, permettant de construire des mots et des phrases, mais pas encore une langue, car les mots et les phrases n'ont pas encore donné de sens» (26). Une telle signification utilisable par l'ordinateur doit être définie par un vocabulaire pour RDF qui spécifie la sémantique des termes .Actuellement, il ya deux tels vocabulaires normalisés: RDF Schema (RDFS) et le langage d'ontologie Web (OWL).

3.3 RDF Schema

RDF Schema (RDFS) (27) est un vocabulaire pour RDF : il définie les termes, leurs usage prévu et leur sémantique. Ces terme permettent la définition de classes, une hiérarchie sous-classe et la classe d'appartenance des ressources. Ils permettent également la définition de propriétés, de leur domaine et d'une hiérarchie de sous-propriété. RDFS est utilisé pour définir RDF, RDFS lui-même et tous les autres schémas de vocabulaire RDF, par exemple le Dublin Core (27) et Friend-f-A-Friend (FOAF) (28). En partageant RDF schémas, la réutilisabilité des définitions de métadonnées peuvent être supportée (29) .

RDFS offre également la propriété spéciale rdfs: subClassOf qui définit la relation sous-classe entre les classes dans un schéma. Depuis cette propriété est définie comme transitive, les définitions sont héritées par les classes le plus spécifiques à partir des classes les plus génériques:

Les ressources qui sont membres d'une classe sont implicitement membres de toutes les superclasses de cette classe. De même, rdfs: subPropertyOf définit une hiérarchie de propriétés. RDFS permet également des restrictions de domaine et de la gamme pour les propriétés. Par exemple, la propriété dcterms:Creator pourrait être limitée à la fois au domaine foaf: image et la gamme foaf: Person de sorte que: (i) n'importe quoi avec Creator doit être une image, et (ii) tout Creator doit être Person.

Enfin, RDFS définie certains termes pour la description informelle des classes et des propriétés, telles que rdfs: comment, rdfs: label et rdfs: seeAlso qui sont destinées à des informations lisibles par l'humain. Elles sont importantes pour les développeurs de schémas qui peuvent les utiliser pour communiquer le sens et l'usage prévus des classes et des propriétés aux développeurs d'applications ou d'autres développeurs de schémas qui réutilisent un schéma, qui est le but même de l'existence du schéma.

3.4 OWL

OWL (Web Ontology Language) (30) est un langage de description d'ontologies conçu pour la publication et le partage d'ontologies en web sémantique.

Le langage d'ontologie Web OWL est conçu pour des applications qui doivent traiter le contenu des informations plutôt que de simplement les présenter aux humains. Le langage OWL offre aux machines de plus grandes capacités d'interprétation du contenu Web que celles permises par XML, RDF et le RDF schéma (RDF-S), grâce à un vocabulaire supplémentaire et une sémantique formelle.

Ce langage y ajoute plus de vocabulaire pour décrire les propriétés et les classes entre autres, les relations entre les classes, cardinalité, égalité, typage de propriétés plus riche, caractéristiques des propriétés et les hiérarchies des propriétés et des classes.

OWL possède des sous-langages de plus en plus expressifs OWL Lite, OWL DL et OWLFull.

- OWL Lite : Il a été conçu pour être utilisé dans des situations qui n'ont besoin que des hiérarchies de classification et des caractéristiques de contraintes simples. Pour les contraintes de cardinalité, seules les valeurs 0 et 1 sont permises.

- OWL DL : Il est plus expressif qu'OWL Lite et est basé sur la description logique. Il est destiné aux utilisateurs qui demandent un maximum d'expressivité tout en maintenant la complétude (garantie de calculer toutes les conclusions) et la décidabilité (tous les calculs

doivent finir en un temps fini). OWL DL contient tous les constructeurs du langage OWL mais sont utilisables avec des restrictions.

- OWL Full : Il est le plus expressif des sous langages d'OWL. Il est destiné aux utilisateurs qui demandent un maximum d'expressivité avec la liberté syntaxique de RDF sans aucune garantie de calcul. Par exemple, une classe peut être traitée comme une collection d'individus et en même temps peut être vue comme un seul individu.

OWL Full permet aussi à une ontologie d'augmenter le sens du vocabulaire prédéfini (RDF et OWL).

Ainsi, les fonctionnalités de OWL Lite sont inclues dans celles de OWL DL. De même, OWL Full inclut toutes les fonctionnalités d'OWL DL.

En résumé, OWL Lite ⊂ OWL DL ⊂ OWL Full.

3.5 SPARQL

SPARQL est un langage de requête recommandé par leW3C .SPARQL est capable d'interroger des informations représentées par des graphes RDF. La requête SPARQL, dans sa forme élémentaire, est constituée d'un ensemble de patrons de triplets appelé un patron de graphe élémentaire (basic graph pattern). Les patrons de triplets sont comme les triplets RDF sauf qu'un sujet, un prédicat et un objet peuvent être des variables. Le tableau présente les résultats de la requête qui interroge le graphe de l'exemple du Listing précédent qui contient une seule image.

```
PREFIX rdf: <http: // www .w3.org /1999/02/22 - rdf -syntax -ns#>
PREFIX dcterms: <http: // purl .org /dc/ terms />
PREFIX foaf: <http: // xmlns .com / foaf /0.1/ >
SELECT ? image ? name WHERE {
? image rdf:type foaf:Image .
? image dcterms:creator ? creator
< http://sfn.fr/ photo2011.jpg > dcterms:creator ? creator .
? image dcterms:created ? created .
FILTER regex (? created , " 2008 ") .
OPTIONAL {? creator foaf:name ? name } }
```

Figure 11: Exemple d'une requête SPARQL

image	name
http://sfn.fr/photo2011.jpg	\Shara"

Figure 12:Résultat de la requête SPARQL qui interroge le graphe RDF

CHAPITRE IV :
LES OUTILS DISPONIBLES

Les outils présentés dans ce chapitre ne sont pas les seuls outils disponibles, ni forcément les meilleurs outils du domaine. Ils sont cités ici uniquement à titre d'exemple de la richesse logicielle qui accompagne RDF et OWL.

