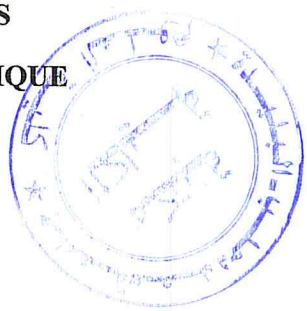
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UVIVERSITE SAAD DAHLEB BLIDA

FACULTE DES SCIENCES

DEPARTEMENT INFORMATIQUE



MEMOIRE DE MASTER

DOMAINE : MATHEMATIQUE ET INFORMATIQUE

FILIERE : INFORMATIQUE

CONCEPTION ET IMPLEMENTATION D'UN SYSTEME D'ANALYSE ET DE CLASSIFICATION DES COURRIERS ELECTRONIQUES

Présenté par :

M. CHETTIH Djaafar

M. TOUAIBI Belgassim

Dirigé par :

Mme. L.GUEMRAOUI (CERIST)

Dr. O.NOUALI (CERIST)

M. M.R.SIDOUMOU (USDB)

Soutenu le :

Devant le jury composé de :

M/ Mme.	Nom/Prénom	Grade	Président(e) Mr. Hamouda.
M/ Mme.	Nom/Prénom	Grade	Examinateur (ice) Mr. BALA
M/ Mme.	Nom/Prénom	Grade	Examinateur (ice) Mr. H. yahia.

2012 – 2013

Dédicaces

A mes chères Parents,

A mes frères et sœurs,

A tous mes amis,

Je dédie ce modeste mémoire.

BELGASSIM

A mes chères Parents,

A ma famille,

A tous mes amis,

Je dédie ce modeste mémoire.

DJAAFER

Résumé

Actuellement le web est utilisé comme source d'informations et moyen de communication, le courriel électronique est devenu l'un des outils d'échange d'informations les plus utiliser sous le web. L'utilisateur se retrouve donc face à une grande masse de message qui provient de sources différentes. Pour faciliter la tâche à l'utilisateur et lui éviter de perdre trop de temps à gérer ses courriels, nous proposons une solution qui analyse et classifie les courriels automatiquement.

Remerciements

Louange à Allah tout puissant qui nous a guidés pour l'accomplissement de ce modeste travail.

Un grand merci à Mme. GUEMRAOUI Lila qui nous a encadrés au cours de ces six mois durant lesquels son aide précieuse, ses conseils et orientations nous ont été d'un précieux apport afin de mener à beau et à bien ce travail que nous espérons être dignes de la confiance qu'elle a placée en nous.

Nous tenons à remercier particulièrement M.SIDOUMOU Ridha pour sa gentillesse et la confiance qu'il nous a accordée pour la réalisation de ce mémoire.

Notre reconnaissance va tout naturellement à Mme. MEKHZOUMI Dalila pour son orientation et ses minutieux conseils qui nous ont aidés pour la réalisation de notre travail.

Nous tenons aussi à remercier M.NOUALI OMAR, qui nous a ouvert les portes.

Nous aimerons exprimer notre gratitude aux personnes qui nous ont fait l'honneur de participer au jury de ce mémoire.

Enfin nous adressons nos sincères remerciements à toute personne qui nous a soutenus et encouragés au cours de la réalisation de notre projet.

Résumé

Actuellement le web est utilisé comme source d'informations et moyen de communication, le courriel électronique est devenu l'un des outils d'échange d'informations les plus utiliser sous le web. L'utilisateur se retrouve donc face à une grande masse de message qui provient de sources différentes. Pour faciliter la tâche à l'utilisateur et lui éviter de perdre trop de temps à gérer ses courriels, nous proposons une solution qui analyse et classifie les courriels automatiquement.

Mots Clés : *NFCE, courriel électronique, classification.*

Table des matières

LISTE DES FIGURES

LISTE DES TABLEAUX

ABREVIATION

NFCE	:	*Nouvelles Formes de Communication Ecrite*
SMS	:	*Short Message Service*
POP	:	*Post Office Protocol*
IMAP	:	*Internet Message Access Protocol*
MIME	:	*Multipurpose Internet Mail Extensions*
SMTP	:	*Simple Mail Transfer Protocol*
IMC	:	Internet Mail Consortium
IETF	:	Internet Engineering Task Force
IP	:	Internet Protocol
E-MAIL	:	Electronic Mail
TF-IDF	:	Term Frequency-Inverse Document Frequency
TIC	:	Technologies de l'information et de la communication
UML	:	Unified Modling language

INTRODUCTION GENERALE

INTRODUCTION GENERALE

CONTEXE DU MEMOIRE

Parmi la gamme des nouvelles réalités que rend possible l'internet, le courrier électronique est sans doute celle qui changera le plus nos habitudes. La croissance de l'internet est directement reliée à l'importance grandissante du courrier électronique.

