MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS MOSTAGANEM

Faculté des Sciences Exactes & de l'Informatique
Département de Mathématiques et d'Informatique
Filière Informatique

MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : Ingénierie des Systèmes d'Information

Modèles d'apprentissage automatique pour l'analyse des sentiments

Présenté par :
BEKADDOURI Nadia
MOHAMMED SEGHIR Nadia

Encadré par:
M. Brahmi Abderrezak

Année Universitaire 2012/ 2013

Dédicace

A mes très chers parents

Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi.

Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir.

Que ce modeste travail, soit l'exaucement de vos veux tant formulés et de vos prières quotidiennes.

A mes très chères sœurs et frère

Vous occupez une place particulière dans mon cœur. Je vous dédie ce travail en vous souhaitant un avenir radieux, plein de bonheur et de succès.

A mes très chers amis

En souvenir de nos éclats de rire et des bons moments. En souvenir de tout ce qu'on a vécu ensemble. J'espère de tout mon cœur que notre amitié durera Éternellement

MOHAMMED SEGHIR Nadia

Dédicaces

A mes très chers parents

Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi.

Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir.

Que ce modeste travail, soit l'exaucement de vos veux tant formulés et de vos prières quotidiennes (Baghdada, Soltana).

A mes très chères sœurs et frère

Vous occupez une place particulière dans mon cœur. Je vous dédie ce travail en vous souhaitant un avenir radieux, plein de bonheur et de succès (Wahiba, Amina, Kheira, Romaissa, Si-Mohammed) sans oublié mon mari MOUNIR et mes belle sœur que je l'adore (Amina, Hayat)

A mes très chers amis

En souvenir de nos éclats de rire et des bons moments. En souvenir de tout ce qu'on a vécu ensemble. J'espère de tout mon cœur que notre amitié durera Éternellement (Nadia, Sara)

BEKADDOURI Nadia

Remerciements

Nous tenons à exprimer nos vifs remercîments à :

Monsieur BRAHMI, pour son soutien, sa gentillesse, et sans laquelle nos projets, qui on espère seront les projets de toute une vie, auraient été incroyablement plus difficiles à mettre en œuvre.

Nous tenons également à remercier toute l'équipe du département de l'informatique.

Enfin, un grand merci à nos famille, avec une mention particulière à nos parents, qui nous ont soutenus et encouragés durant ce parcours.

Sommaire

Liste des figures

Liste des Tableaux

Introduction générale

Avec le début du troisième millénaire, le contenu du Web généré par les utilisateurs connaît une explosion extraordinaire. Les blogs individuels, les forums de discussion mais aussi les réseaux sociaux, d'ordre public ou professionnel, gagnent de plus en plus d'intérêts dans la vie des individus et des groupes. Ceci a conduit à de nouvelles opportunités et des défis importants pour les entreprises, qui sont plus préoccupés par la surveillance de la discussion autour de leurs produits afin de les améliorer ou de les commercialiser plus efficacement. Par ailleurs, des institutions gouvernementales portent plus d'attention au contenu des communautés sur le Web afin de suivre les diverses tendances de leurs citoyens.

Un élément important d'une telle analyse est de caractériser le sentiment exprimé dans les blogs sur un sujet spécifique. Des méthodes à base d'apprentissage (supervisé et non supervisé) et de traitement automatique du langage naturel sont développées et intégrées dans des applications plus complexes pour détecter, classer et suivre les tendances de consommation ou les opinions sociopolitiques des internautes.

Ce projet vise à étudier et évaluer un modèle, à base d'apprentissage supervisé, combinant sentiment et thème pour l'analyse des opinions et les sentiments des utilisateurs sur différents sujets.

Le présent mémoire est organisé sur quatre chapitres : dans le premier, nous présentons les concepts de base de l'analyse des sentiments ou ce qu'on appelle classification des opinions, tout en donnant les définitions et les raisons précises pour faire l'analyse des sentiments. Nous y exposons les différentes techniques utilisés en ce domaine tel que la catégorisation des textes basée sur l'apprentissage automatique. Le deuxième chapitre dresse un état de l'art des différentes approches existantes (linguistique, statistique et hybride) pour la classification des opinions et leur domaine d'application. En troisième chapitre, nous décrivons notre application de la conception par UML à l'implémentation avec l'environnement NetBeans et la plateforme Dragon. Les résultats des différentes expérimentations, obtenus pour une variété de collections, sont dressés dans le dernier chapitre.

Chapitre 1 : Analyse des sentiments

1. Introduction

Le développement des forums, des blogs et de la vente en ligne pousse les utilisateurs à laisser de plus en plus d'informations en libre accès sur le web. Une partie de ces informations décrit des sentiments: elles permettent de développer des modèles d'analyse des sentiments et de faire des sondages dans divers domaines en récupérant simplement ces données textuelles.

Dans ce chapitre nous introduisons les concepts de base de l'analyse des sentiments ou se qu'on appel classification des opinions en donnant une définition et des raisons précises pour faire l'analyse des sentiments en suite les différents techniques utilisés en ce domaine tel que la catégorisation des textes ,l'apprentissage automatique et en fin la recherche d'information

2. L'analyse des sentiments

2.1. Définition

Dans la littérature, l'analyse des sentiments est connue sur le nom « Opinion Mining » et elle est récemment devenu un domaine en plein développement en raison de ses nombreuses applications et ses 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.

C'est le domaine qui s'occupe de traitement d'opinion, du sentiment, et de la subjectivité dans le texte et nous avons précisé que c'est un sous domaine de la catégorisation de texte.

2.2. A quoi sert l'analyse des sentiments

Connaitre l'opinion des autres personnes a toujours été un élément d'information important durant le processus de décision. Avant de prendre des décisions, les gens s'intéressent énormément aux avis des autres personnes dans déférents domaines. Ils consultent les avis des autres consommateurs avant d'effectuer un achat, grâce à l'Internet nous pouvons découvrir

les opinions et les sentiments de très grand nombre de personnes dont leurs opinions peuvent être très utiles pour nous avant de faire notre choix et d'avoir notre propre idée sur un sujet donné.

Les entreprises peuvent répondre aux besoins des consommateurs en effectuant de la surveillance et de l'analyse des opinions pour améliorer leur produit [Zabin&Jefferies (2008)], donc il est nécessaire d'avoir un système capable d'analyser automatiquement les comportements généraux liés à la consommation, afin de mieux comprendre comment les différents produits et les services sont perçus par les clients

Analyse des sentiments est capable de déterminer la tonalité, les comportements émotionnels et les tendances dans les documents (corpus) afin d'évaluer si le l'opinion exprimée sur un sujet est positive ou négative en utilisant les techniques d'apprentissage automatique et de la recherche d'information.

3. La Catégorisation de Textes [Dziczkowski, D. (2008)]

La Catégorisation de Textes consiste en l'attribution d'une valeur booléenne à chaque paire <dj, c_i >∈ D × C où D est un domaine des documents et C = c_1, ..., $c_{|C|}$ est un ensemble de catégories prédéfinies. Une valeur de T attribuée à la paire < d_j, c_i > indique une décision de déposer dj sous c_i, et une valeur de F indique une décision de ne pas déposer dj sous c_i. Plus formellement, la tâche consiste à approximer une fonction inconnue d'une cible $\overline{\phi} : D \times C \rightarrow \{T, F\}$

(Qui décrit la façon dont les documents doivent être classifiés) par le biais d'une fonction $\phi : D \times C \rightarrow \{T, F\}$ appelée le classificateur de telle sorte que φ et φ coïncident autant que possible.

La Catégorisation de Texte a été utilisée dans un certain nombre d'applications différentes. Les premières applications concernées étaient l'indexation automatique pour les systèmes de Recherche d'Information (IR) booléens. Une autre application des techniques de TC est le Fouille de Textes (Ang : Text Mining). Le Fouille de Textes est l'activité de classification d'un flux de documents expédiés de manière asynchrone par un producteur d'information à destination d'un consommateur d'information.

4. L'apprentissage automatique

Depuis le début des années 90, l'approche de Machine Learning (ML) pour le besoin de la catégorisation du texte (TC) a gagné en popularité et a fini par devenir l'approche dominante. Dans cette approche, un processus inductif (également appelé l'apprentissage) construit automatiquement un classificateur pour une catégorie c_i en observant les caractéristiques d'un ensemble de documents classés manuellement pour c_i ou ci par un expert du domaine. De ces caractéristiques le processus inductif tire les caractéristiques que doit avoir le nouveau document pour être classé dans la catégorie ci.

Dans l'approche de ML, les documents pré-classifiés sont alors les ressources clés.

L'approche de ML repose sur la disponibilité d'un corpus initial

$\Omega = d_1, ..., d_{|\Omega|} \subset D$ De documents pré-classifiés sous $C = c_1, ..., c_{|C|}.$

En d'autres termes, les valeurs de la fonction $\breve{\Phi} : D \times C \rightarrow T, F$ sont connues pour chaque paire $< d_j, c_i > \in \Omega \times C.$

Un document dj est un exemple positif de ci si : $\breve{\Phi}(d_j, c_i) = T$

Un document dj est un exemple négatif de ci si : $\breve{\Phi}(d_j, c_i) = F$

Dans les paramètres de recherche, une fois qu'un classificateur Φ a été construit il est souhaitable d'évaluer son efficacité. Dans ce cas, avant la construction du classificateur, le corpus initial est divisé en deux séries, pas nécessairement de taille égale : un ensemble d'apprentissage et un ensemble de test.

L'ensemble d'apprentissage est $EA = \{d1, ..., d_{|EA|}\}.$

Le classificateur Φ pour les catégories C = {c_1, ..., $c_{|C|}$ } est construit en observant les caractéristiques de ces documents. L'ensemble de test ; $ET = \{d_{|EA|+1}, ..., d_{|\Omega|}\}$ est utilisé pour tester l'efficacité des classificateurs. Chaque dj ∈ ET est donné au classificateur, et les décisions du classificateur Φ (dj, c_i) sont comparées avec les décisions

d'expert ˇ Φ (d_j, c_i). Une mesure d'efficacité de la classification est basée sur la fréquence des valeurs de Φ (d_j, c_i) correspondant aux valeurs de ˇ Φ (dj, c_i).

Les documents de ET ne peuvent pas participer d'une façon quelconque à la construction d'induction du classement. Si cette condition n'était satisfaite, les résultats expérimentaux obtenus seraient probablement trop bons, et l'évaluation n'aurait donc pas de caractère scientifique [Mitchell (1996)]. La validation est une phase indispensable à tout processus d'apprentissage. Elle consiste à vérifier que le modèle construit sur l'ensemble d'apprentissage permet de classer tout individu avec le minimum d'erreurs possible.

Nous citerons deux méthodes de validation généralement utilisées : validation par le test, validation croisée.

4.1. Validation par le test

Dans le cas de la validation par le test, les résultats de l'évaluation précédente seraient une estimation pessimiste de la performance réelle, la dernière classification ayant été formée sur plus de données que le classificateur évalué. L'ensemble d'apprentissage permet de générer le modèle, l'ensemble de test permet d'évaluer l'erreur réelle du modèle sur un ensemble indépendant évitant ainsi un biais d'apprentissage. S'il s'agit de tester plusieurs modèles et de les comparer, nous pouvons sélectionner le meilleur modèle selon ses performances sur l'ensemble de validation et ensuite évaluer l'erreur réelle sur l'ensemble de test

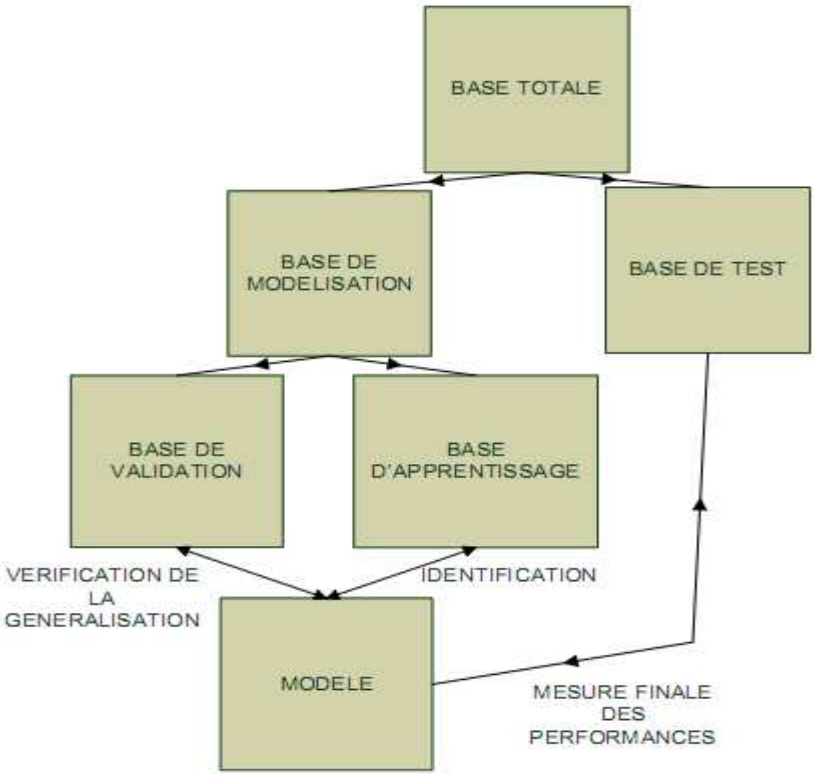

Figure 1.1 : Processus de validation par le test

4.2. Validation croisée

Une alternative est la validation croisée [Mitchell (1996)], dans laquelle k différents classificateurs Φ_1, ...,Φ_k sont construits par le partitionnement initial du corpus en k ensembles disjoints ET_1, ...,Et et la validation par test est ensuite appliquée de façon itérative sur les paires $EA_i = \Omega - ET_i, ET_i$. L'efficacité finale est obtenue par le calcul individuel de l'efficacité de Φ_1, ...Φ_k. La validation croisée ne construit pas de modèle utilisable, elle estime juste l'erreur réelle

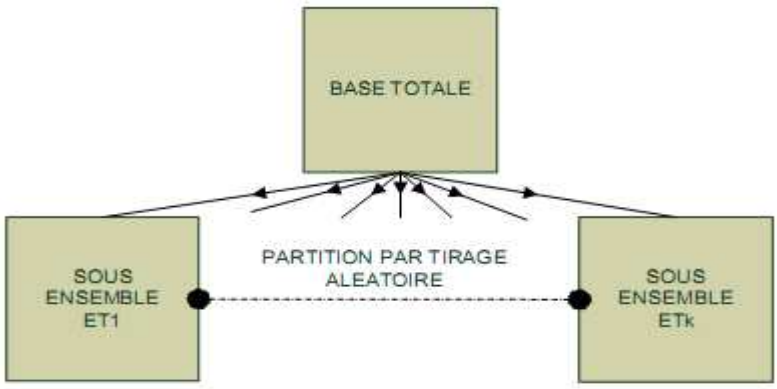

Figure 2.2 : Processus de validation croisé

5. La recherche d'information [El charif, R. (2006)]

Recherche d'Information (RI) ou recherche documentaire, propose de retrouver parmi une masse volumineuse de documents textuels, ceux qui correspondent au besoin informationnel d'un utilisateur généralement formulé par une requête en langage naturel.

5.1. Processus de recherche d'information

Le processus de Recherche d'Information a pour but de mettre en correspondance les représentations des informations contenues dans un fond documentaire d'une part avec celle des besoins de l'utilisateur d'autre part.

Ce processus est composé de trois fonctions principales :

- l'indexation des documents de la collection et des requêtes utilisateur.

- l'appariement des représentations requête-documents pour le calcul de la pertinence des documents en réponse un besoin utilisateur.

- la reformulation de requêtes qui permet de réécrire autrement la requête utilisateur puisqu'il est quasi-impossible aujourd'hui, de retrouver des informations pertinentes en utilisant la seule requête initiale de l'utilisateur, et ce à cause du volume croissant des bases documentaires. Pour résumer ces fonctions, nous pouvons représenter schématiquement certaines fonctionnalités d'un SRI par ce que l'on appelle communément le processus «enU », tel que dans la figure

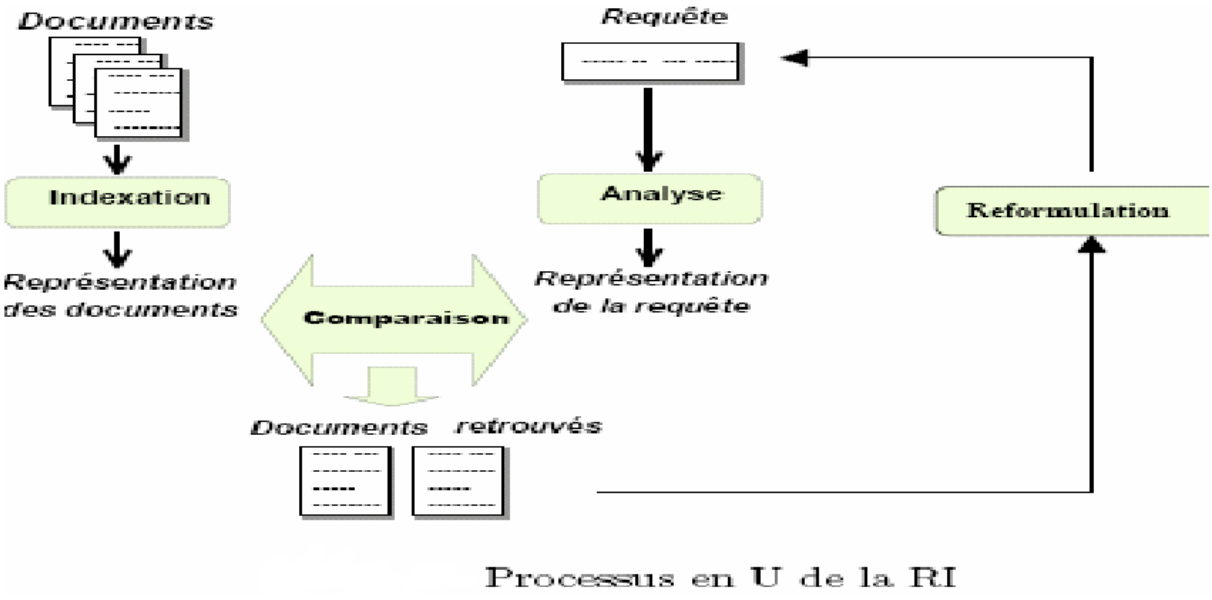

Figure 1.3 : Processus en U de la recherche d'information

5.1.1 Processus d'indexation

L'indexation est l'opération qui vise à construire une structure d'indexe qui permet de retrouver très rapidement les documents incluant des mots demandés. Cette étape consiste à analyser chaque document de la collection afin de créer un ensemble de mots-clés. Son objectif est de trouver les concepts les plus importants du document (ou de la requête), qui formeront le descripteur du document. Ainsi, en pratique, on cherche plutôt des représentants des concepts. Ces représentants peuvent être de formes différentes: des mots simples, ou de groupes de mots (mots composés).