4.1 Editeur d'ontologies Protégé

Protégé (31) est un éditeur d'ontologies distribué en open source par l'université en informatique médicale de Stanford. Protégé n'est pas un outil spécialement dédié à OWL, mais un éditeur hautement extensible, capable de manipuler des formats très divers. Le support d'OWL, comme de nombreux autres formats, est possible dans Protégé grâce à un plugin dédié.

4.1.1 Définition des classes et des propriétés

L'interface de Protégé est assez simple, l'ensemble des fonctionnalités de l'éditeur étant regroupé en huit onglets. Il est possible de créer, modifier, supprimer une classe, et de lui attacher des propriétés. Ces propriétés peuvent elles-mêmes être caractérisées. La figure 13 représente la capture de l'écran principal de Protégé.

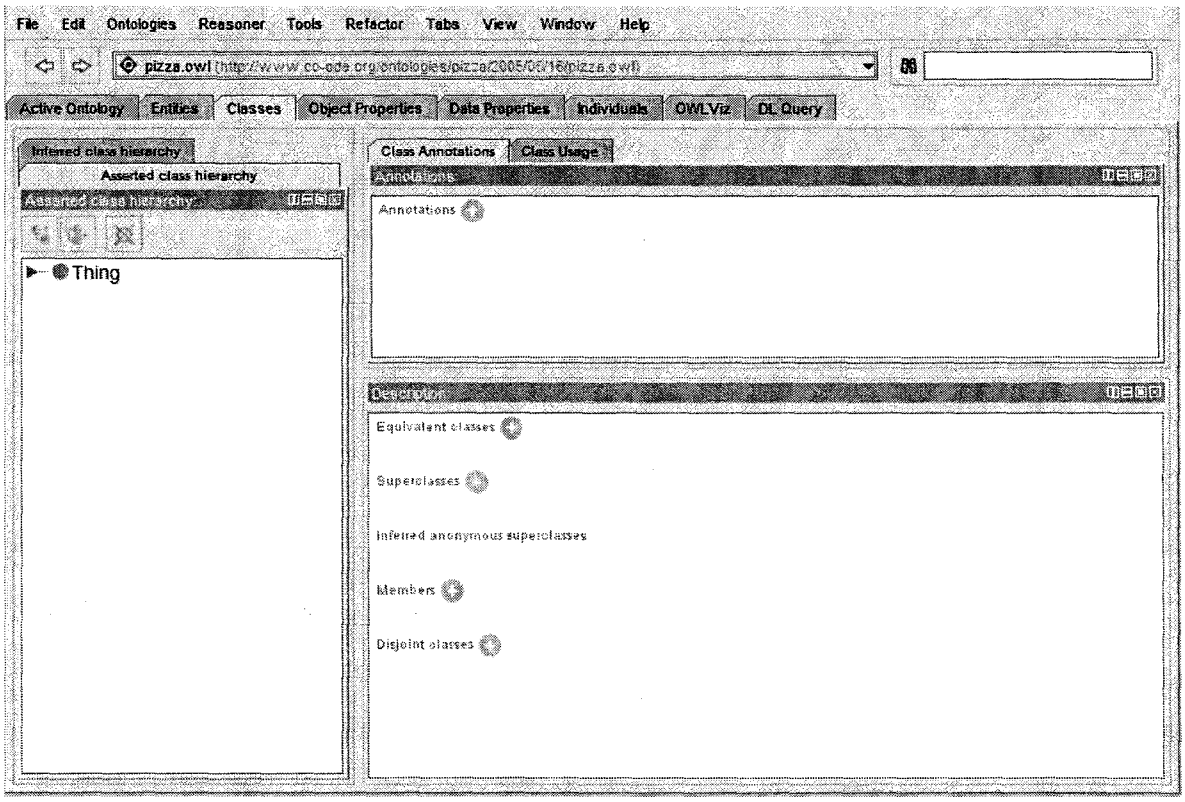

Figure 13:L'écran principal de Protégé

4.1.2 Gestion des instances de classe et de leurs propriétés

Il est possible de créer des instances et de leur affecter des propriétés, conformément à la définition des classes et des propriétés effectuée dans l'onglet « Classes ». Il est important de comprendre que, dans l'instanciation, un individu est désigné par son identifiant (son «rdf:ID »).

4.1.3 Possibilité d'effectuer des requêtes

Enfin, une fonctionnalité intéressante de Protégé concerne la possibilité d'effectuer des requêtes sur l'ontologie en cours d'édition.

4.2 Framework Jena

Jena (32) est un framework écrit en Java, dont l'objectif est de fournir un environnement facilitant le développement d'applications dédiées au web sémantique. Jena permet de manipuler des documents RDF, RDFS et OWL, et fournit en plus un moteur d'inférence permettant des raisonnements sur les ontologies.

4.3 OWL validator

Une fois qu'un document OWL est écrit, il faut s'assurer de sa validité et de la cohérence des concepts qu'il exprime. Il existe différents validateurs d'ontologies OWL. Certains valident uniquement la syntaxe du document, tandis que d'autres vérifient également la cohérence des informations contenues dans l'ontologie.

Le validateur RDF du W3C (http://www.w3.org/RDF/Validator/) permet de valider des documents RDF. Il permet donc également de s'assurer qu'un document OWL respecte la syntaxe de RDF, ce qui donne déjà une première indication de la validité d'une ontologie.

4.4 KIM – Semantic Annotation, Indexing, and Retrieval

KIM (http://www.ontotext.com/kim) est une plate-forme d'annotation, d'indexation sémantique,et la récupération d'information. Il permet l'annotations (semi-)automatique pour le Web sémantique, en utilisant la technologie d'extraction d'information (IE). KIM est basé sur deux principales plates-formes, il combine GATE (33) et Sesame/OMM[1] afin de combler l'écart entre les résultats courant d'IE et les exigences du Web sémantique.