Pour indexer un document ou une requête, différentes stratégies de pondération existent. La pondération consiste à donner aux termes de l'index un poids mesurant leur importance dans les documents qui les contiennent.

5.1.2 Pertinence et appariement document-requête

La pertinence concerne d'une manière générale la sélection des documents susceptibles d'être pertinents à une requête donnée. Elle est basée sur une fonction d'appariement (matching) qui effectue une comparaison entre les représentants des documents et des requêtes construits lors de la phase d'indexation. La comparaison revient à calculer un score représentant la pertinence du document vis-à-vis de la requête. Cette valeur est calculée à partir d'une fonction ou d'une probabilité de similarité notée RSV (Q, D) (Retrieval Status Value), où Q est une requête et D un document.et elle tient compte du poids des termes dans les documents, déterminé en fonction d'analyses statistiques et probabilistes.

5.1.3 Reformulation des requêtes

L'utilisateur exprime son besoin en information sous forme d'une requête afin de trouver des résultats qui l'intéressent. Cependant, le SRI lui rend parfois des résultats qui ne lui conviennent pas. L'augmentation continue du volume des bases documentaires rend quasi-impossible le fait de retrouver des informations pertinentes en utilisant la requête initiale de l'utilisateur. Pour cela, une étape de reformulation de la requête est souvent utilisée dans l'espoir de retrouver plus de documents pertinents. Ce processus permet de générer une requête plus adéquate que celle initialement formulée par l'utilisateur.

5.2. Modèles de recherche d'information

Il existe trois modèles de recherche d'information, et ces modèles diffèrent principalement par la façon dont les informations disponibles sont représentées, et par la façon d'interroger la base documentaire.

5.2.1 Modèle booléen

Le plus simple des modèles de RI, le premier imposé dans le monde de la RI, reconnu pour sa force pour faire une recherche très restrictive et obtenir, pour un utilisateur expérimenté, une information exacte et spécifique, il se base sur la théorie des ensembles dont le document est représenté par un ensemble des termes

Il considère que les termes de l'index sont présents ou absents d'un document, en conséquence, les poids des termes dans l'index sont binaires c.à.d. Wij= {0,1}.

Dans ce modèle, un document d est représenté comme une conjonction logique des termes non pondérés. Exemple : $d= t_1 \wedge t_2 \wedge t_3 \ldots\ldots\ldots t_n$.

Une requête q est composée de termes liés par 3 connecteurs logiques ET, OU, NON : Exemple : $q= (t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4)$ ou q : la requête, t_i : terme d'indexation.

5.2.2 Modèles vectoriels

Ce modèle représente les requêtes et les documents sous forme de vecteurs en fonction des termes d'indexation qui les composent dans un espace vectoriel spécifique. L'espace est de dimension N (N étant le nombre de termes d'indexation de la collection de documents). Cet espace vectoriel est définit par l'ensemble de termes que le système a rencontré durant l'indexation, parmi les modèles vectoriels on cite le modèle LSI.

A. Modèle LSI

Le premier modèle latent pour la RI se nomme l'indexage sémantique latent (Latent Semantic Indexing – LSI), aussi connu sous le nom de Latent Semantic Analysis. Celui-ci utilise des techniques d'analyse en composantes principales ou, de manière équivalente, de décomposition en valeurs singulières. En utilisant des principes géométriques, on effectue une approximation du vecteur des mots d'un document par un ensemble réduits de vecteurs. Ces vecteurs représentent les tendances principales des documents de la collection.

L'objectif fondamental du modèle LSI [Dumais 1995] est d'aboutir à une représentation conceptuelle des documents.

5.2.3 Modèles probabilistes

Le modèle probabiliste utilise un modèle mathématique fondé sur la théorie de la probabilité [Robertson et al. 1976] [Salton, 1983] [Kuhn, 1960] [Robertson, 1977].

Les modèles probabilistes représentent un document comme « sac de mots », plus précisément comme des vecteurs d'occurrences indexés par les mots, ils permettent de construire, de manière non supervisé, des partitionnements non déterministes d'un ensemble de documents, qui associent à chaque un vecteur donnant sa probabilités dans chacun des thèmes identifies par la méthode.

A. Le modèle PLSI

Plus tard, Hoffman propose le LSI probabiliste – pLSI, aussi connu sous le nomde modèle d'aspect. Celui-ci vise à corriger l'approche heuristique du LSI en représentant les documents par une mixture d'un ensemble de sujets. Chaque sujet est une distribution de mots reliés et ceux-ci sont appris par un algorithme EM(Expectation et Maximisation). C'est une approche à vraisemblance maximale, et ainsi souffre de problèmes de sur-apprentissage. A ces fins, Hoffman propose l'utilisation d'un contrôle de la capacité par tempérance, évaluant les sujets sur un ensemble externe.

B. Le modèle LDA

Latent Dirichlet Allocation [Blei et al, 2003] est un puissant algorithme d'apprentissage pour, de façon automatique et conjointe, classer des mots dans des documents dans des mélanges de contextes. Il a été appliqué avec succès pour modéliser les changements dans les domaines scientifiques au cours du temps. Dirichlet vient de Johann Peter Gustav Lejeune Dirichlet, Mathématicien allemand étudiait en France en 19ème siècle, qui a fait des travaux dans le domaine de l'analyse complexe et les lois de probabilité.

L'Allocation Latente de Dirichlet est un modèle génératif probabiliste. Partant de l'hypothèse que l'ordre des documents dans la collection est celui des mots dans un texte sont indifférents, LDA définit des modèles de mélanges finis sur des ensembles de sujets sous-jacents pour générer la collection, chaque sujet étant modélisé comme un mélange infini sur des probabilités des sujets sous-jacentes. Il a été démontré que PLSI est un cas particulier de LDA [Girolami and Kaban, 2003].

Le LDA a un but essentiel de classement, il permet d'associer un contexte à un document à partir des mots contenus dans ce document, lesquels mots pris individuellement pourraient appartenir à des contextes différents.

5.3. Evaluation

Pour une requête appliquée sur un ensemble de documents, nous identifions d'une part, l'ensemble des documents initialement pertinents et d'une autre part, l'ensemble des documents trouvés par le système. Ainsi les deux mesures seront définies comme suit :

La précision est définie par le taux des documents pertinents dans ceux qui sont trouvés.

$$\text{Precision}\,(P) = \frac{\#\{\text{documents pertinents trouvés}\}}{\#\{\text{documents trouvés}\}}$$

Le rappel peut être défini par le taux des documents trouvés parmi ceux qui sont pertinents.

$$\text{Rappel}\,(R) = \frac{\#\{\text{documents pertinents trouvés}\}}{\#\{\text{documents pertinents}\}}$$

Ces notions peuvent être clarifiées par la définition de quatre quantités :

- TP (vrai positif) : les pertinents correctement trouvés par le système.
- TN (vrai négatif) : les non pertinents correctement exclus.
- FP (faux positif) : les non pertinents mais trouvés à tord.
- FN (faux négatif) : les non pertinents mais exclus à tord.

Ainsi, les deux mesures seront définies comme suit :

$$P = \frac{TP}{TP + FP} \qquad\qquad R = \frac{TP}{TP + FN}$$

Dans un problème de classification multi-classes, on besoin d'avoir des mesures moyennes pour l'ensemble des classes. Dans ce cas, il est pratique d'appliquer la micro et la macro mesure moyenne de précision et de rappel comme suit :

	Micro-moyenne	**Macro-moyenne**
Précision	$P_{micro} = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k} TP_i + FP_i}$	$P_{macro} = \frac{1}{k}\sum_{i=1}^{k} \frac{TP_i}{TP_i + FP_i}$
Rappel	$R_{micro} = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k} TP_i + FN_i}$	$R_{macro} = \frac{1}{k}\sum_{i=1}^{k} \frac{TP_i}{TP_i + FN_i}$

Tableau 0 : Calcul de précision et rappel

6. Conclusion

Dans ce chapitre nous avons réalisé une étude détaillé sur les différents concepts clés dans le domaine de l'analyse des sentiments qui nous a permet de bien comprendre le fonctionnement de notre application en se basent sur des méthodes de la classification qu'on va les entamer dans le chapitre suivant.

Chapitre 2 : Approches de classification des opinions

1. Introduction

L'analyse de texte en terme d'étude des sentiments, opinions ou points de vue n'est pas récente (Carbonell, 1979; Wilks et Bien, 1984). Cependant le domaine de la fouille d'opinion et de l'analyse des sentiments a pris une grande place dès le début des années 2000 avec l'arrivée du Web communautaire et la multiplication des forums sur la toile. Depuis ce jour, le domaine est devenu un enjeu majeur pour toute entreprise désireuse de mieux comprendre ce qui plait et déplait à ses clients ainsi que pour les clients qui souhaitent comparer les produits avant de les acquérir. Par exemple, Morinaga et al. (2002) expliquent comment ils vérifient les réputations de produits ciblés en analysant les critiques des clients. Ils recherchent tout d'abord les pages Web parlant du produit concerné et extraient les phrases qui expriment de l'opinion. Ils classent ensuite les phrases selon qu'elles expriment une opinion négative ou une opinion positive et en déduisent la popularité du produit. Dans le même genre, Turney(2002) classe les commentaires selon deux catégories : *Recommended*et *Not Recommended*. Wilson et al. (2004) ajoutent à la classification selon la polarité, la force de l'opinion exprimée.

Enormément de travaux ont été effectués sur le sujet, et trois grandes catégories de méthodes peuvent être mises en avant : l'approche linguistique, l'approche statistique et l'approche hybride.

2. Les approches de la classification

2.1. L'approche linguistique

La principale tâche dans cette approche est la conception de lexiques ou dictionnaires d'opinion. L'objectif de ces lexiques ou dictionnaires est de répertorier le plus de mots porteurs d'opinion possible. Ces mots permettent ensuite de classer les textes en deux (positif et négatif) ou trois catégories (positif, négatif et neutre). Liu et al. (2005) décrivent un système, Opinion Observer, qui permet de comparer des produits concurrents en utilisant les

commentaires écrits par les internautes. Ils ont une liste prédéfinie de termes désignant des caractéristiques de produits. Lorsqu'une de ces caractéristiques est présente dans un texte, le système extrait les adjectifs proches dans la phrase. Ces adjectifs sont ensuite comparés aux adjectifs présents dans leur dictionnaire d'opinion et ainsi, une polarité est attribuée à la caractéristique du produit.

Cette méthode nécessite donc la construction d'un dictionnaire d'opinion. Pour construire un tel dictionnaire, trois genres de techniques sont possibles :

- La méthode manuelle.
- La méthode basée sur les corpus.
- La méthode basée sur les dictionnaires.

➢ La ***méthode manuelle*** demande un effort important en termes de temps mais il faut savoir que toutes les autres méthodes nécessitent également de créer initialement, de façon manuelle, un ensemble de mots et expressions porteurs d'opinions. Cet ensemble de mots est appelé graine. Il est ensuite utilisé afin de trouver d'autres mots et expressions porteurs d'opinions.

Une solution afin d'agrémenter cet ensemble de mots est donc l'utilisation de corpus de textes. Turney (2002) propose la méthode suivante : afin de déterminer la polarité de mots ou expressions non classés, il compte le nombre de fois où ces mots ou expressions apparaissent dans le corpus à côté de mots ou expressions déjà classés. Un mot apparaissant plus souvent à côté de mots positifs sera donc classé dans la catégorie positif et inversement. Yu et Hatzivassiloglou(2003) proposent une méthode similaire, mise à part qu'ils utilisent la probabilité qu'un mot non classé soit proche d'un mot classé afin de mesurer la force de l'orientation du premier nommé. D'autres méthodes (Pereira et al., 1993; Lin, 1998) utilisent également cette hypothèse dans le but d'agrémenter les lexiques d'opinion : deux mots ou groupes de mots ayant un fort degré d'apparition commune possèdent une forte proximité sémantique.

➢ Une autre ***méthode basée sur le corpus*** permettant d'agrémenter le dictionnaire d'opinion consiste à utiliser les conjonctions de coordination présentes entre un mot déjà classé et un mot non classé (Hatzivassiloglou et McKeown, 1997; Kanayama et Nasukawa, 2006; Ding et Liu,

2007). Par exemple, si la conjonction *AND* sépare un mot classé positif dans le dictionnaire d'opinion et un mot non classé, alors le mot non classé sera considéré comme étant positif. À l'inverse, si la conjonction *BUT* sépare un mot classé positif et un mot non classé, alors le mot non classé sera considéré comme étant négatif. Les conjonctions utilisées sont les suivantes : *AND*, *OR*, *BUT*, *EITHER-OR*, et *NEITHER-NOR*.

➢ La ***méthode basée sur les dictionnaires*** consiste à utiliser des dictionnaires de synonymes et antonymes existants tels que Word Net (Miller et al. 1990). Afin de déterminer l'orientation sémantique de nouveaux mots, Hu et Liu (2004a) utilisent ces dictionnaires afin de prédire l'orientation sémantique des adjectifs. Dans Word Net, les mots sont organisés sous forme d'arbres (voir figure 2.1). Afin de déterminer la polarité d'un mot, ils traversent les arbres de synonymes et d'antonymes du mot et, s'ils trouvent un mot déjà classé parmi les synonymes, ils affectent la même polarité au mot étudié, ou bien la polarité opposée s'ils trouvent un mot déjà classé parmi les antonymes. S'ils ne croisent aucun mot déjà classé, ils réitèrent l'expérience en partant de tous les synonymes et antonymes, et ce jusqu'à rencontrer un mot d'orientation sémantique connue.

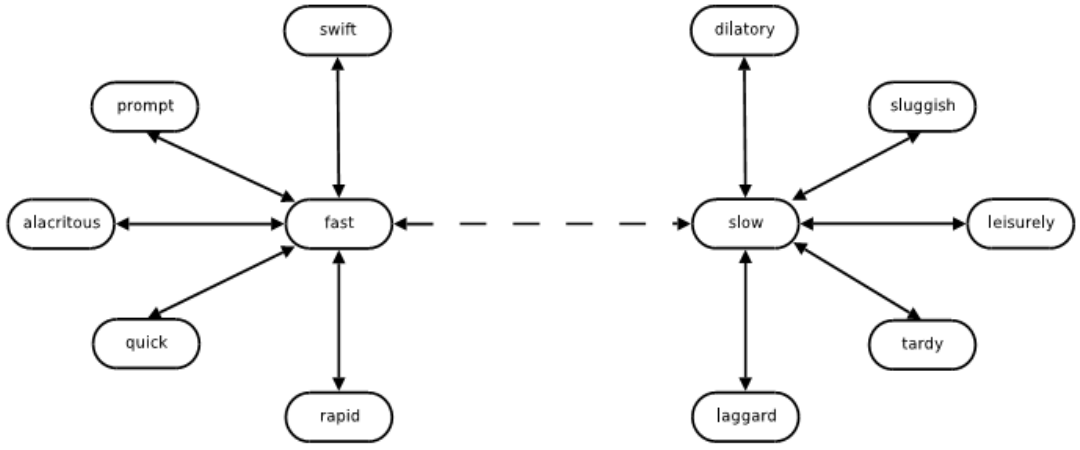

Figure 2.1 : Exemple d'arbre de synonymes et antonymes présents dans WordNet

Afin de mesurer plus précisément la force de l'opinion exprimée, un moyen utilisé est l'extraction des adverbes associés aux adjectifs. Pour ce faire, Benamara et al. (2007) proposent une classification des adverbes en cinq catégories : les adverbes d'affirmation, les adverbes de doute, les adverbes de faible intensité, les adverbes de forte intensité et les

adverbes de négation et minimiseurs. Un système d'attribution de points en fonction de la catégorie de l'adverbe permet de calculer la force exprimée par le couple adverbe-adjectif.
Toutes ces catégories d'adverbes ne sont pas toujours prises en compte car elles n'ont pas la même importance au niveau de la prédiction de note. Les négations paraissent logiquement être des termes importants à détecter, en plus des adjectifs et des verbes, car ils permettent d'inverser la polarité d'une phrase. Das et Chen (2001) proposent par exemple d'ajouter des mots dans le dictionnaire d'opinion comme "*like-NOT*" qui sont utilisés lors de la détection d'un couple like-négation. Les négations peuvent être *not, don't, didn't, never, ever...*
Le problème de la détection de la négation reste un problème très ouvert, les méthodes existantes n'étant pas réellement convaincantes. Ceci est aussi dû aux différentes façons d'utiliser la négation comme le sarcasme ou l'ironie.
Pour finir, il s'agit ensuite de déterminer la polarité d'une phrase à l'aide de ces dictionnaires.
La solution la plus simple consiste à compter le nombre de mots positifs et le nombre de mots négatifs présents. S'il y a une majorité de termes positifs, la phrase est déclarée positive.
À l'inverse, si les mots négatifs sont les plus nombreux, la phrase est déclarée négative. Les phrases possédant autant de mots négatifs que de mots positifs peuvent être déclarées neutres (Yu et Hatzivassiloglou, 2003), ou encore, la polarité de la phrase peut dépendre du dernier mot d'opinion parcouru (Hu et Liu, 2004a). On peut encore extraire plusieurs opinions dans une même phrase et les associer aux caractéristiques discutées (Hu et Liu, 2004b).