[1] OMM (the Ontology Middleware Module), http://sesame.aidministrator.nl, http://www.ontotext.com/omm et http://www.ontoknowledge.org

Les principaux objectifs peuvent être résumés comme suit:

- Faire l'extraction de connaissances sémantiquement bien fondées à partir du texte. Techniquement, cela signifie la création d'annotations liées à des classes et instances formelle de l'ontologie, exprimée en RDF (S) (ou langage compatible);
- De permettre à IE de bénéficier de l'ontologie formelle et la représentation des connaissances, principalement pour la désambiguïsation;
- Pour faire la récupération possible des documents de texte basé sur la connaissance, qui comprend l'information qui à besoins d'un une satisfaction, qui est actuellement fournie à un mode incompatible de trois technologies différentes - le SGBD, information Retrival , et IE. Un exemple d'une requête : "donner moi, classés par pertinence, tous les documents relatifs à l'entreprise impliquée dans un accident en France, qui a eu lieu en Novembre 2010 ";
- Pour fournir des moyens pour le développement d'un Web sémantique dynamique - KIM permet l'annotation automatique du contenu sur le serveur.

Pour atteindre les objectifs ci-dessus, KIM s'appuie sur énorme instances des données et des thesaurus des informations représentées en RDF (S). Le système est basé sur l'ontologie de niveau supérieur appelé KIMO qui a environ 300 classes couvrant de façon sémantiquement les types d'entités les plus importantes et fournissant un terrain pour (i) l'expansion afin d'inclure des connaissances plus complexes comme les relations, les scénarios, événements, (ii) domaine ou une tâche spécifique à la connaissance et (iii) l'intégration avec des tiers / systèmes d'information client.

4.4.1 Annotation sémantique

Les annotations sémantiques offertes par KIM sont assez proches de la sortie de la reconnaissance des entités nommées offertes par de nombreux systèmes existants d'IE. La différence majeure est que l'information sémantique appropriée est gardée pour le type de l'entité (via l'URI d'une classe de l'ontologie) combiné avec des références à des informations spécifiques à une méta-donnée sur l'entité elle-même, comme l'illustre le diagramme ci-dessous.

Bien que différentes conventions de codage des types d'annotations soient présentes dans les systèmes d'IE, ceux habituelles ont un manque de représentation des connaissances

appropriées et de cohérence, ainsi que la complétude de la taxonomie. C'est le problème qui a été ciblée et résolue dans KIM via l'extension et la réingénierie de GATE.

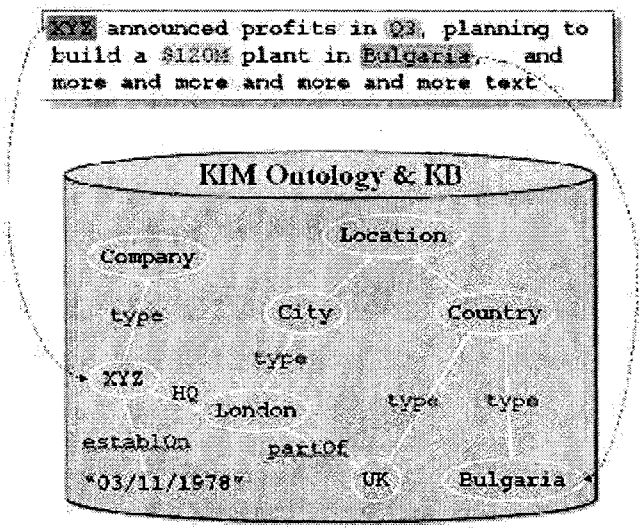

Figure 14:Annotation dans KIM

Comme présenté sur la figure 14, les annotations pour les entités ont des références, à savoir les URI, vers les ressources appropriées dans le référentiel RDF(S) portant l'ontologie KIM, KIM world KB, et toutes les connaissances sur les entités supplémentaires, soient importées d'une autre source officielle, soient automatiquement extraites de ce texte.

4.4.2 KIM Front-ends

Le fronts-ends de KIM offre à l'utilisateur final les avantages de Kim en forme simple et intuitive. Ils nécessitent une simple installation du serveur de KIM, qui coopère avec Sésame et utilise les outils d'IE basé sur GATE pour traiter les documents.

Ces outils démontrent comment une fois avoir les documents sémantiquement annotés, une visualisation à usage général est proposée, la navigation et des outils d'interrogation pourraient être utilisés en plus des composants d'interface spécialisés.

4.4.2.1 Exploration des Entités

Le Plug-in de KIM peut mettre les entités dans la page Web chargée, dans des couleurs correspondants à leurs classes. L'explorateur de KIM est un simple navigateur de méta-data, permettant à l'utilisateur de surfer des connaissances sur l'entité, via leurs représentations RDF(S).

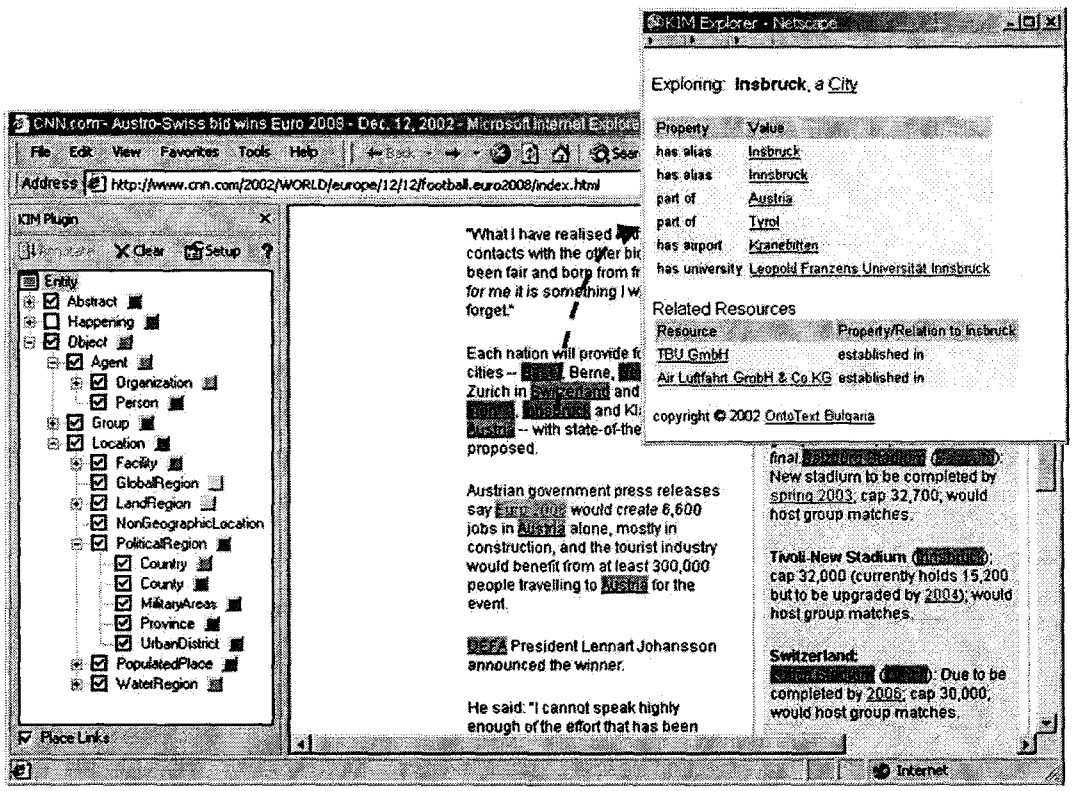

Figure 15: le plug-in KIM, et l'explorateur KIM (36)

Techniquement, le plug-in envoie le contenu de la page à un serveur KIM qui le traite et renvoie les annotations pour l'affichage. De cette façon, le plug-in est un petit module client. Tout le traitement réel est fait sur le serveur.

Pour chaque entité l'explorateur présente (i) les classes les plus spécifiques à laquelle elle appartient, (ii) ses propriétés et les relations aux autres entités, et enfin (iii) les entités qui lui sont liées. Toutes les autres entités sont en hyperlien, alors, ils peuvent être explorées davantage. Les abstractions à travers la représentation RDF(S) "natif" comprennent:

- Les ressources sont présentées avec leurs étiquettes, plutôt qu'avec les URI
- le nombre de propriétés «auxiliaire» est filtré.

4.4.2.2 Interrogation sémantique de KIM

Les requêtes sémantique de KIM permettent l'interrogation des entités selon des schémas arbitraires de la " base de connaissance» existante.

Un exemple

Donne-moi toutes les sociétés X, dont le nom contient "Bahn", impliqués dans des accidents en Europe dans la période 5-10.11.2002.

L'interface utilisateur est mis en forme de page HTML dynamique comme ci-dessous :