2.2. L'approche Statistique

Les méthodes statistiques les plus utilisées sont les méthodes à apprentissage supervisé. Ce type de méthode consiste à représenter chaque commentaire comme un ensemble de variables, puis à construire un modèle à partir d'exemples de textes dont on connaît déjà le label. Le modèle est ensuite utilisé pour attribuer sa classe à un nouveau commentaire non étiqueté.
Pang et al. (2002) montrent que des techniques d'apprentissage automatiques offrent de meilleurs résultats que les méthodes linguistiques décrites précédemment. Ils précisent toutefois que les dictionnaires d'opinion utilisés ne sont peut-être pas optimaux. Pour faire leurs comparaisons, ils ont basé leurs expérimentations sur trois méthodes de classification automatique : un classifieur naïf bayesien, un algorithme de Machines à Vecteurs Support et un classifieur basé sur le principe d'entropie maximale.
Mais en fait, peu de travaux sont basés uniquement sur des méthodes statistiques. Le plus souvent, des prétraitements linguistiques sont effectués sur les textes, soit pour réduire le nombre de variables, ou encore pour sélectionner uniquement les traits grammaticaux

susceptibles d'exprimer une opinion et ainsi éviter le bruit avec des mots inutiles pour ce type de classification. Ces approches constituent les approches dites hybrides.

2.3. L'approche hybride

Une première façon de faire est d'utiliser les outils linguistiques afin de préparer le corpus avant de classer les textes à l'aide d'outils d'apprentissage supervisé. Wilson et al. (2004) préparent les données à l'aide d'outils de Traitement Automatique des Langues afin de sélectionner un vocabulaire d'opinion. Ces mots pré-sélectionnés sont ensuite utilisés comme vecteurs de représentation des textes pour les outils d'apprentissage supervisé. Trois algorithmes d'apprentissage sont comparés : BoostTexter (Schapire et Singer, 2000), Ripper (Cohen, 1996)et SVMlight (Joachims, 1999a). Nigam et Hurst (2006) utilisent des techniques provenant du traitement Automatique des Langues afin de détecter dans les textes les mots et expressions porteurs d'opinion et ajoutent des marques dans le texte (traits grammaticaux et + ou – pour opinion positive et opinion négative). Ils utilisent ensuite l'apprentissage automatique pour classer les textes selon leur opinion générale.

Une autre façon de combiner les méthodes est d'utiliser les techniques d'apprentissage automatique dans le but de construire les dictionnaires d'opinion nécessaires à l'approche linguistique.

Hatzivassiloglou et McKeown (1997) présentent une méthode ayant pour objectif de définir l'orientation sémantique des adjectifs pour la construction du dictionnaire d'opinion.

Ils extraient tout d'abord tous les adjectifs du corpus à l'aide d'un analyseur syntaxique, puis utilisent un algorithme de clustering afin de classer les adjectifs selon leur polarité. Riloffet Wiebe (2003) combinent les deux approches afin de répertorier les expressions porteuses d'opinion qui, selon eux, sont plus riches que des mots pris individuellement. Turney et Littman (2003) utilisent une approche statistique pour classer un plus grand nombre de types de mots selon leur polarité : adjectifs, verbes, noms...

Une dernière façon d'utiliser conjointement les approches linguistiques et statistiques est de construire plusieurs types de classifieurs et de combiner leurs résultats, soit par des sytèmes de vote, soit par un algorithme d'apprentissage (Dziczkowski et Wegrzyn-Wolska, 2008).

3. Application des différentes approches

On présente ici plusieurs techniques ayant pour objectif de classer des textes selon l'opinion qu'ils expriment. La première méthode est une approche linguistique et la deuxième, une approche statistique. Nous abordons également une approche hybride consistant à préparer et

nettoyer le corpus à l'aide de méthodes linguistiques puis à classer les documents à l'aide d'outils d'apprentissage automatique. Nous allons présenter ici les travaux de Damien Poirier, Françoise Fessant, Cécile Bothorel Émilie Guimier de Neef, Marc Boullé sur les exemples de chaque approche, nous allons tout d'abord présenter le corpus sur lequel ils travaillent puis nous présenterons les différentes expérimentations réalisées.

3.1. Description du corpus

Le corpus d'expérimentations est extrait du site Flixster. Ce site est un espace communautaire américain destiné aux amateurs de cinéma qui permet entre autres choses aux utilisateurs de se créer un espace personnel et de partager leurs impressions sur des films et des acteurs, le plus souvent en anglais. Les commentaires faits sur les films sont associés à une note comprise entre 0,5 et 5 précisant l'impression générale portée sur le film en question.

La principale difficulté de ce corpus est la grande variété de commentaires. En effet, que ce soit au niveau du style d'écriture ou de leur taille, les commentaires peuvent présenter de grandes dissimilaires. Ceci rend la classification d'opinion plus difficile, parfois même pour un humain. De plus, une grande partie du corpus est composée de messages plus proches des messages de forums que des critiques faites par des journalistes ou des professionnels. Ils présentent des caractéristiques telles que des smileys (" :-) «) ,

Des accumulations de ponctuation (" ! ! ! "), du langage SMS (" ur "," gr8 ") ou encore des étirements des mots (" veryyyyyycooooool "). Le tableau 1 contient des exemples de commentaires associés aux notes.

Note	Commentaire
POS	Great movie !
NEG	this wasn't really scary at all i liked it but just wasn't scary...
POS	I loved it it was awsome !
NEG	I didn't like how they cursed in it.....and this is suppose to be for little kids....
NEG	Sad ending really gay
POS	sooo awsome ! ! (he's soo hot)
POS	This is my future husband lol (orlando bloom)
NEG	Will Smith punches an alien in the face, wtf ! ! ? ?
NEG	i think this is one of those movies you either love or hate, i hated it ! :o)

Tableau 1: Exemples de commentaires accompagnés de leur note

Le corpus extrait est composé de 60.000 commentaires. Comme ils l'avaient annoncé, la taille des commentaires est très variable, allant de 1 à 518 mots, avec une moyenne de 13 mots par texte. La figure 2 .2 donne une idée de la disparité rencontrée au niveau de la taille

des différents commentaires. Par exemple, on observe que 60% des commentaires contiennent moins de dix mots.

Pour faciliter l'interprétation des résultats de classification, ils ont décidé de réduire l'espace des notes (10 classes allant de 0,5 à 5) à deux classes : positive (notes supérieures ou égales à 3) et négative (notes inférieures à 3). Nous expliquons dans la section 3.3.4 commentce découpage a été déterminé. La moitié des commentaires composant le corpus appartient àla classe positive et l'autre moitié à la classe négative. Ils ont partagé le corpus en deux sous-ensembles de commentaires. Une partie pour les phases d'apprentissage (20 000 commentaires négatifs et 20 000 positifs) et une partie destinée aux tests (10 000 commentaires négatifs et 10 000 positifs). Tous les résultats qui suivent ont été obtenus à partir du même corpus.

Les seuls prétraitements appliqués sur le corpus, et qui sont valables pour tous les résultats qui vont suivre, sont la minusculisation de tous les caractères ainsi que la suppression de la ponctuation. Dans le cas de l'approche statistique, ils ont également supprimé tous les mots n'apparaissant qu'une seule fois dans le corpus dédié à l'apprentissage. Cela réduit de moitié l'espace de représentation des textes, et a un impact négligeable sur les résultats de la classification. Ils ont ensuite appliqué d'autres prétraitements linguistiques suivant l'expérimentation visée.

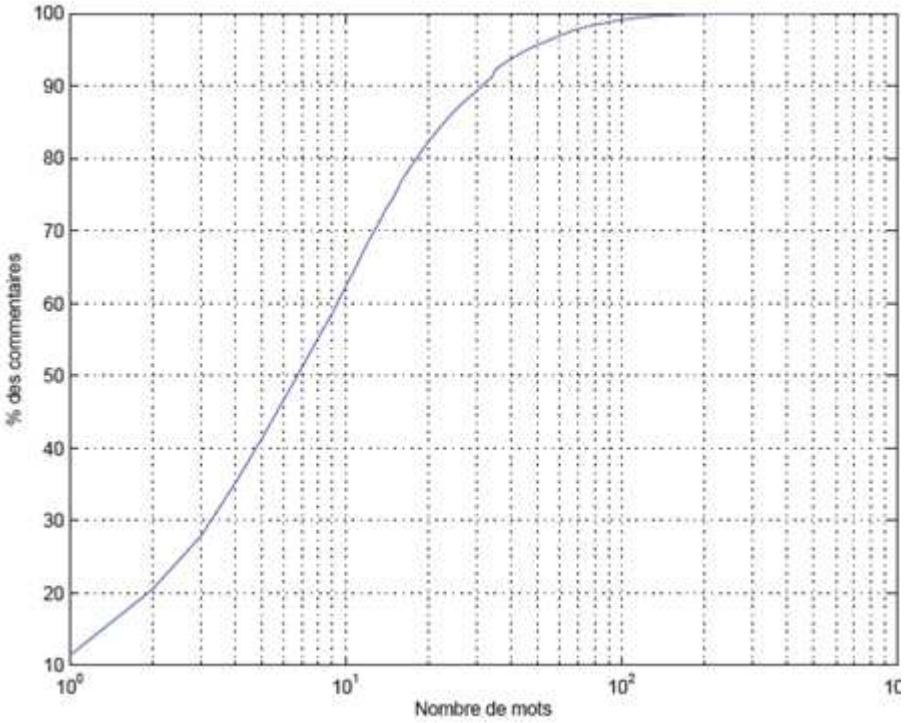

Figure 2.2 : Distribution cumulée du pourcentage de commentaires en fonction de leur taille

3.2. L'approche linguistique

Comme nous l'avons vu dans l'état de l'art, la première étape consiste à construire des dictionnaires contenant les mots porteurs d'opinion. Ils supposent donc que les opinions sont exprimées par certaines catégories de mots et que ces mots à eux seuls permettent de déterminer la polarité d'un texte. Nous allons tout d'abord expliquer comment ils ont construit leurs dictionnaires puis nous verrons comment ils déterminent la polarité des textes et les résultats obtenus avec cette méthode.

3.2.1 Construction des lexiques

Ils ont fait le choix de construire deux lexiques distincts. Le premier d'entre eux contient tous les mots porteurs d'opinion positive et le second tous les mots porteurs d'opinion négative. Pour trouver les mots exprimant une opinion et les classer, ils ont tout d'abord séparé le corpus d'apprentissage en plusieurs parties en fonction des notes attribuées à chaque commentaire. Ils ont donc obtenu dix sous-ensembles de commentaires notés respectivement de 0,5 à 5. Pour commencer nous avons appliqué, sur chacun des dix sous corpus, un analyseur syntaxique (Guimier de Neef et al. 2002) afin de lemmatiser et étiqueter chaque mot du texte. Ils ont basés sur l'hypothèse que les adjectifs et les verbes étaient les deux traits

grammaticaux les plus utilisés pour exprimer des opinions. Ils ont donc filtré les mots selon leur traits grammatical et leur fréquence dans chaque sous corpus, et conservé les adjectifs et les verbes ayant le plus d'occurrences. Les mots sélectionnés apparaissant souvent dans les ensembles de commentaires notés de 4 à 5 ont été intégrés dans le lexique de mots positifs et inversement avec les mots apparaissant dans les commentaires notés de 0,5 à 2. Les lexiques ont ensuite été nettoyés manuellement afin de supprimer les termes n'exprimant apriori aucune opinion, ou encore les termes ambigus. Par exemple, le mot "terrible" n'apparaît dans aucun des lexiques car il peut exprimer les deux types d'opinion.

Ils ont fait le choix de construire les dictionnaires d'opinion manuellement pour qu'ils ne contiennent que des mots vraiment spécifiques au corpus étudié. Ils pensent en effet que les lexiques d'opinion construits à l'aide des méthodes basées sur les dictionnaires (tels que WordNet), où l'on détermine la polarité des mots en fonction de leur synonymie, sont un peu trop aléatoires car beaucoup de mots peuvent avoir plusieurs sens selon le contexte. Ils ont aussi jugé que le corpus étudié n'est pas adapté aux constructions de lexiques d'opinion basées sur les corpus, les commentaires étant en règle générale très courts.

Au final, 183 mots a priori porteurs d'opinion ont été classés dans deux catégories. Le lexique de mots positifs contient 115 éléments et le lexique de mots négatifs en contient 68. Le tableau 2 présente des exemples de termes contenus dans les deux lexiques.

Mots postifs	good, great, funny, awesome, cool, brilliant, hilarious, favourite, well, hot, excellent, beautiful, cute, sweet ...
Mots négatifs	bad, stupid, fake, wrong, poor, ugly, silly, suck, atrocious, abominable, awful, lamentable, crappy, incompetent ...

Tableau 2 : Exemples de mots contenus dans les lexiques d'opinion

3.2.2 Classification d'opinion

Cette dernière étape consiste à compter les mots porteurs d'opinion répertoriés dans les deux lexiques afin de déterminer la polarité de chaque commentaire. Pour ce faire, ils ont appliqué, sur le corpus de test, les mêmes prétraitements que précédemment, à savoir la suppression de la ponctuation et la lemmatisation, et conservé uniquement les adjectifs et les verbes. Le fait de ne garder que ces deux catégories de mots permettent d'éviter quelques mauvaises interprétations de mots à double sens comme par exemple avec le terme "like" qui peut avoir plusieurs significations selon le contexte. Ils n'ont pas essayés à des analyses linguistiques

plus sophistiquées comme les analyses de structures grammaticales ou de dépendances au vu du style de langage utilisé dans le corpus.

La classification des commentaires se fait donc en comptant le nombre de mots d'opinion présents dans chaque commentaire. On calcul un score pour chaque commentaire en ajoutant1 lorsque l'on rencontre un mot positif, et en soustrayant 1 lorsque le terme rencontré est négatif. Les commentaires possédant donc une majorité de mots positifs (score positif) sont classés dans la catégorie Commentaires Positifs et inversement. Les commentaires possédant autant de mots positifs et négatifs sont ignorés. Il en est de même pour les commentaires quine contiennent aucun mot appartenant aux lexiques.

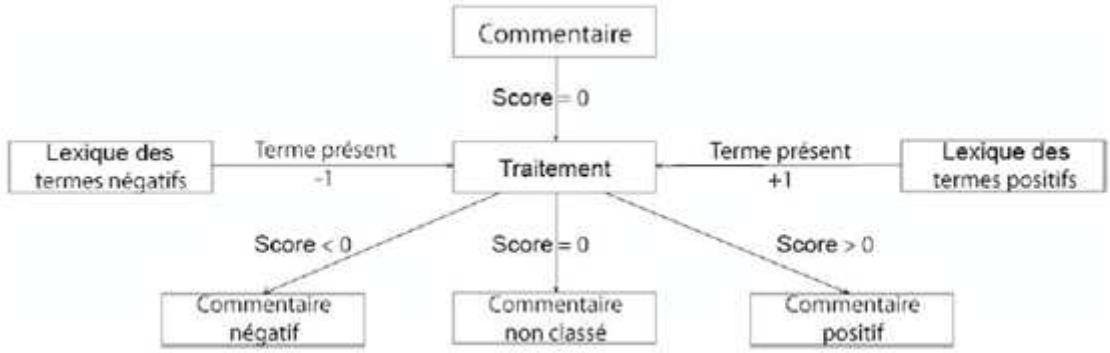

Figure 2.3 : Méthode de classification d'opinion

3.2.3 Résultats

Cette méthode il a permis de classer 74% des 20 000 commentaires présents dans le corpus de test. Afin de pouvoir comparer les résultats obtenus avec les résultats des approches statistiques, ils ont décidé que tous les commentaires que la méthode n'a pas réussi à classer sont des commentaires négatifs. En d'autres termes, tous les commentaires qui ne sont pas positifs sont négatifs.

Dans le but de comparer ces résultats avec les résultats des autres méthodes, ils calculent trois valeurs: la précision, le rappel et le Fscore. Les résultats de cette première expérimentation sont présentés dans la première colonne du tableau 3.

	Avec notre dictionnaire	Avec *General Inquirer*	Avec détection de la négation
Précision	72,8	65	70
Rappel	72,1	66,5	69,7
F_{score}	71,8	65,5	69,9

Tableau 3 : Résultats des expérimentations de l'approche linguistique

Ils peuvent voir dans la matrice de confusion (tableau 4) que la grande difficulté de la tâche se situe au niveau de la classification des commentaires négatifs. Ce problème pourrait être dû aux lexiques qu'ils ont créés. En effet, le lexique de mots positifs contient pratiquement le double d'éléments en comparaison au lexique de mots négatifs. Mais le problème n'est pas seulement la détection des commentaires négatifs mais également leur interprétation.

En effet, plus des 2/3 des commentaires classés négatifs sont mal classés.

	Commentaires Positifs	Commentaires Négatifs
Commentaires Positifs Prédits	8 089	3 682
Commentaires Négatifs Prédits	1 911	6 318

Tableau 4 : Matrice de confusion obtenue à partir de ses lexiques

Afin de vérifier la qualité de nos lexiques d'opinion, ils ont fait la même expérimentation en utilisant le lexique General Inquirer déjà construit par Stone et al. (1966) et Kelly et Stone (1975). Ce lexique contient 4 210 mots porteurs d'opinion (2 293 mots négatifs et 1 914mots positifs).

L'utilisation de ce nouvel ensemble de mots d'opinion permet de classer plus de commentaires (78% contre 74% précédemment) mais les résultats de la classification sont moins bons (voir deuxième colonne du tableau 3). Ces scores peuvent être expliqués par le fait que le dictionnaire General Inquirer a été construit pour analyser des corpus traitant de sujet généraux, et non juste des commentaires de films. Ils peuvent également observer, grâce à la matrice de confusion (tableau 5), que le problème concernant la classification des commentaires négatifs est présent cette fois encore, bien que le lexique contienne plus de mots négatifs que de mots positifs. L'explication la plus plausible à ce problème paraît donc être l'utilisation de la négation qui est plus présente dans les commentaires négatifs que dans les commentaires positifs.

	Commentaires Positifs	Commentaires Négatifs
Commentaires Positifs Prédits	7 027	3 743
Commentaires Négatifs Prédits	2 973	6 257

Tableau 5 : Matrice de confusion obtenue à partir du lexique General Inquirer

Afin de palier à ce problème, sans avoir recours aux techniques de Traitement Automatique des Langues qui pourraient être coûteuses au niveau de l'adaptation à ce type de corpus, ils

ont tenté de tenir compte des négations de manière plus simpliste. Ils ont créé un troisième lexique contenant les négations et minimiseurs. ILS ont répertorié six: not, ever, never, less, no, badly.

La détection, dans un commentaire, d'un terme appartenant à ce nouveau lexique inverse la polarité du prochain terme détecté appartenant aux deux autres lexiques (termes négatifs et termes positifs). Par exemple, la détection du mot "not" suivie du mot "good" soustraira 1 au score du commentaire alors qu'auparavant le score aurait été incrémenté. Ils peuvent voir dans la matrice de confusion (tableau 6) que les résultats de la classification des commentaires négatifs sont légèrement améliorés par rapport à ceux de la première expérimentation. Par contre ceux concernant la classification des commentaires positifs sont dégradés. Les résultats globaux (voir troisième colonne du tableau 3) sont également moins bons que ceux obtenus avec la classification qui ne tient pas compte de la négation. Précisons que, cette fois-ci, les commentaires classés représentent 80% du corpus de test, soit 6% de plus que la première expérimentation.

	Commentaires Positifs	Commentaires Négatifs
Commentaires Positifs Prédits	7 057	3 132
Commentaires Négatifs Prédits	2 943	6 868

Tableau 6: Matrice de confusion obtenue à partir de ses lexiques en prenant en compte les négations

3.2.4 Conclusion de l'approche linguistique

La prise en compte des négations comme elle a été faite ici n'est pas réellement efficace et les meilleurs résultats obtenus correspondent à ceux de la première expérimentation, avec l'utilisation de leur dictionnaire, en tenant compte uniquement des adjectifs et verbes. Les résultats pourraient peut-être être améliorés en considérant les négations comme des mots négatifs Benamara et al. (2007). Une analyse relationnelle des phrases (qui permet d'extraire les relations existant entre les mots) serait certainement la solution la plus efficace, mais il faudrait pour cela un outil adapté au style de langage utilisé dans le corpus étudié ici.

Ils peuvent également remarquer que les résultats obtenus avec le dictionnaire construit à partir du corpus traité sont significativement supérieurs à ceux obtenus à partir du dictionnaire General Inquirer. On peut donc en déduire que si une application vraiment ciblée est envisagée (par exemple une classification de commentaires de films), l'utilisation d'un dictionnaire d'opinion adapté est plus recommandée que l'utilisation d'un dictionnaire général.

3.3. L'approche statistique

Nous nous intéressons ici à deux techniques d'apprentissage supervisé : les Machines à Vecteur Support (SVM) et les Classifieurs Naïfs Bayesiens (NB), dont une méthode de classification Naïve Bayesienne avec sélection de variables (SNB). Nous allons tout d'abord présenter rapidement les trois méthodes utilisées. Puis nous expliquerons comment nous avons procédé pour le choix des classes à prédire.

3.3.1 Classification Naïve Bayesienne

Le classifieur Bayesien Naïf a démontré son efficacité sur de nombreuses applications réelles (Hand et Yu, 2001). Cette méthode de classification repose sur l'estimation de la probabilité d'occurrence d'évènements avec la règle de Bayes. Ce classifieur, qui suppose que les variables explicatives sont conditionnellement indépendantes, nécessite uniquement l'estimation des probabilités conditionnelles univariées. Les études menées dans la littérature (Liuet al, 2002) ont démontré l'intérêt des méthodes de discrétisation pour l'évaluation de ces probabilités conditionnelles.

3.3.2 Classification Naïve Bayesienne avec sélection de variables : KHIOPS

Khiops 2 est un outil de data mining permettant de construire automatiquement un modèle de classification performant, sur de très grandes volumétries (Boullé, 2005, 2006, 2007). Dansune première phase de préparation de données, les variables explicatives sont évaluées individuellement au moyen d'une méthode de discrétisation optimale dans le cas numérique et de groupement de valeurs optimal dans le cas catégoriel. Dans la phase de modélisation, un modèle de classification est construit, en moyennant efficacement un grand nombre de modèles basés sur des sélections de variables. L'outil Khiops a été évalué avec succès lors de plusieurs challenges internationaux de data mining, et est utilisé à France Télécom par une trentaine d'utilisateurs pour des problèmes de marketing (churns, scores d'appétence...), de texteManning,wemining, étude technico-économique, ergonomie, sociologie.

3.3.3 Machine à Vecteur Support : SVMlight

L'objectif des Machine à Vecteur Support est de trouver un hyperplan qui sépare les exemples positifs des exemples négatifs. SVMlight 3 est une implémentation de la Machine à Vecteur, Support de Vapnik (Vapnik, 1995). Les algorithmes d'optimisation utilisés sont décrits dans Joachims (2002) et Joachims (1999a). Ce code a déjà été utilisé pour un grand nombre de problèmes, notamment la classification de texte (Joachims, 1999b, 1998), ainsi que des tâches de reconnaissance d'image, de bioinformatique et des applications médicales.

3.3.4 Choix des classes de commentaires

Comme ils l'ont précisé plus tôt, les commentaires composant le corpus d'analyse sont à l'origine notés sur dix classes (de 0,5 à 5). Il aurait été possible, au moins pour les méthodes NB et SNB, de réaliser la classification sur plus de deux classes. Mais l'augmentation du nombre de classes de projection rend l'interprétation des résultats d'autant plus difficile.

En effet, imaginent une classification sur 5 classes de notes allant de 1 à 5, 1 désignant un commentaire très négatif et 5 un commentaire très positif. Le fait qu'un commentaire noté 1par son auteur soit noté 2 par l'outil d'apprentissage doit-il être considéré comme une erreur, ou doit-on fixer une tolérance, et si oui, à combien doit-on la fixer. Pour éviter ces problèmes d'interprétation et faciliter l'analyse des résultats, ils ont donc décidé de réduire le nombre de classes.

Ils ont tout d'abord tenté une projection sur deux classes : NEG pour négatif et POS pour positif. Ils sont partis de l'hypothèse que les commentaires notés 0,5 et 1 étaient tous négatifs, et que les commentaires notés 5 étaient tous positifs. Ils ont donc réalisé la phase d'apprentissage uniquement sur les commentaires notés 0,5, 1 et 5, puis réalisé la classification indépendamment sur chaque classe restante. Les résultats sont présentés dans le tableau 7.

	1,5	2	2,5	3	3,5	4	4,5
NEG	78,18	75,46	65,58	53,62	33,58	28,2	17,4
POS	21,82	24,54	34,42	46,38	66,42	71,8	82,6

Tableau 7 : Résultats de la projection sur deux classes

Ils peuvent observer que les résultats sont assez cohérents. Comme on pouvait s'y attendre, les plus mauvais résultats se situent au niveau des commentaires notés 2,5, 3 et 3,5.

Ils ont donc tenté d'introduire une troisième classe (NEUTRE) en considérant, pour l'apprentissage, que les commentaires notés 3 appartenaient tous à cette nouvelle classe. La phase d'apprentissage se fait donc avec une classe négative contenant les reviews notées 0,5 et 1, une classe neutre contenant les reviews notées 3 et une classe positive contenant les reviews notées 5, les résultats qui sont présentés dans le tableau 8, montrent que, pour pratiquement toutes les classes, plus de la moitié des commentaires tombent dans la classe NEUTRE.

	1,5	2	2,5	3	3,5	4	4,5
NEG	40,6	32,76	20,88	14,92	9,64	9,72	8,28
NEUTRE	51,56	59,08	67,02	65,98	55,36	44,06	30,14
POS	7,84	8,16	12,1	19,1	35	46,22	61,58

Tableau 8 : Résultats de la projection sur trois classes

Au vu de ces derniers résultats, ils ont donc décidé de partager le corpus en 2 classes. La classe positive regroupant les commentaires ayant une note comprise entre 3 et 5, et les commentaires négatifs contenant les commentaires notés de 0,5 à 2,5.

3.3.5 Résultats

Ici, ils ont décidé de n'avoir aucun a priori sur les données. Ils ont conservé les commentaires tels qu'ils ont été écrits par leurs auteurs avec seulement des prétraitements minimaux. Rappelons que les seuls traitements subis par le corpus sont la minusculisation des caractères, la suppression de la ponctuation ainsi que la suppression des mots n'apparaissant qu'une seule fois dans le corpus d'apprentissage. Ils n'ont appliqué sur le texte aucun traitement linguistique. Chaque commentaire est représenté sur un vecteur composé de 12 153variables. Comme on peut le voir dans le tableau 9 qui présente les résultats des trois expérimentations, les meilleurs scores sont obtenus avec SVMlight, suivis par KHIOPS et enfin le classifieur naïf bayesien.

	KHIOPS	Naïve Bayes	SVM^{light}
Accuracy	76,3	69,8	79,6
Précision	76,6	71	81,6
Rappel	76,2	69,8	76,5
F_{score}	76,4	70,4	79

Tableau 9 : Résultats des expérimentations sans prétraitement

3.3.6 Analyse des résultats obtenus avec la méthode SNB

Voici une analyse plus approfondie des résultats obtenus avec l'outil KHIOPS, appliqué sur le corpus représenté en sac de mots, lors de la phase d'apprentissage, sans autres prétraitements que la minusculisation des caractères et la suppression de la ponctuation. Le nombre de variables (mots différents) présentes dans le corpus d'apprentissage s'élève à 12.153. L'outil en a sélectionné 305 qui lui paraissent les plus informatives pour la classification d'opinion. L'étude de ces variables sélectionnées permet d'apprendre des informations sur le corpus.

La première constatation que l'on peut faire est que les 305 variables sélectionnées possèdent des degrés d'information très variables, et peu d'entre elles sont très informatives (voir figure 2.4). Les variables sont classés en fonction de leur levé. Le levé est directement relié à la probabilité a posteriori d'un modèle de discrétisation, avec une normalisation O-1. Il vaut 0 lorsque la variable n'est pas du tout informative, et 1 lorsque la variable a un niveau d'information maximal.

Cette liste de variables informatives, contient une majorité de mots que l'on peut catégoriser comme mots porteurs d'opinion (voir tableau 10), mais pas seulement. D'autres mots, dont la présence dans la liste est plus surprenante, sont aussi informatifs que les mots d'opinion, voire d'avantage. On peut par exemple voir dans le tableau 11 l'importance du mot "and". Le tableau se lit de la façon suivante. La première colonne donne la fréquence du mot "and" dans un commentaire, la deuxième colonne donne le nombre de commentaires contenant autant de fois le mot "and" qu'indiqué dans la première colonne. Enfin, les deux dernières colonnes nous présente comment sont réparties ces commentaires dans les classes positive et négative.

On peut donc observer que plus le mot "and» n'apparait dans un commentaire, plus la probabilité que ce commentaire soit positif est élevée. On peut donc penser que certains auteurs ont tendance à être plus prolixe lorsqu'ils ont apprécié un film que lorsqu'ils ne l'ont pas aimé.

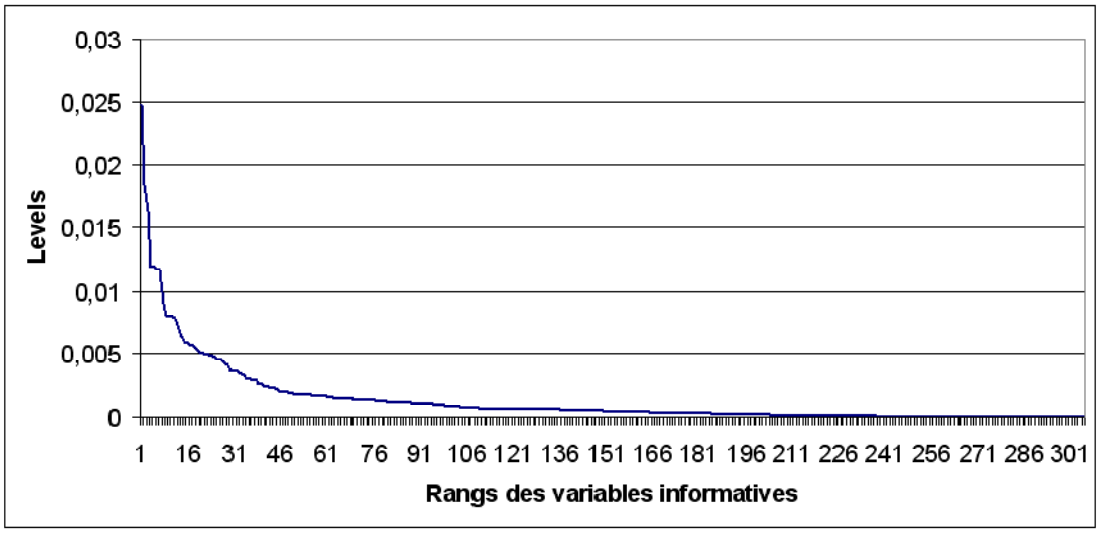

Figure 2.4 : Evolution du niveau d'information des variables sélectionnées par KHIOPS.

Variables informatives	love, great, best, loved, boring, ok, brilliant, awesome, worst good, stupid, funny, amazing, hilarious, crap, awsome, hate bad, excellent, fantastic, favorite, ...

Tableau 10 : Exemples de mots contenus dans la liste des variables informatives

Fréquence du Mot	Commentaires Concernés	Commentaires Négatifs	Commentaires Positifs
n = 0	33 725	52,54%	47,46%
n = 1	4 439	41,17%	59,83%
1 < n < 5	1 619	29,96%	70,04%
n > 4	217	5,99%	94,01%

Tableau 11 : Statistique concernant le mot « and »

Fréquence du Mot	Commentaires Concernés	Commentaires Négatifs	Commentaires Positifs
n = 0	30 849	53,54%	46,46%
n = 1	9 151	38,05%	61,95%

Tableau 12 : Statistique concernant le mot « movie »

3.3.7 Conclusion sur l'approche statistique

Les meilleurs scores ont été obtenus avec la méthode SVM. Malgré tout, la méthode SNB reste très intéressante à utiliser car elle permet d'obtenir des informations sur les variables les plus informatives, autrement dit celles qui permettent de classer les commentaires, que l'on n'aurait pas devinées instinctivement (importance du mot "and" par exemple).

4. Conclusion

Nous avons présenté dans ce chapitre l'essentiel des approches de classification d'opinion. La première approche consiste à construire un dictionnaire d'opinion manuellement avec l'aide de techniques simples de Traitement Automatique des Langues. Ce dictionnaire permet ensuite de classer les textes selon leur polarité, positive ou négative. Pour la seconde approche, ce sont des outils d'apprentissage automatique qui ont été utilisés dans le but, toujours, de classer les textes selon qu'ils expriment une opinion générale positive ou

négative. Plusieurs outils d'apprentissage supervisé ont été comparés : un classifieur Naïf Bayésien, un classifieur Naïf Bayésien avec sélection de variables et une Machine à Vecteur Support. Pour finir ils ont appliqué des tâches linguistiques sur le corpus afin de changer la représentation des textes et comparer les résultats de la classification effectuée avec les méthodes d'apprentissage. Les expérimentations effectuées sur des commentaires de films issus de blogs d'opinion ont montré que les méthodes statistiques étaient plus performantes que leur approche linguistique.

Chapitre 3 :
Conception et implémentation

1. 1. Introduction :

La partie conception dans un projet informatique a une très haute importance, elle permet d'avoir une idée de ce qu'on doit programmer, et déterminer les différentes fonctionalités de l'application, leurs conditions et l'ordonnancement de leurs déroulements. Dans ce chapitre on abordera la modélisation de notre application en utilisant le langage de modélisation UML.