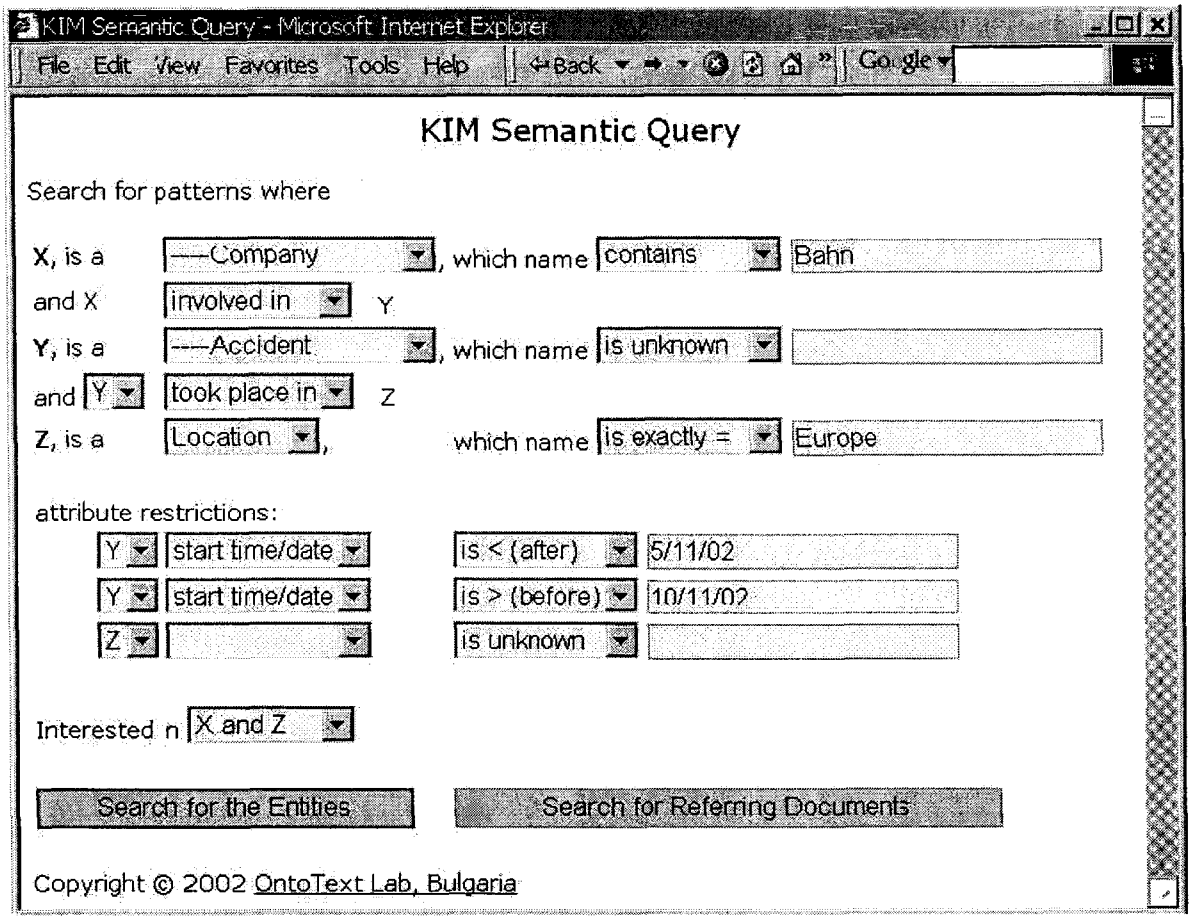

Figure 16: L'interface utilisateur d'interrogagtion de KIM

4.4.3 KIMO Ontology

KIMO couvre les 300 classes d'entité et 100 relations, avec les objectifs suivants:

- niveau de base d'intelligence /reconnaissance de puissance pour l'analyse de texte général;
- meilleures performances pour les affaires et les nouvelles politiques;
- fournir des bases bien structurées pour l'extension avec le domaine et les ressources des applications spécifiques.

L'ontologie est constituée des classes sous la classe kimo:Entity et toutes les sémantiques liées à leurs descriptions et leurs relations. Il peut être considéré comme un niveau supérieur typique d'ontologie qui combine:

- Certaines distinctions philosophiques connus;
- L'expérience d'un certain nombre existant d'ontologies de niveau supérieur, telles que Cyc[2]2 et DOLCHE (voir (35));
- L'expérience des bases de connaissances lexicales, telles que WordNet et Euro Wordnet, y compris le projet OntoClean (voir (34)).

Les plus hauts niveaux de distinctions sont:

- kimo: Object - des entités pour lesquelles on pourrait dire qu'elles existent. Les objets peuvent jouer un certain rôle dans certains événements. Des Objets pourraient être importants (comme le Tour d'Eifel ou le corps de Lénine) ou immatériels (par exemple, un courant électrique entre deux points). Une de leurs importantes caractéristiques est que celles-ci peuvent occuper certaines régions dans l'espace.
- kimo: Happening - des entités pour lesquelles on pourrait dire qu'elles se déroulent. Elle peut être soit dynamique comme "dessiner un cercle" ou statique comme «être un président". Dans tous les cas, les événements ont un certain endroit dans le temps, un point de début et un point de fin dans les cas les plus simples.
- kimo: Abstract - entités qui ni existe, ni se déroule, par exemple Devise, un Théorème ou une sorte de sport.

4.4.4 KIM World Knowledge Base

Le KIM World KB a été construit avec l'objectif d'une couverture quasi-exhaustive des entités les plus importantes au monde, leurs noms, les relations et les propriétés. KIM " knows":

- les emplacements géographiques: montagnes, villes, routes, les océans, etc - plus de 5M des noms avec la sous-région, les relations entre eux;
- Les organisations, toutes sortes importantes: affaires, internationale, politique, gouvernement

  Les personnes spécifiques avec leurs positions et d'autres informations.

Le KIM World KB est utilisée dans KIM d'une façon assez similaire aux index (répertoires) qui sont utilisé dans les systèmes classiques d'IE. Pour chacune des entités, le

[2] http://www.cyc.com/cyc-2-1/cover.html et le project, http://www.opencyc.org

nombre d'alias est entretenus avec les informations correspondantes à leur sujet, par exemple des caractéristiques telles que le «langage». "Court / long", "officielle", "vieux", etc

Ce n'est pas une surprise qu'un tel répertoire étendue des informations augmente la phase de rappel de reconnaissance d'entités nommées, mais s'il n'est pas bien géré il va apporter des niveaux d'ambiguïté qui peuvent réduire la précision à des niveaux tout à fait inacceptable. Pour résoudre ce problème, KIM utilise le modèle de Markov caché. Une fois formé, le corpus est annoté manuellement.

CHAPITRE V :

SYSTEME D'ANALYSE

SEMANTISUE D'OPINION

Les produits ou les services sont souvent discutés par les clients sur le Web. Les sites des entreprises présentent habituellement les discutions des utilisateurs à propos des avantages ou des problèmes avec certains produits .Ces opinions constituent une source pour une meilleure compréhension d'un marché. Également dans la vie sociale commune, les communautés sur le Web peut avoir un fort impact sur le réglage de tendance.

5.1 Travaux existants

Des recherches récentes sur l'opinion mining se sont concentré sur l'analyse des sentiments, la classification simple "pro" et "contre" (36) et la détermination de l'orientation sémantique dans les modèles de l'opinion basées sur un mot, une phrase ou un document. Typiquement, le Traitement Automatique des Langues (TAL) (5) (9)et les techniques d'apprentissage automatique (36) (37) ont été utilisées dans les modes supervisés ou non supervisés (38), (39) permettant l'extraction et la classification du sentiment et la de polarisation des avis. Le workflow comprend généralement trois phases principales: l'extraction, la structuration et la récapitulation des résultats. En général, nous nous abonnant à ce modèle, toutefois varier dans un certain nombre de détails concernant principalement la représentation explicite de l'information.

Notre système est basé sur les technologies du Web sémantique (RDF, SPARQL, etc.).Nous utilisons largement les vocabulaires de Semantically-Interlinked Online Communities (SIOC) -ainsi que les API existantes (40) pour la phase d'extraction et de structuration. En ce qui concerne la représentation formelle des éléments de l'analyse et de leurs caractéristiques, il est intéressant de noter que le W3C a récemment lancé le «Product Modelling Incubator Group"3 visant à créer une ontologie de modélisation des produits.