2. 2. Architecture globale :

L'architecture adoptée pour notre application est illustré dans la figure 3, D'abord, il est nécessaire de préparer une collection de documents lisibles par machine. Deuxièmement, la transformation de la langue naturelle (NLP) est appliquée à chaque article de la collection. Les techniques NLP fréquemment utilisées comprennent l'atomisation (tokenisation), lemmatisation (lemmatisation) et stemming, Troisièmement, tous les mots extraits, des phrases ou des concepts ontologiques sont enregistrées souvent dans les matrices creuses pour une utilisation future. Quatrièmement, les tâches d'exploration des données de texte telles que la recherche, le regroupement ,la classification, sont exécutés sur des données indexées pour réaliser des modèles d'apprentissage automatique.

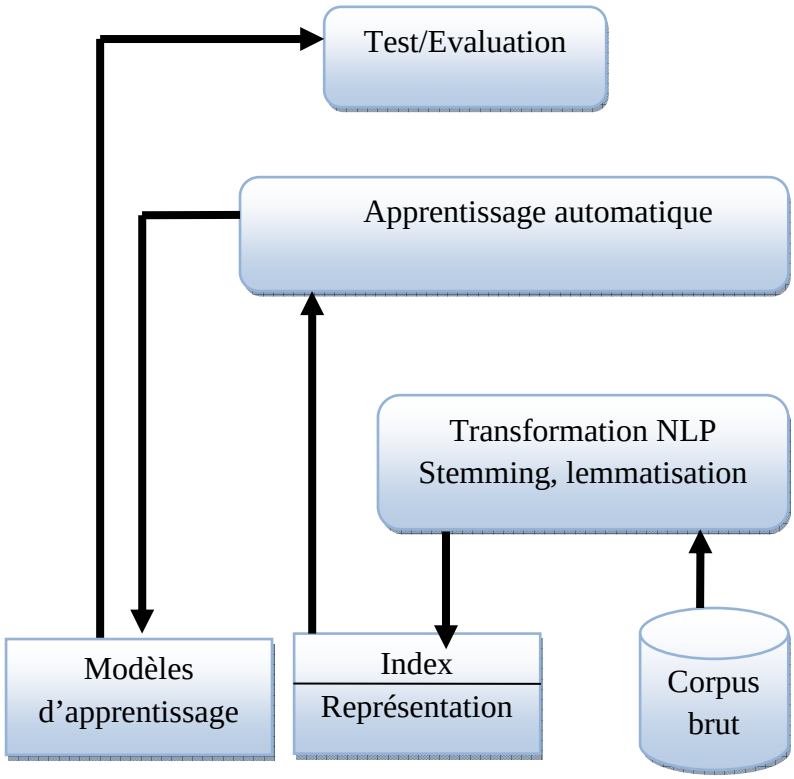

Figure 3.1 : Architecture globale

3. 3 Modélisation UML :

3.1. 3.1 Diagramme de classes :

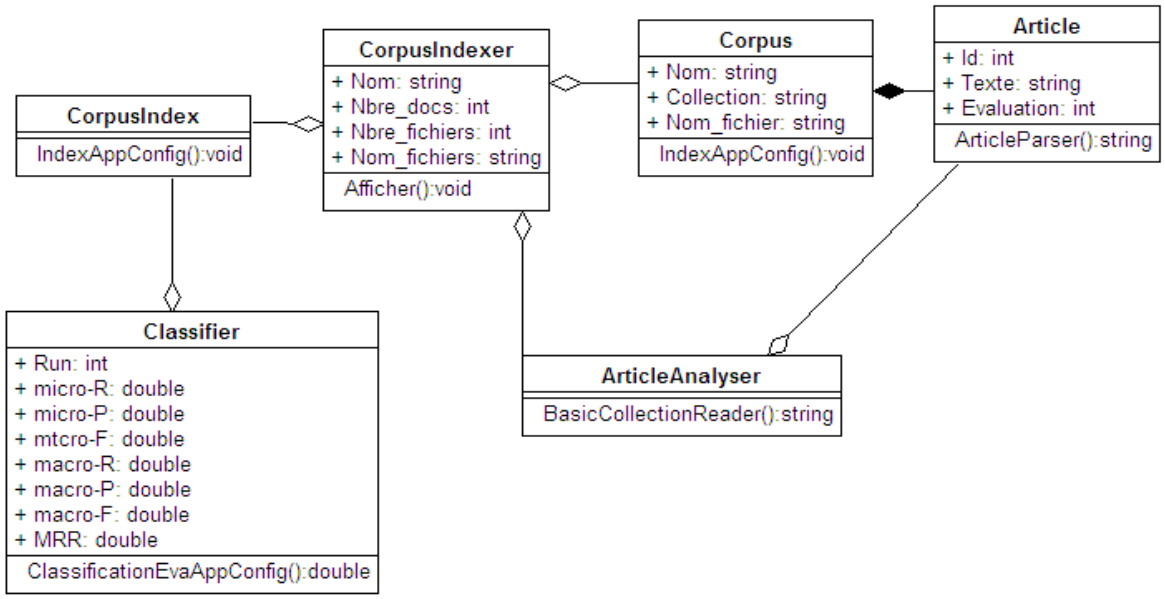

Figure 3.2 : Diagramme de classes.

3.2. 3.2 Diagramme de cas d'utilisation :

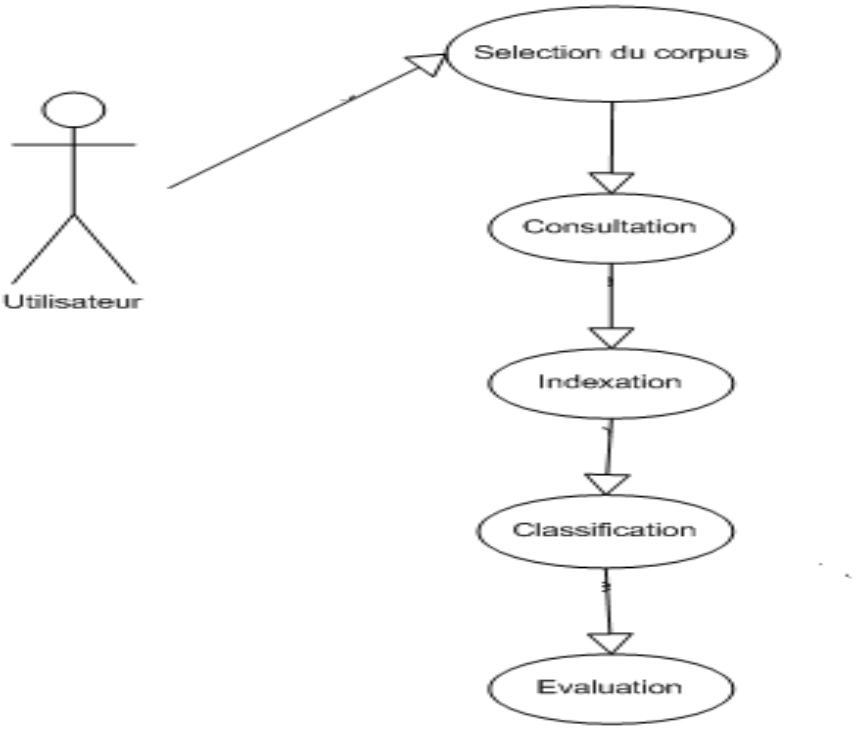

Figure 3.3 : Diagramme de cas d'utilisation.

3.3. 3.3 Diagramme d'activité :

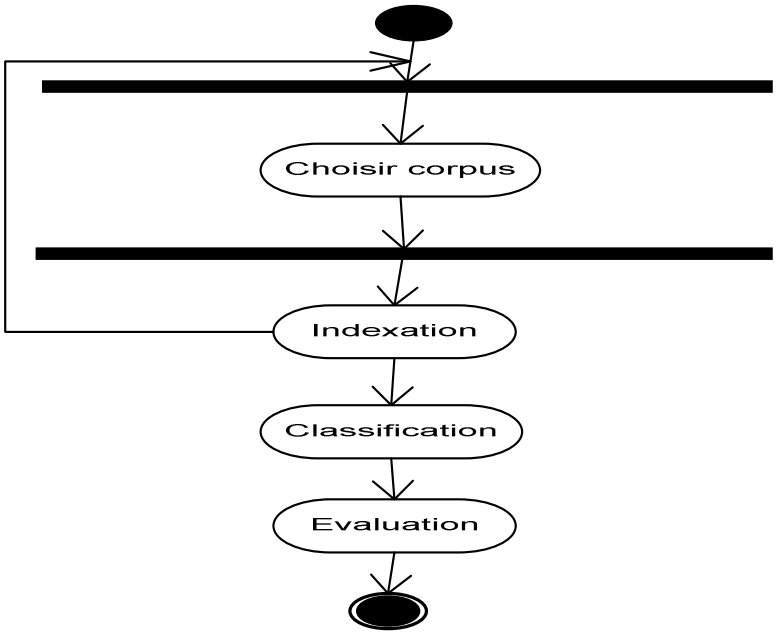

Figure 3.4 : Diagramme d'activité.

3.3.1 3.3.1 Diagramme d'activité pour le choix d'un corpus :

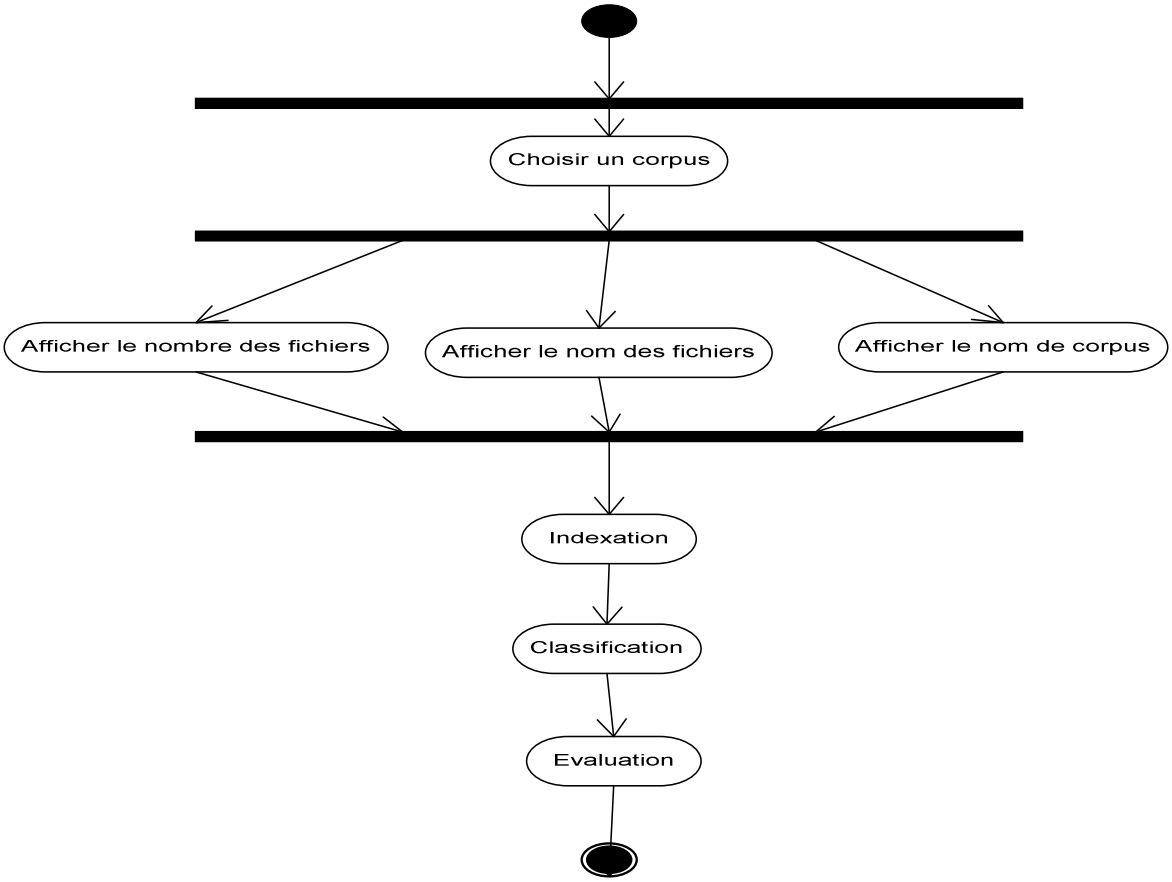

Figure 3.5 : Diagramme d'activité pour le choix d'un corpus.

3.3.2 3.3.2 Diagramme d'activité pour l'étape de l'indexation :

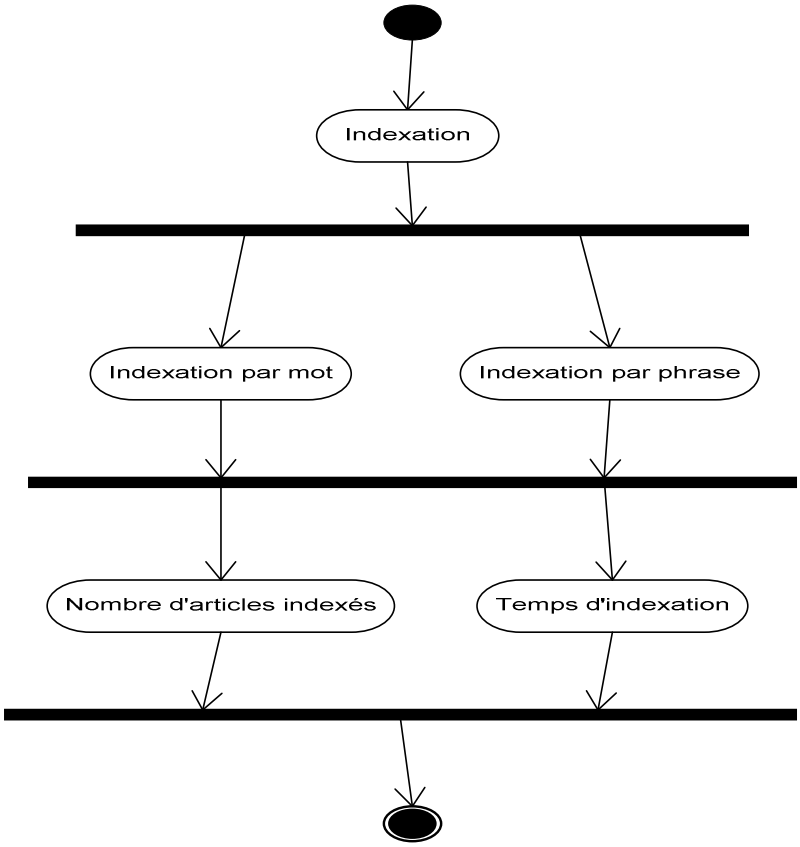

Figure 3.6 : Diagramme d'activité pour l'étape de l'indexation.

3.3.3 3.3.3 Diagramme d'activité pour l'étape de la classification :

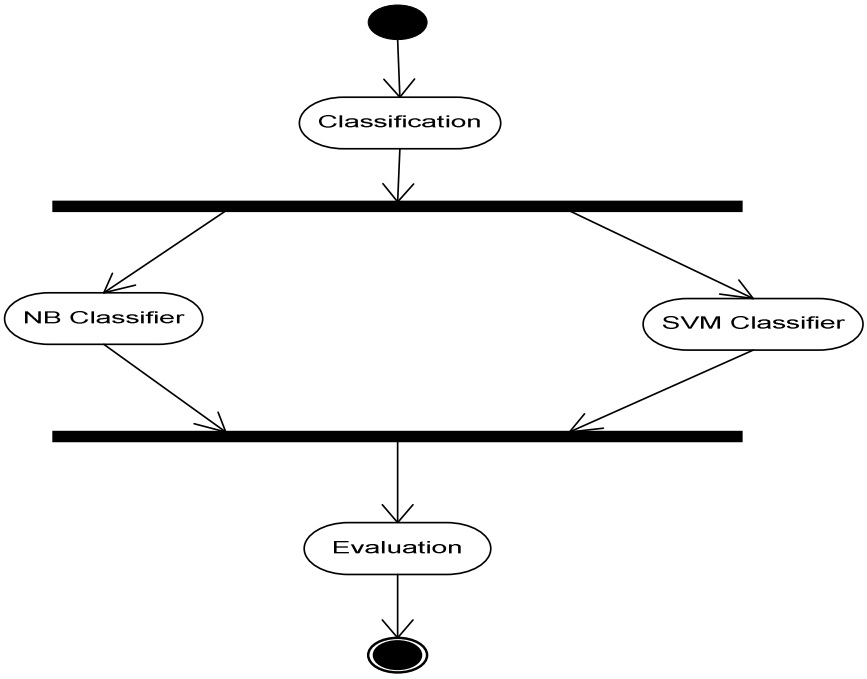

Figure 3.7 : Diagramme d'activité pour l'étape de la classification.

3.3.4 Diagramme d'activité pour l'étape de l'évaluation :

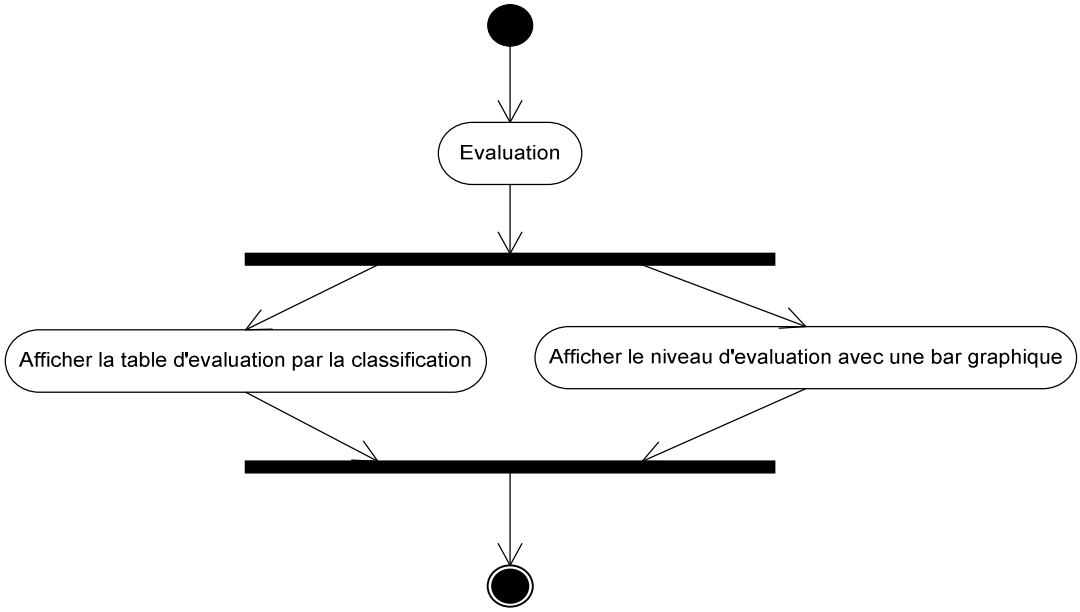

Figure 3.8 : Diagramme d'activité pour l'étape de l'évaluation.

3.3.4 3.3.4. Diagramme d'activité pour intégrer un corpus dans Dragon :

N'importe quel corpus contient une collection, un fichier XML pour la configuration et un fichier de document texte pour les sujets de la collection peut l'intégré dans dragon.

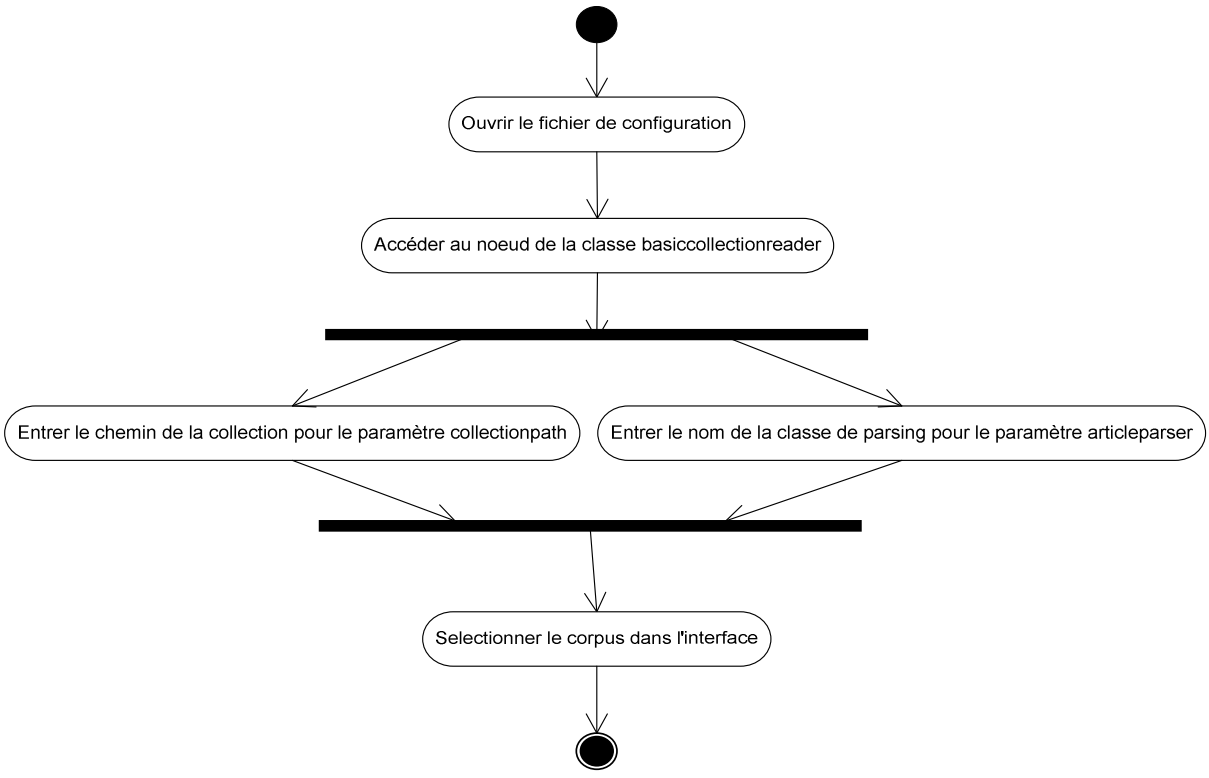

Figure 3.9: Diagramme d'activité pour intégrer un corpus dans Dragon.

4. Les outils de développement

4.1. Outils et ressources NLP

Le package NLP (Natural language processing) fournit les fonctionnalités pour l'extraction des concepts différents (des mots, des phrases à plusieurs mots individuels, noms propres) ou des relations entre les articles lors de l'indexation.

Afin de faciliter l'extraction, dragon intègre divers outils de NLP comme stemmers, une partie des tagueurs de la parole (speech taggers), les analyseurs de phrases (sentence parsers), extracteus de phrase (phrase extractors), et nommé reconnaissance de l'entité. Pour la commodité du traitement du langage naturel, la boîte à outils fournit quatre structures de base de données, Document, paragraphe, une phrase, et la Parole, pour analyser et symboliser le contenu d'un article. Les extracteurs et d'autres supportant les outils PNL utilisent ces quatre structures de base de données pour partager des données.

La boîte à outils définit trois types d'extracteurs de concept. La première est l'extracteur symbolique, qui extrait une séquence de mots individuels d'une phrase ou d'un document. Le second est l'extracteur de phrase, à savoir extraire plusieurs mots d'une phrase ou d'un document. L'extracteur phrase a besoin d'un Dictionnaire en entrée; le dictionnaire phrase pourrait être construit automatiquement par les outils de phrase comme Xtract. La troisième est l'extracteur de terme, qui extrait des termes d'une phrase ou un document.

Il existe des bibliothèques utilitaires pour le développement d'application relatives aux tâches de la recherche d'information telles que les packages Dragon, LingPape, Lemur. De notre part, nous nous sommes basés dans la réalisation de notre application sur la bibliothèque fournie par Dragon.

4.2. Plateforme Dragon

Dragon est un package de développement basé sur le langage Java pour l'utilisation académique dans la recherche d'information (IR) et text mining (y compris la classification du texte, le regroupement de texte (text clustering,) et la modélisation des thèmes). Il est adapté pour les chercheurs qui travaillent dans IR et TM à grande échelle et préfèrent la programmation Java. En outre, il est différent de Lucene et Lemur, il fournit l'intégration des supports pour la IR sémantique et TM sémantique. La boîte à outils dragon intègre de façon transparente un ensemble d'outils NLP (naturel language processing), qui permettent

à la boîte à outils d'indexer les collections de documents avec des systèmes de représentation différentes, y compris des mots, des phrases, l'ontologie à base de concepts. La boîte à outils n'a pas certaines fonctionnalités, y compris IR distribués et IR cross-language qui est une partie de Lemur.

Une autre caractéristique importante du Dragon est son évolutivité. Contrairement à de nombreux outils de text mining comme Weka, la boîte à outils dragon est spécialement conçue pour les applications à grande échelle. La boîte à outils utilise une matrice creuse pour faire des représentations des textes et ne charge pas toutes les données en mémoire dans le temps d'exécution. Par conséquent, il peut gérer des centaines de milliers de documents dotés d'une mémoire très limitée.

5. Les corpus

5.1. Corpus JeuxVidéo

Les corpus rassemblés ont été utilisés lors de la campagne d'évaluation DEFT'07. Le corpus de tests de jeux vidéo (http://www.jeuxvideo.com) comprend environ 4 000 critiques. Chaque critique comporte une analyse des différents aspects du jeu - graphisme, jouabilité, durée, son, scénario - et une synthèse globale du jugement.

5.2. Corpus aVoiraLire

Les corpus rassemblés ont été utilisés lors de la campagne d'évaluation DEFT'07.

Le corpus de critiques de films, livres, spectacles et bandes dessinées (http://www.avoir-alire.com) comporte environ 3 000 critiques et les notes qui leur sont associées. En effet, beaucoup d'organes de diffusion de critiques de films ou de livres attribuent, en plus du commentaire, une note au film ou au livre sous une forme icônique.

6. L'environnement de programmation

6.1. Le langage JAVA

Le langage Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy, présenté officiellement le 23 mai 1995 au SunWorld. Java est un langage de programmation orienté objets basé sur le langage C++ mais avec des fonctionnalités qui en rendent la programmation plus simple et plus sure (absence de pointeur, gestion automatique de la

mémoire centrale, suppression des concepts complexes du C++, source de bugs (template, surcharge des opérateurs, opérateurs de conversion ...)).

6.2. L'IDE Netbeans

L'IDE Netbeans est un produit gratuit, utilisée pour la création d'application bureautique, les partenaires privilégiés fournissent des modules à valeur rajoutés qui d'intègrent facilement la plate forme et peuvent être utilisés pour développer des propres outils et solution.

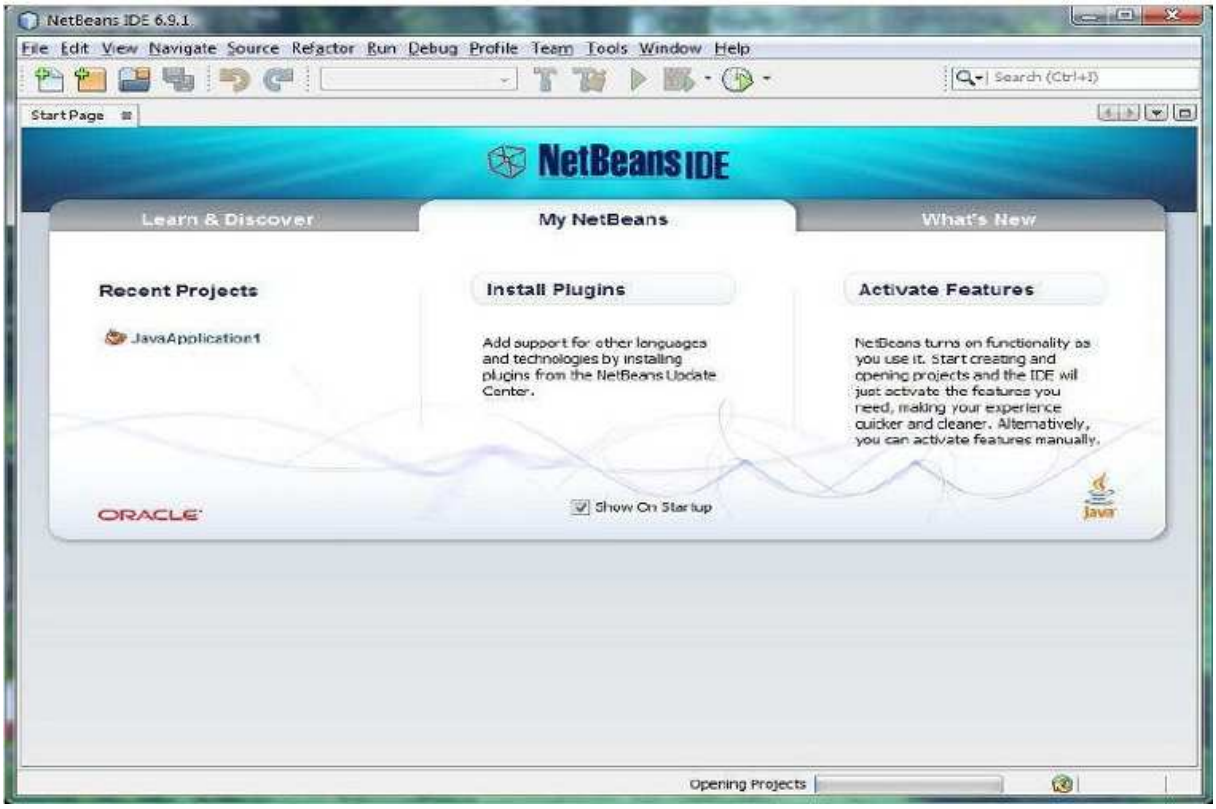

Figure 3.10 : l'environnement de développement intégré NetBeans

7. Conclusion

Dans ce chapitre, nous avons décrit le fonctionnement de notre application ainsi que la modélisation UML ensuite nous avons donné une idée générale sur les outils NLP et en précisent l'une de ces outils que nous avons utilisé durent notre application

Chapitre 4 :
Mise en œuvre et résultats

1. Introduction

Nous décrivons dans ce chapitre les aspects pratiques relatifs à la mise en œuvre de notre application pour l'évaluation de la Classification des opinions. Nous dressons ensuite des échantillons des résultats préliminaires obtenus à partir de trois corpus de références.

2. Choix du corpus

Pour effectuer une classification des opinions il faut choisir un corpus et ce dernier doit contenir une collection de document. Dragon fournit divers outils pour collecter, préparer, et de lire des collections. Tous les packages et les classes avec le préfixe "dragon.onlinedb" sont liées à cette ligne de fonctionnalités.

L'unité d'une collection est le document. Un document contient l'identificateur, le texte et une note. Le lecteur de la collection (collectionreader) est un agent qui lit les documents d'une collection. Souvent, les lecteurs de la collection travaillent avec l'analyseur d'article (articleparser). Lecteurs de la collection généralement identifier la limite et extraire le document de la collection et l'analyseur de l'article analyse le texte extrait brut en un document.

3. Corpus utilisés

3.1. Corpus 20Newsgroups

On appliquera, a titre illustratif, notre technique au corpus "20 Newsgroups" qui est constitue de 1997 documents, issus de 20 forums différents et décrits par 145,980 descripteurs. Ce corpus est devenu une référence sur laquelle des techniques de Text Mining telles que la catégorisation ou la classification non supervisée sont testées et comparées. Sa caractéristique essentielle est son hétérogénéité en termes de taille des documents, en termes de thématiques et en termes de style. [J. Ah-Pine, H. Benhadda, J. Lemoine (2005).]

3.2. Corpus Jeux vidéo

On a effectué la classification sur le corpus qui est utilisés lors de la campagne d'évaluation DEFT'07. Pour utiliser ce corpus, vous devez avoir signé les accords de restriction d'utilisation des

données de DEFT'07.L'objectif de la tâche était la classification de textes d'opinions. Le corpus de tests de jeux vidéo comprend environ 4 000 critiques. Chaque critique comporte une analyse des différents aspects du jeu - graphisme, jouabilité, durée, son, scénario - et une synthèse globale du jugement. Nous avons retenu une échelle de 3 niveaux de notes, qui donne les 3 classes : 0 (mauvais), 1 (moyen), et 2 (bien).

L'ensemble comporte un corpus d'entraînement (corpus_jeuxvideo_learn.xml) qui comporte les solutions, et un corpus de test (corpus_jeuxvideo_test.xml) dont les solutions sont dans le fichier corpus_jeuxvideo_test_classes.xml. (http://www.jeuxvideo.com).

3.3. Corpus avoir à Lire

Les corpus rassemblés ont été utilisés lors de la campagne d'évaluation DEFT'07. Pour utiliser ce corpus, vous devez avoir signé les accords de restriction d'utilisation des données de DEFT'07. L'objectif de la tâche était la classification de textes d'opinions. Le corpus de critiques de films, livres, spectacles et bandes dessinées comporte environ 3 000 critiques et les notes qui leur sont associées. En effet, beaucoup d'organes de diffusion de critiques de films ou de livres attribuent, en plus du commentaire, une note au film ou au livre sous une forme icônique. Nous avons retenu une échelle de 3 niveaux de notes, qui donne 3 classes bien discriminées : la classe 0 (mauvais), la classe 1 (moyen), et la classe 2 (bien).

L'ensemble comporte un corpus d'entraînement (corpus_aVoiraLire_learn.xml) qui comporte les solutions, et un corpus de test (corpus_aVoiraLire_test.xml) dont les solutions sont dans le fichier corpus_aVoiraLire_test_classes.xml. (http://www.avoir-alire.com).

4. Fenêtres d'exécution

4.1. Corpus

La figure 4.1 montre comment on peut visualiser le contenu d'un corpus (ex : jeuxvidéo), l'élément sélectionné est la collection des documents qui est affiché dans la zone du texte.