5.2 Architecture du système

Nous avons constaté que des données pertinentes pour l'étude de marché sur le Web ne sont faciles ni en accès ni en traitement. Les coûts du temps pour recueillir et évaluer les données nécessaires à une meilleure compréhension du marché sont encore énormes. Comme l'a souligné Peter Mika (Yahoo! Recherche) (35):

La Technologie de recherche actuelle est incapable de satisfaire toutes les requêtes complexes nécessitant l'intégration des informations telles que l'analyse, de prévision, planification, etc .Un exemple de l'intégration basée sur des tâches est l'opinion mining au

[3] http://www.w3.org/2005/Incubator/w3pm/

sujet des produits ou services. Même s'il ya eu quelques succès dans l'opinion mining avec l'analyse pur des sentiments, il est souvent le cas que l'utilisateur voudrait savoir quels aspects spécifiques d'un produit ou service sont décrits en des termes positifs ou négatifs et d'avoir les résultats de recherche apparaissants regroupés et organisés.

Dans ce projet nous visons à développer une méthodologie permettant des études des opinions pour comprendre un certain marché.

L'analyse effectuée dans ce projet est faite en utilisant les informations disponibles sur le Web.

La figure 17 représente l'architecture globale du système, composée de (i) le module d'acquisition des données, (ii) le module d'analyse, et (iii) le module d'interrogation .les données et les informations sont rassemblées à partir du Web à travers des services. Nous avons développé des méthodes de conversion de contenu Web (HTML) en données structurées représentées en XML et RDF qui les permettent d'être à la fois flexibles et complètes.

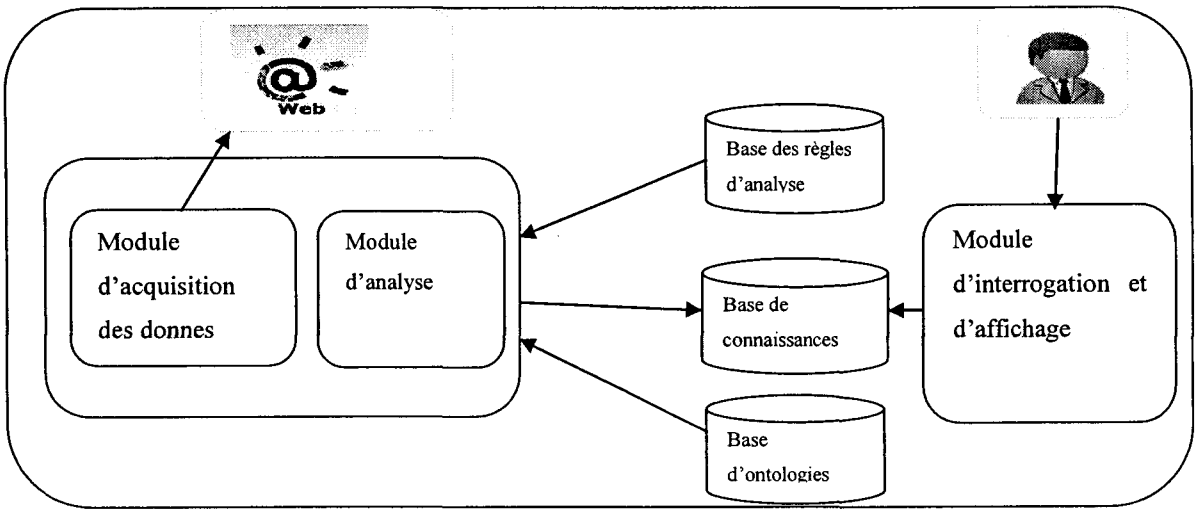

Figure 17:Architecture du Système d'analyse d'opinion

5.3 Représentions des commentaires et Opinions

Pour permettre des requêtes structurées et la navigation, il est nécessaire de représenter les discussions d'une manière interprétable et de les améliorer avec la sémantique.

Notre objectif est de modéliser d'une manière explicite l'opinion dans une discussion conforme aux données web. Nous avons décidé de réutiliser un vocabulaire existant pour représenter les opinions. En raison de sa popularité et son utilisation à grande échelle,

Semantically-Interlinked Online Communities (SIOC)4 a été sélectionné pour représenter les commentaires publiés.

Toutefois, dans le cas de représentation explicite d'opinions que nous n'avons pas trouvé un vocabulaire approprié. Nous avons définie notre propre vocabulaire dédié pour cette tâche.

Notre «Ontologie Opinion Mining" définit essentiellement les classes et les propriétés suivantes:

- OOM: DiscussionOpinion, le noyau central qui connecte des fils de discussion avec des opinions sur une certaine entité;
- OOM: opinion, une représentation abstraite d'une opinion;
- OOM: Sujet, un concept pour déclencher les aspects d'un certain sujet.

Nous utilisons SKOS: Concept de SKOS (42) pour représenter à propos de quoi une discussion est faite, par exemple, un certain film tel que *Matrix*. De plus, nous utilisons le vocabulaire sioc :Thread de SIOC pour indiquer le lieu .

Il est à noter que les OOM: opinion n'est pas étendu actuellement .On a l'intention d'étendre et de perfectionner cette partie de l'ontologie basée sur des travaux antérieurs de (39).En outre, nous voulons souligner que le concept OOM: Sujet est utilisé pour représenter un certain aspect de la discussion. Elle pourrait indiquer que les utilisateurs se discutent au sujet de la tarification, à propos des problèmes avec un certain produit ou tout simplement exprimer leur satisfaction. La sémantique de ce concept sont telles que si l'un des mots déclencheurs assignés a été trouvé dans une discussion, le sujet est censé correspondre (d'où l'étiquetage de la propriété type de donnée OOM: hasTrigger).

L'ontologie introduite joue un rôle déterminant dans notre processus d'opinion mining. Afin d'atteindre une meilleure évolutivité et la réutilisabilité, elle agit comme un lien entre le domaine et les données RDF. C'est pourquoi il n'y a aucune différence pour notre modèle d'opinion mining, s'il ya derrière une autre ontologie de domaines spécifiques. Par conséquent, notre approche offre une grande flexibilité par le choix du domaine et donne un avis générique.