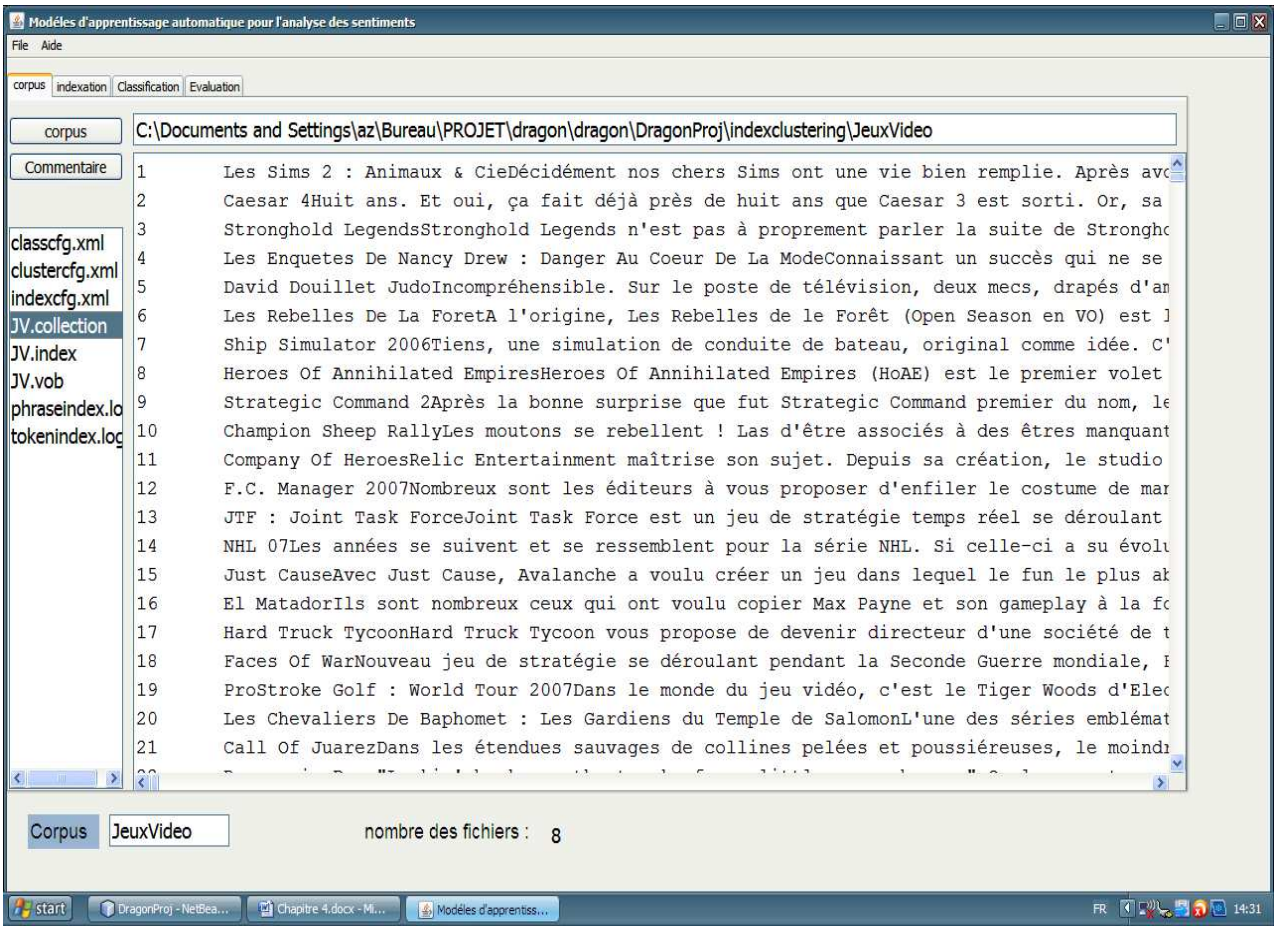

Figure 4.1 : Ouverture et lecture des différents fichiers du corpus «jeuxvideo»

Comme on peut visualiser le contenu du corpus en cliquant sur le bouton commentaire.

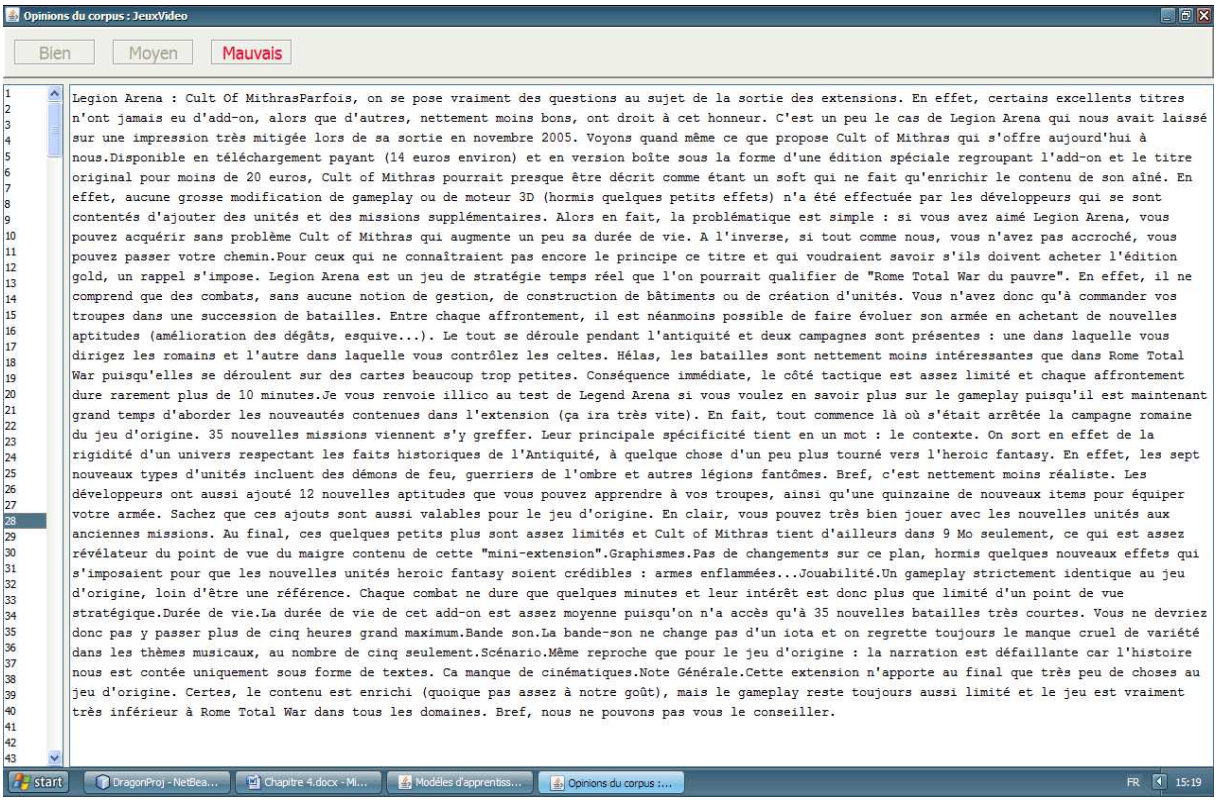

Figure 4.2 : affichage les opinions correspond à le corpus « jeuxvideo »

4.2. Indexation

L'indexation est l'opération qui vise à construire une structure d'indexe mais avant d'indexer un document il faut appliquer le processus de Traitement Automatique du langage naturel (NLP). Le package NLP fournit les fonctionnalités pour l'extraction des concepts différents ou des relations entre les articles lors de l' indexation. Les concepts peuvent être des mots, des phrases à plusieurs mots individuels, noms propres. Afin de faciliter l'extraction, dragon intègre divers outils de la NLP comme stemmers, une partie des tagueurs de la parole (speech taggers), les analyseurs de phrases (sentence parsers), extracteurs de phrase (phrase extractors), et nommé de reconnaissance de l'entité.

Pour faire l'indexation on a utilisé un fichier de configuration basé sur XML « indexcfg ». Ce fichier de configuration est placé dans le chemin corpus et comporte un nœud racine appelé configure. Le nœud racine peut contenir des nœuds d'objets multiples. Un nœud de l'objet correspond à un objet ou une application. Un nœud de l'objet à deux attributs obligatoires (type et id) et un attribut optionnel appelé la classe. Le type du nœud objet est souvent l' interface de l'objet outils et l' identifiant est un nombre entier pour distinguer des objets ganglions différents avec le même type. Ainsi, la combinaison de type et id doit être unique. Par exemple

pour indexer le corpus jeuxvideo on a spécifié le lecteur de la collection (BasicArticleParser).

```
<basiccollectionreader type="collectionreader" id="1">
        <param name="collectionpath" value="indexclustering/JeuxVideo"/>
        <param name="collectionname" value="JV"/>
        <param name="articleparser"
value="dragon.onlinedb.BasicArticleParser"/>
```

Comme on peut spécifie le type de l'indexation selon le paramètre basicindexer

```
<basicindexer type="indexer" id="1">
```

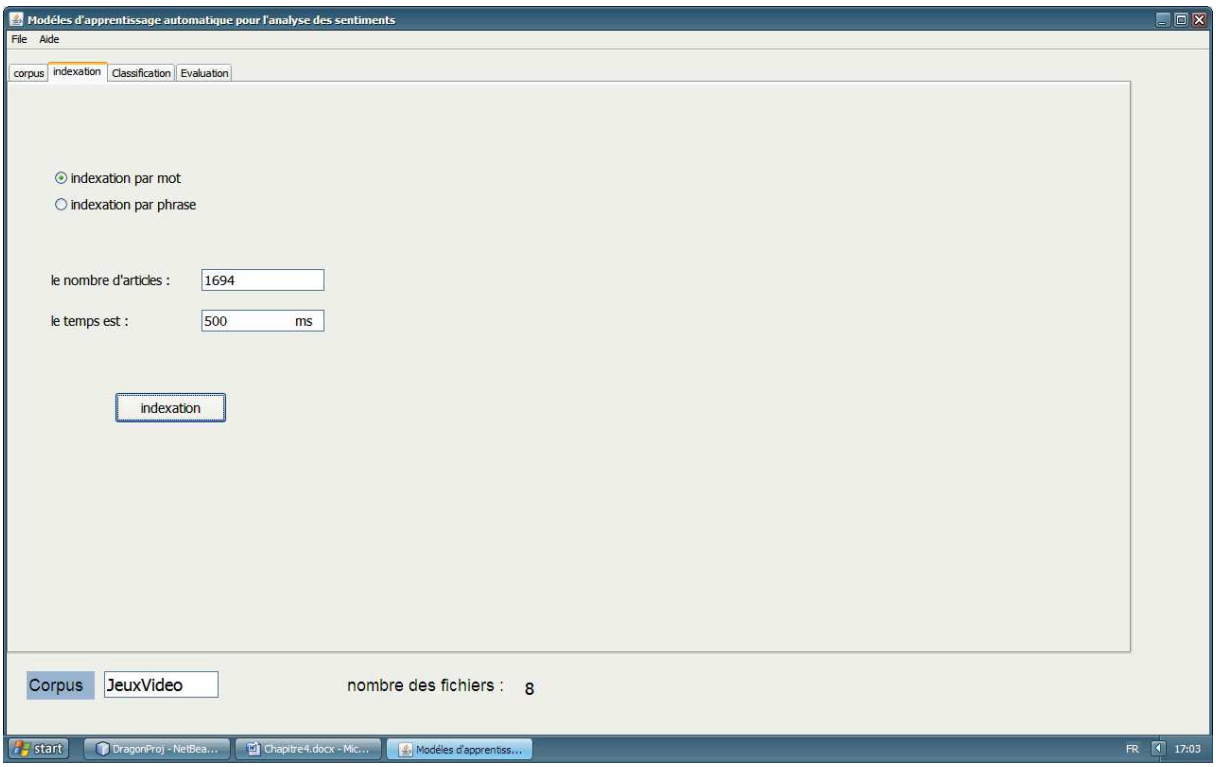

Figure 4.3 : l'indexation par mot du corpus « jeuxvideo »

Et le même travail est effectué pour les autres corpus newsgroup et Avoiralire sauf que nom de la collection est changé selon le corpus à indexer.

Il y a d'autre type d'indexation, l'indexation par phrase. L' indexation de base convertit les concepts extraits dans un indice basé sur des entiers et enregistrer les indices de concepts et de leur fréquence dans une matrice doc-terme; la fréquence du document inversé est enregistré dans une matrice terme-doc. L' indexation par phrase est presque la même que l' indexation de base, sauf qu' il sépare un article en phrases d'abord, puis traite chaque phrase comme un document à indexer. En d'autres termes, dans la matrice doc-term entraîné par l'indexation phrase, chaque le document désigne une phrase.

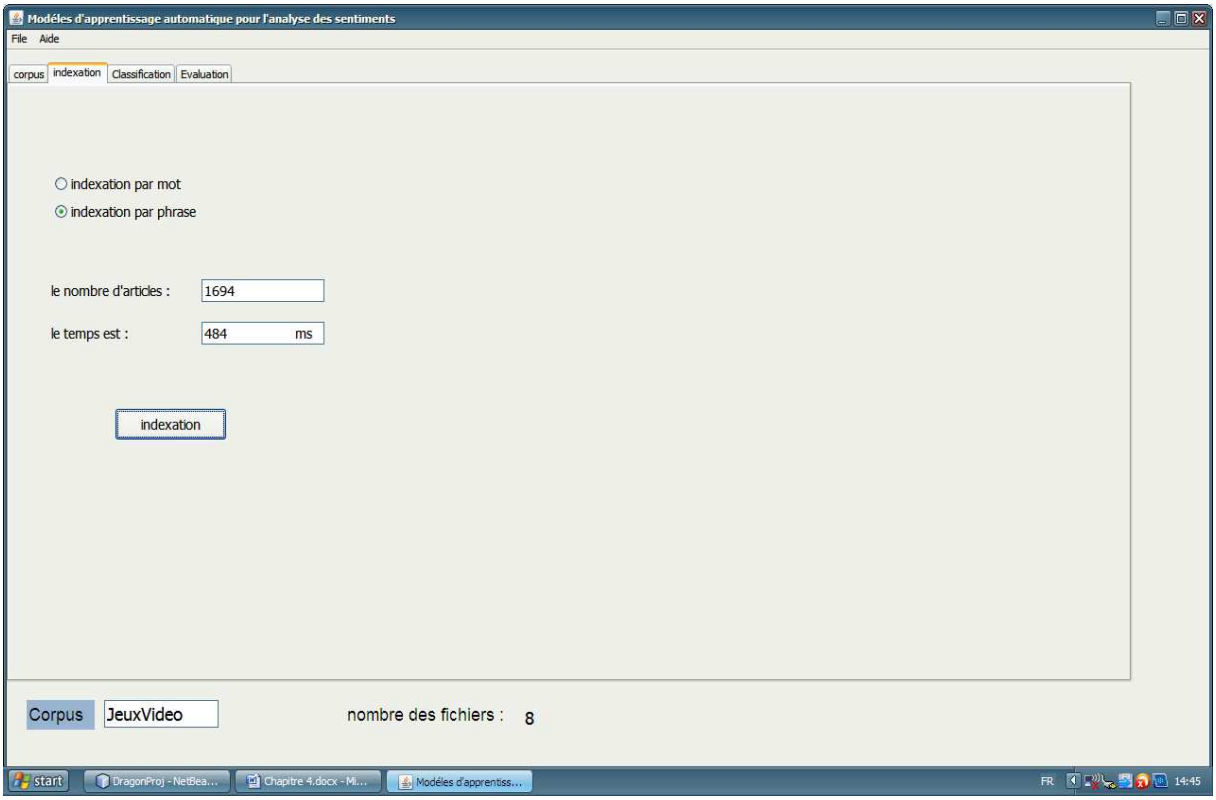

Figure 4.4 : indexation par phrase du corpus « jeuxvideo »

4.3. La classification

Après l'indexation des documents, deuxième partie nécessaire pour Analyse des sentiments est la classification des opinions, cette étape consiste à choisir le classifier qu'on veut utiliser : SVMLight, Naïve bayésien, ces derniers utilisent aussi des modes pour faire la classification tel que le mode pourcentage qui est le mode par défaut, prendre comme paramètre le ''1'' ou bien le mode

cross validation qu'a le paramètre ''7''. Ces paramètres se trouvent dans fichier XML « classcfg.xml » dans chaqu'un des corpus.

- Le classifier SVMLight avec le mode « crossvalidation » :

```
<classificationevaapp type="classificationevaapp" id="7">
            <param name="classifier" type="classifier" value="6"/>
            <param name="classnum" value="20"/>
            <param name="mode" value="CrossValidation"/>
            <param name="answerkeyfile"
value="indexclustering/Newsgroup/experiment/answerkeys_large.list"/>
            <param name="outputfile"
value="indexclustering/Newsgroup/experiment/20ng.result"/>
            <param name="runs" value="10"/>
            <param name="percentage" value="0.01"/>
            <param name="randomseed" value="10"/>
            <param name="runname" value="svm_1 %"/>
      </classificationevaapp>
•     Le classifieur Naive Bayésien avec le mode « pourcentage » :
<classificationevaapp type="classificationevaapp" id="1">
            <param name="classifier" type="classifier" value="1"/>
            <param name="classnum" value="20"/>
            <param name="mode" value="percentage"/>
            <param name="answerkeyfile"
value="indexclustering/Newsgroup/experiment/answerkeys_large.list"/>
            <param name="outputfile"
value="indexclustering/Newsgroup/experiment/20ng.result"/>
            <param name="runs" value="10"/>
            <param name="percentage" value="0.01"/>
            <param name="randomseed" value="10"/>
            <param name="runname" value="lap_1 %"/>
      </classificationevaapp>
```

Les principales métriques de la classification en utilisant les deux classifier sont implémentés, le micro précision, macro précision, micro rappel, macro rappel, micro F_mesure et la macro F_mesure. Ces mesures sont calculées pour chaque classifier, pour chaque corpus sélectionné et sont affichés dans un tableau.

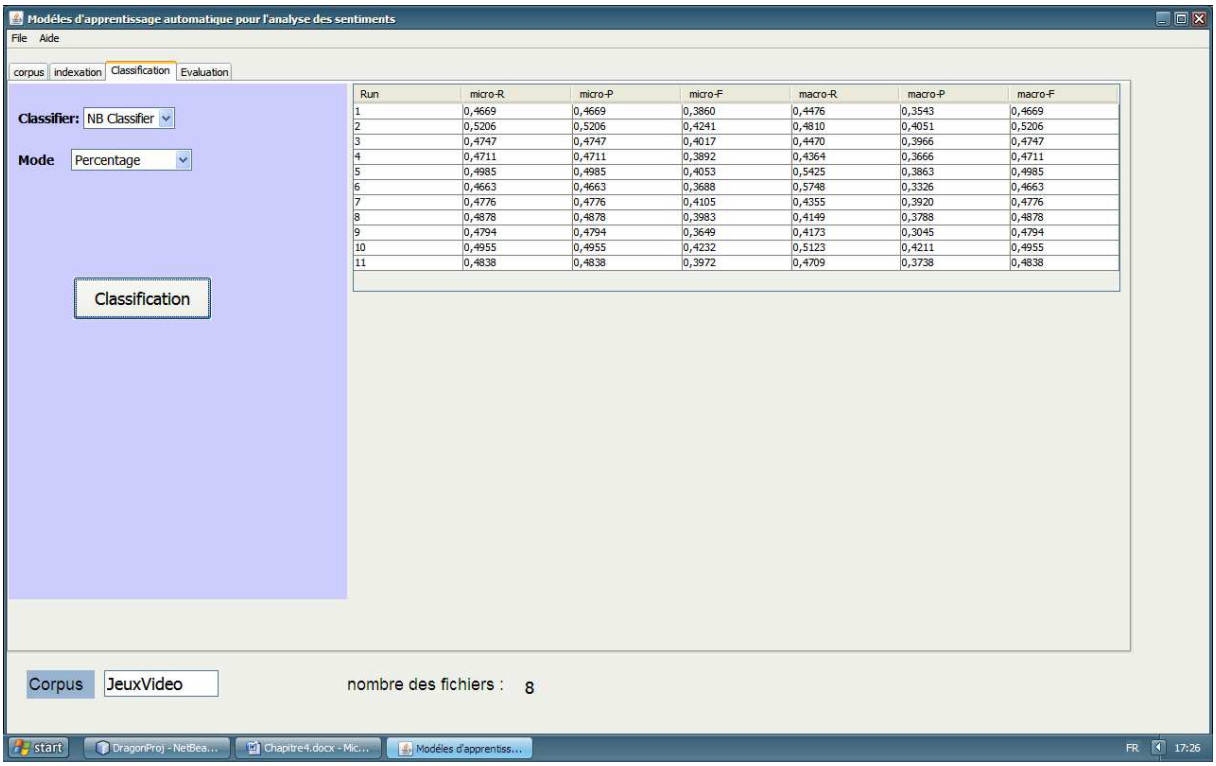

Figure 4.5 : classification du corpus « jeuxvideo » avec le classificateur naive bayésien et le mode pourcentage

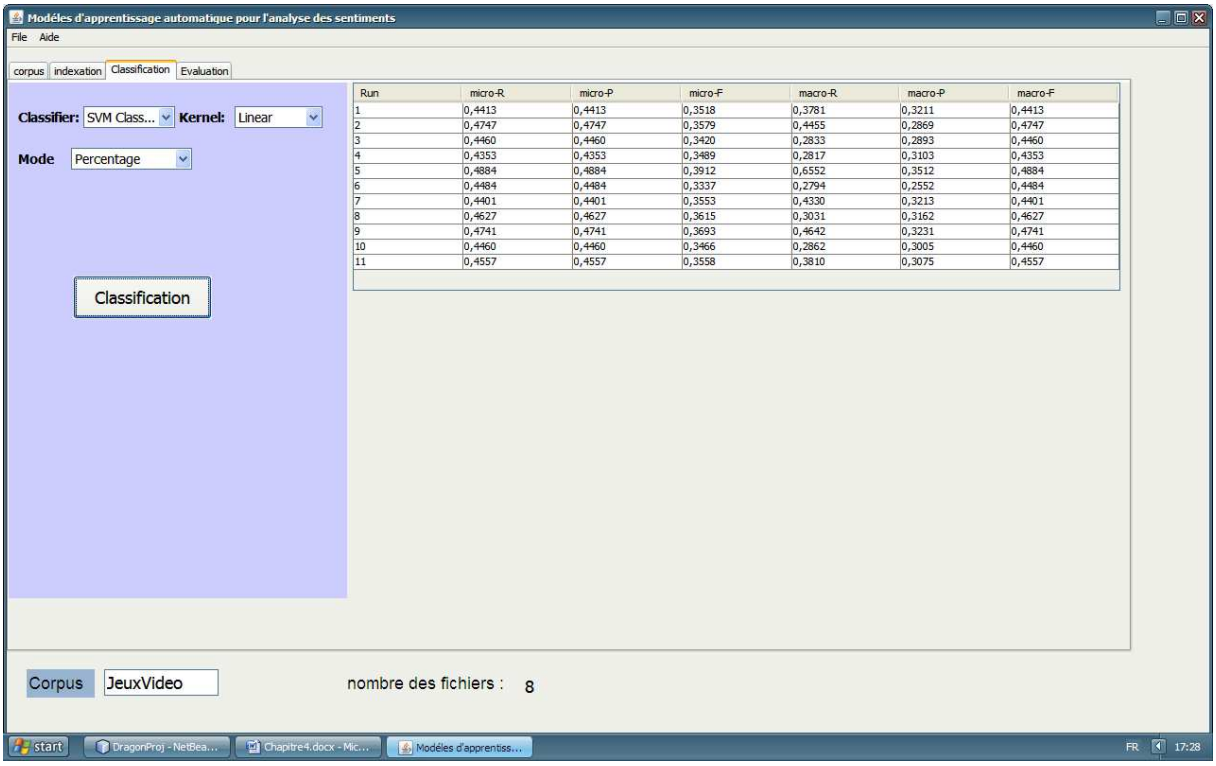

Figure 4.6 : classification du corpus jeuxvideo avec le classificateur SVM et le mode pourcentage

4.4. L'évaluation

Après l'indexation des documents et la classification, on peut tout simplement appeler le programme d'évaluation correspondant à représenter graphiquement les résultats de la classification par des histogrammes dans le but de comparer entre les deux classifier, le premier histogramme représente le micro précision de SVMLIGHT et Naïve bayésien, la macro précision, le troisième le micro rappel, le quatrième la macro rappel , toujours des deux calssifiers SVMLIGHT et Naive bayésien.

La dernière valeur calculée «average » du tableau qui est la moyenne de chaque colonne donne la représentation graphique des mesures calculées.

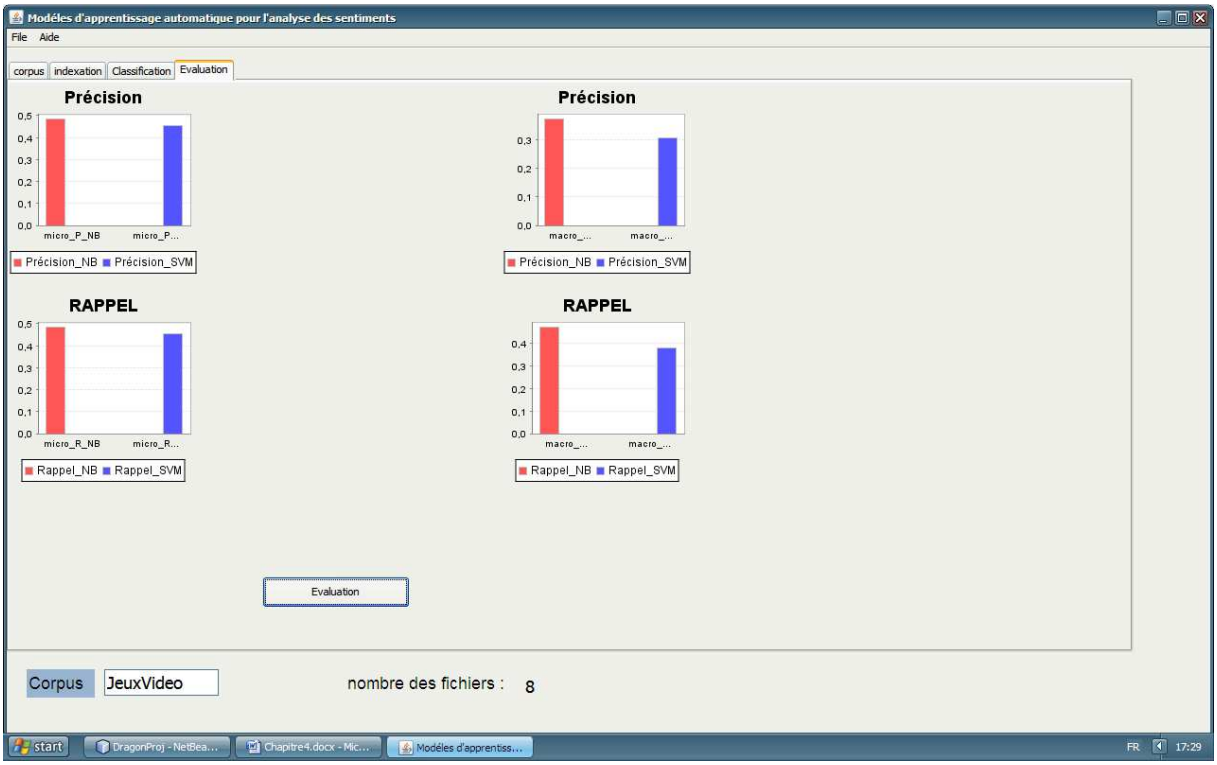

Figure 4.7 : les mesures moyennes (précision et rappel) d'évaluation du corpus « jeux vidéo »

5. Résultats

A partir de l'exécution de la classification selon deux classifieurs on peut extraire les résultats illustrés dans le tableau 1, et 2.

Corpus	Micro_R	Micro_P	Micro_F	Macro_R	Macro_P	Macro_F
Newsgroup	0,4272	0,4272	0,4272	0,4269	0,4885	0,4215
A voir à lire	0,5446	0,5446	0,5446	0,3346	0,2935	0,2584
Jeux video	0,4838	0,4838	0,4838	0,3972	0,4709	0,3738

Tableau 13: les résultats de la classification par le classifier Naïve bayésien

Corpus	Micro_R	Micro_P	Micro_F	Macro_R	Macro_P	Macro_F
Newsgroup	0,4714	0,4714	0,4714	0,4713	0,4759	0,4637
A voir à lire	0,5538	0,5538	0,5538	0,3332	0,1846	0,2376
Jeux video	0,4557	0,4557	0,4557	0,3558	0,3810	0,3075

Tableau 14: les résultats de la classification par le classifier SVMLight

Apres l'étape de l'évaluation de deux classificateurs que nous avons appliqué sur les trois corpus on peut déduire que Les meilleures performances de classification des opinions est varié selon le corpus choisie tel que :

- le corpus newsgroup a atteint plus d'amélioration en appliquant le classifier SVM.

- le corpus Avoir à lire a atteint plus d'amélioration en appliquant le classifier SVM pour les mesures d'évaluation (micro) et pour les mesures d'évaluation (macro) en appliquant le classifier naive bayésien atteint plus d'amélioration.

-le corpus jeux video a atteint plus d'amélioration en appliquant le classifier naive bayésien.

6. Conclusion

Ce chapitre a été consacré à la mise en œuvre de notre application et à la visualisation des résultats d'évaluation d'un modèle d'apprentissage automatique pour l'analyse des sentiments en appliquant la classification sur plusieurs corpus. Les expérimentations menées dans ce projet nous ont permises de valider notre application pour intégrer un corpus, l'indexer (par mot ou bien par phrase) et enfin d'afficher les résultats d'évaluation. Les représentations graphiques offertes par notre application permettent de mieux comparer et analyser différent classificateurs.

Conclusion générale

Ce projet décrit l'étude et la réalisation des modèles d'apprentissage automatique conçu pour l'évaluation des sentiments des corpus. Un tel modèle permet:

- la recherche automatique des critiques sur Internet,
- la classification des opinions en utilisant des méthodes d'apprentissage automatique supervisé non supervisé.
- l'évaluation et la notation des opinions des critiques.

La partie la plus intéressante au niveau de ce mémoire est la notation automatique des sentiments. Pour cette partie nous avons présenté trois approches différentes pour effectuer la classification des sentiments. Nous avons cité les approches suivantes :

- l'approche linguistique,
- l'approche statistique,
- l'approche hybride.

L'objectif de ce travail était d'étudier, expérimenter et comparer différents clssificateurs. Nous avons implémenté une application permettant de tester deux classificateurs : SVMLight et Naïve Bayésien sur, En les appliquant sur deux corpus d'opinions, nous avons opté à effectuer l'indexation par mot ou bien l'indexation par phrase après un paramétrage standard via un fichier XML. Chaque corpus indexé, peut être par la suite classifié selon un modèle avant d'être évalué. Cette évaluation est basée sur deux métriques principales (précision, rappel).

Les outils d'analyse offerts par notre application permettent d'apprécier les performances d'un modèle sur différents corpus de tests. Nous estimons que, par ce modeste travail, nous avons pu déblayer le terrain dans un domaine très intéressant et d'actualité en vue pouvoir suivre les opinions du grand public dans les blogs et les réseaux sociaux et bien même d'améliorer les systèmes de veille stratégique dans l'entreprise.

Bibliographie

1. [Blei et al, 2003] : D. Blei, A. Ng, et M. Jordan, 2003. Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022.
2. [Benamara, F., C. Cesarano, A. Picariello, D. Reforgiato, ET V. Subrahmanian (2007)]. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In International Conference onWeblogs and SocialMedia (ICWSM), Boulder, Colorado, U.S.A, 26/03/2007-28/03/2007, http: //www.aaai.org/Press/press.php, pp. 203–206. AAAI Press.
3. [Boullé, M. (2005). A Bayes] optimal approach for partitioning the values of categorical attributes.Journal of Machine Learning Research 6, 1431–1452.
4. [Boullé, M. (2006). MODL: a Bayes] optimal discretization method for continuous attributes. Machine Learning 65(1), 131–165.
5. [Boullé, M. (2007)]. Compression-based averaging of selective naive Bayes classifiers. Journal of Machine Learning Research 8, 1659–1685.
6. [Carbonell, J. (1979)]. Subjective Understanding: Computer Models of Belief Systems. Phd thesis, Yale.
7. [Cohen, W. (1996)]. Learning trees and rules with set-valued features. In In Proceedings of the 13th National Conference on Artificial Intelligence, pp. 709–716. AAAI Press.
8. [Das, S. et M. Chen (2001)]. Yahoo! for amazon: Extracting market sentiment from stock message boards. Proceedings of the Asia Pacific Finance Association Annual Conference APFA).
9. [Ding, X. ET B. Liu (2007)]. The utility of linguistic rules in opinion mining. In SIGIR ’07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, pp. 811–812. ACM.
10. [Dziczkowski, G. et K. Wegrzyn-Wolska (2008)]. An autonomous system designed for automatic detection and rating of film reviews. Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on 1, 847–850.
11. [Guimier de Neef, E., M. Boualem, C. Chardenon, P. Filoche, et J. Vinesse (2002)]. Natural language processing software tools and linguistic data developped by france télécom rd.
12. [Hand, D. et K. Yu (2001).] Idiot bayes? Not so stupid after all? International Statistical Review 69(3), 385–399.
13. [Hatzivassiloglou, V. et K. R. McKeown (1997)]. Predicting the semantic orientation of adjectives. In Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics, Morristown, NJ, USA, pp. 174–181. Association for Computational Linguistics.
14. [Hu, M. et B. Liu (2004a)]. Mining and summarizing customer reviews. In KDD ’04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, pp. 168–177. ACM.

15. [Dumais 1995] : Dumais.S(1995). Latent Semantic Indexing (LSI), TREC-3 Report In proceedings of TREC-3, pages: 319-230.
16. [Dziczkowski, D. (2008)] :Dziczkowski, D. (2008). Analyse des sentiments : système autonome d'exploration des opinions exprimées dans les critiques cinématographiques.
17. [El charif, R. (2006)] : El charif, R. (2006). analyse des paramètres de pondération dans le cadre de collections volumineuses. pages 10-18.
18. [Girolami and Kaban, 2003]:Girolami. Mark, Kaban. Ata. (2003). « On an equivalence between PLSI and LDA » Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 433-434.
19. [Kuhn, 1960]: Maron, M. E., & Kuhn, J. L. (1960). On relevance, probabilistic indexing, and information retrieval. Journal of the Association for Computing Machinery, 7(3), 216-244
20. [Mitchell (1996)]: Mitchell, T.M. 1996. Machine Learning. McGraw Hill. 12, 14, 27
21. [Robertson, 1977]: Robertson, S. E. (1977). The probability ranking principle in IR. Journal of Documentation, 33 (4), 294-304
22. [Robertson et al., 1976]: S.E. Robertson, K. Sparck Jones, Relevance weighting of search terms. Journal of the American Society for Information Science, pages 129 à 146, maijuin1976.
23. [Zabin&Jefferies, 2008]: Zabin, J, & Jefferies, A. 2008(January). Social Media Monitoring and Analysis: Generating Consumer Insights from Online Conversation. Aberdeen Group Benchmark Report. 3
24. http://dragon.ischool.drexel.edu/