[4] http://www.sioc-project.org/ontology

5.4 Discussion

Dans le processus de discussion du système, deux composants majeurs sont impliqués, à savoir (i) l'acquisition de données, où des discussions sont récoltées sur le Web, transformation en RDF et inter-liaison, et (ii) l'Analyseur, permettant d'interroger et d'accéder aux données.

5.4.1 Acquisition des données

L'acquisition de données est réalisée en trois phases: dans une première étape les données dans les discussions sur Internet, tels que titre, auteur, date de création, etc sont transformées en RDF utilisant SIOC. Dans une deuxième phase les entités clés dans les commentaires sont identifiées et interprétées sur un certain domaine (dans notre cas le domaine est «films»). Cette deuxième étape consiste à l'interconnexion aux ensembles de données liés tels que des instances de quelques autres ontologies de domaines spécifiques.

Dans une troisième phase les commentaires (subjectives) des participants sont analysés, et ajoutés à la base de connaissances analysée. Ceci est principalement réalisé par la création des instances OOM: DiscussionOpinion et leurs propriétés respectives. Ces sont les instances de l'OOM: Sujet qui déclenchent la création des opinions.

Nous avons mis en place un système client/serveur pour effectuer l'acquisition de données. Dans le cadre de notre recherche nous nous appuyons sur des sites autorités. L'extraction des données se fait automatiquement .Le serveur a été mis en œuvre en utilisant un serveur d'application Java (Tomcat).

Sur le côté client, une interface permettra à un utilisateur d'interroger la base des connaissances analysée sera développer.

5.4.2 Analyseur

L'Analyseur est une application permettant l'étude et l'analyse des données recueillies par le serveur d'acquisition. L'utilisateur peut limiter les données en sélectionnant certaines classifications de films en limitant par exemple la période de temps.

5.5 Conclusion

Dans ce chapitre nous avons proposé une nouvelle approche pour l'opinion mining sur le Web en utilisant des technologies Web sémantique. Notre but est de modéliser explicitement les commentaires trouvés sur le Web, nous avons développé un vocabulaire qui

représente ces opinions formellement (en RDF) et on a rapporté une mise en œuvre d'une partie de cette approche.

Nous allons la contempler à l'aide d'une ontologie GoodRelations pour relier les descriptions et les entités commerciales sur le Web afin de décrire plus précisément la cible d'une discussion.

Pour augmenter la précision nous réfléchissons sur l'extension de notre mécanisme central d'opinion mining avec des techniques de traitement du langage naturel et / ou l'utilisation des réseaux de neurones pour catégoriser les sujets automatiquement. Dans le cadre de la classification de sentiment nous allons utiliser une approche similaire pour la création du classement d'opinion basée sur la polarité de contenu. Actuellement nous ne sommes pas plonger dans l'interprétation de sentiment d'opinions.

Conclusion

Dans ce mémoire, nous avons présenté une étude et une conception d'un système d'analyse sémantique d'opinion. On a commencé par présenter les technologies de base et les outils utilisés dans l'Opinion Mining, ensuite une étude détaillée de certains travaux existants pour nous donner des approches exemplaires et des idées ,le web sémantique et ses apports ainsi que des outils de réalisation des applications web sémantique , enfin nous avons proposé un système d'analyse sémantique d'opinion .

Lors de nos études et recherches nous avons remarqué l'existence d'un grand intérêt aux étapes de réalisation d'une ontologie, et de différentes technologies. Au niveau conceptuel, nous pourrions également enrichir notre modèle ontologique.

Nous sommes globalement satisfaits du travail réalisé, et nous sommes convaincus que les traitements sémantiques faciliteront grandement les applications de demain. En particulier, nous pensons qu'elles contribueront fortement à transformer la recherche d'informations sur Internet.

Des questions sur l'analyse des sentiments sont ouvertes. Concevoir une ontologie de sentiment et les méthodes pour permettre l'interrogation constituent des défis pour l'avenir de cet axe de recherche.

Bibliographie

1. **Bing., Liu.** *Web Data Mining. Springer, 2007.*

2. **Torsten, Suel et Vladislav, Shkapenyuk.** *Design and implementation of a high-performance distributed web crawler, 2001.*

3. **Fayyad, Piatetsky-Shapiro et Smyth.** *From data mining to knowledge discovery in databases. AI Magazine, 17(3), pp. 37-54. 1996.*

4. **Tanasa.** *Web usage mining : Contributions to intersites logs preprocessing and sequential pattern extraction with low support.*

5. **Cooley.** *Web usage mining : Discovery and application of interesting patterns from web data.*

6. **Liu, Bing et Hu, Minqing.** *Mining and summarizing customer reviews. KDD'04, 2004.*

7. **Philip, S. Yu Ding Xiaowen et Liu, Bing.** *A holistic lexicon-based approach to opinion mining. WSDM'08, 2008.*

8. **Chunping, Li et Lili, Zhao.** *Ontology based opinion mining for movie reviews. KSEM 2009.*

9. **Rakesh, Agrawal et Ramakrishnan, Srikant.** *Fast algorithms for mining association rules. 1994.*

10. **David, M. Pennock, Kushal, Dave et Steve, Lawrence.** *Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. WWW'03, 2003.*

11. **Simon, Corston-Oliver, Eric, Ringger et Michael, Gamon.** *An-thony Aue. Pulse: Mining customer opinions from free text. IDA, 2005.*

12. **Oren, Etzioni et Ana-Maria, Popescu.** *Extracting product features and opinions from reviews. EMNLP-05, 2005.*

13. **Tim, Berners-Lee, James, Hendler et Ora, Lassila.** *The semantic web (Berners-Lee et. al 2001). May 2001.*

14. **Handschuh, S.** *Creating Ontology-based Metadata by Annotation for the Semantic Web.PhD thesis, University of Karlsruhe, February 2005.*

15. **van, Harmelen.** *Semantic web research anno 2006: main streams, popular fallacies,current status and future challenges. In M. Klusch, M. Rovatsos, and T. Payne,editors, International Workshop on Cooperative Information Agents (CIA), number 4149 in Lecture No.*

16. **Fensel, D.** *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, Berlin, 2nd edition, 2004.*

17. **Gruber, T. R.** *A Translation Approach to Portable Ontology Speci_cations. Knowl-edge Acquisition, 5(2):199{220, 1993. URL http://tomgruber.org/writing/ontolingua-kaj-1993.pdf.].*

18. —. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies, 43(5-6):907{928, 1995.URL http://tomgruber.org/writing/onto-design.pdf.*

19. **Guarino, N.** *Formal Ontology and Information Systems. In Formal Ontology and Information Systems (FOIS'98), pages 3{15, Trento, Italy, June 1998. IOS Press.URL http://www.loa-cnr.it/Papers/FOIS98.pdf.*

20. **Studer, R, Benjamins, R et Fensel, D.** *Knowledge engineering: Principles and methods. IEEE Transactions on Data and Knowledge Engineering, 25(1):161{197, March 1998. URL http://www.das.ufsc.br/~gb/pg-ia/KnowledgeEngineering-Principles And Methods.pdf.*

21. **Sure, Y et Studer, R.** *Semantic Web technologies for digital libraries. Library Management, 26(4/5):190{195, April 2005. URL http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/2005_sw_for_dl.pdf.*

22. **Corcho, O et Gomez-Perez, A.** *A roadmap to ontology speci_cation languages.In 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000), pages 80{96, Juan-les-Pins, France, October 2000. Springer. URL http://www.cs.man.ac.*

23. **Klyne, G et Carroll, J. J.** *Resource Description Framework (RDF): Concepts and abstract syntax. Recommendation, W3C, February 2004. URL http://www.w3.org/TR/2004/REC-rdf-concepts-20040210.*

24. **Beckett., D.** *RDF/XML syntax speci_cation. Recommendation, W3C, February 2004.URL http://www.w3.org/TR/rdf-syntax-grammar/.],.*

25. **Berners-Lee, Teem.** *Notation 3 specification. Technical report, W3C, March 2006b. URL http://www.w3.org/DesignIssues/Notation3.html.].*

26. **Grant, J et Beckett, D.** *RDF test cases. Recommendation, W3C, February 2004.*

27. **Oren, E.** *Algorithms and Components for Application Development on the Semantic Web. PhD thesis, National University of Ireland, Galway, November 2007. URLhttp://www.eyaloren.org/pubs/phd-thesis.pdf.*

28. **Brickley, D et Guha., R.V.** *RDF vocabulary description language 1.0: RDF Schema.Technical report, W3C, February 2004. URL http://www.w3.org/TR/rdf-schema/.].*

29. *http://purl.org/dc/terms/] .*

30. *http://xmlns.com/foaf/0.1/]. .*

31. **Manola, F et Miller, E.** *RDF primer. Recommendation, W3C, February 2004. URL http://www.w3.org/TR/rdf-primer/.].*

32. *W3C Consortium (2004), OWL Specification Development, http://www.w3.org/2004/OWL/#specs , (2004).*

33. *The Protégé Ontology Editor and Knowledge Acquisition System . http://protege.stanford.edu/ .*

34. *Jena – A Semantic Web Framework for Java .http://jena.sourceforge.net/] .*

35. *http://gate.ac.uk.*

36. **Borislav, Popov, et al.** *KIM – Semantic Annotation Platform.*

37. **Masolo, C, et al.** *The WonderWeb Library of Foundational Ontologies and the DOLCE ontology,WonderWeb Deliverable D17. Preliminary Report (ver. 2.0, 15-08-200.*

38. **Oltramari, A, et al.** *Restructuring WordNet's Top-Level: The OntoClean approach In Proc. of LREC 2002 workshop "Ontologies and Lexical Knowledge bases" OntoLex 2002. 27 May, Las Palmas, Spain.*

39. **Kim, S et Hovy., E.** *Automatic identification of pro and con reasons in online reviews. In Proceedings of the COLING/ACL on Main conference poster sessions, pages 483–490, Morristown, NJ, USA, 2006. Association for Computational Linguistics.*

40. **Kobayashi, N, Inui, K et Matsumoto, Y.** *Opinion Mining from Web Documents:Extraction and Structurization. Informational and Media Technologies 2(1), 12(1):326–337, 2007.*

41. **Ghose, A, Ipeirotis, P et Sundararajan, A.** *Opinion Mining using Econometrics: A Case Study on Reputation Systems. In Proceedings of the Association for Computational Linguistics (ACL), 2007.],.*

42. **Gamon, M et Aue, A.** *Automatic identification of sentiment vocabulary: Exploiting low association with known sentiment terms. In Proceedings of the ACL-05 Workshop on Feature Engineering for Machine Learning in Natural Language Processing, 2005.*

43. **Fernandez, , S, Giasson, , F et Idehen., K.** *SIOC Ontology: Applications and Implementation Status. http://www.sioc-project.org/applications#creating-api,2007.*

44. **Mika., Peter.** *Microsearch: An Interface for Semantic Search. In Proc. of the Workshop on Semantic Search (SemSearch 2008) at the 5th European Semantic Web Conference (ESWC 2008) , Tenerife, Spain, volume 334 of CEUR Workshop Proceedings. CEUR-WS.org, 2008.*

45. *Semantic Web Deployment Working Group. SKOS Simple Knowledge Organization System Reference. W3C Working Draft, Semantic Web Deployment Working .*

MEMOIRE

En vue de l'obtention du

MASTER

Produits de l'Information Spécialisée et Médiation Electronique

Par

Soufiene KATET

ANALYSE SEMANTIQUE D'OPINION

Soutenu le 15 Septembre 2011, devant le jury composé de :

M.	*Rapporteur*
M.	*Examinateur*
M.	*Encadrant*