Plusieurs sites web lui sont maintenant consacrés et il est même possible d'obtenir un diplôme entièrement en ligne. Presque tous les gens qui ont accès à l'internet ont au moins une adresse de courrier électronique qu'ils vérifient quotidiennement.

En le comparant aux autres façons de communiquer (par écrit, par téléphone et en personne), on s'aperçoit que les nombreux avantages du courrier électronique surpassent de loin ses inconvénients. Sa grande force réside dans son medium de transport.
La rapidité à laquelle les courriers circulent, combinée à la possibilité de les envoyer à plusieurs personnes en même temps, améliore considérablement la productivité des groupes de travail séparés par des endroits différents et fuseaux horaires opposés.

Pour les entreprises, le courrier électronique apporte une toute nouvelle dimension au service de la clientèle, du point du vue du marketing, les compagnies peuvent rejoindre plus facilement leurs client et leur envoyer des annonces de produits, des offres spéciales, etc. à moindre cout. Elles peuvent également personnaliser le contenu selon les préférences de chaque client, et fournir l'information en totalité ou en partie, avec des hyperliens pour avoir plus de détails. Mais plus important encore, le courrier électronique offre la possibilité aux entreprises de se rendre plus accessibles à leurs clients.

PROBLEMATIQUE

Cependant, il n'y a pas que des avantages au courrier électronique, parmi les problèmes majeurs, sont face au volume des courriers électroniques, on retrouver le problème d'organisation au sein de l'entreprise, dépassement de délais de réponse, autrement dit nous nous plaçons dans le cas ou une boite aux lettres reçoit un grand nombre de courriers (plus d'une centaine par jour) correspondant à plusieurs thématiques. Gérer et répondre à tous ces courriers devient une tâche extrêmement difficile (voir, impossible dans certain cas).

Un autre inconvénient, est la nature du langage électronique, qui se caractérise par une prise de liberté par rapport aux canons de l'écrit classique. La phrase ci-dessous présente un extrait d'e-mail :

'' *Slt tu vien de fere un tour sur mon blog alor tou dabor jte di merci...^^* ''.

Le traitement automatique de tels énoncés est à peu près impossible avec des outils standards.

SOLUTION PROPOSEE

Dans le cadre de ce mémoire, nous proposons de :

1. Analyser les caractéristiques du courrier électronique et transformation de texto vers texte normal.
2. Etudier la possibilité de bénéficier des différents travaux liés au traitement automatique du texte pour accueillir la compréhension des e-mails.
3. définir une représentation normalisée et pertinente des e-mails, sur la quelle nous pouvons appliquer la classification automatique du texte existants.
4. Etablir une classification des e-mails entrants.
5. Evaluer la solution sur un corpus existant.

ORGANISATION DU MEMOIRE

L'organisation retenue est la suivante :

Chapitre 1 : *Etat de l'art*

Présente le domaine dans lequel s'inscrit notre travail, il est composé de deux parties :

Partie 1 - Classification *automatique du texte* : introduit les notions de base et les principales démarches à suivre pour la classification automatique du texte.

Partie 2 - *Courrier électronique* : Consacré à la définition des principales caractéristiques du courrier électronique.

Chapitre 2 *: Conception*

Ce chapitre réunis tous ce qui concerne la conception du système, incluant l'expression des besoins sous forme de diagramme de cas d'utilisation, et une conception plus détaillée sous forme de diagramme de classe et de séquence.

Chapitre 3 : *Mise en œuvre*

Ce chapitre englobe la partie de mise en œuvre du système, il aborde la description de l'outil ainsi que le développement de l'application.

Chapitre 4 : *Validation*

Ce chapitre introduit dans un premier temps l'étude du processus (*Analyse et classification*) appliqué à un corpus d'essai, et se termine par une évaluation des performances du système.

Conclusion générale : Finalement, une conclusion conclut ce mémoire.

CHAPITRE I : ETAT DE L'ART

CHAPITRE I : ETAT DE L'ART

I. INTRODUCTION

Dans ce chapitre, l'ensemble des concepts et méthodes utilisés dans notre travail sont présentés, il comporte deux parties :

- *La première partie,* aborde la description de la classification automatique du texte, une description des approches de la classification automatique, et se termine par un schéma général du processus de classification.
- *La deuxième partie*, aborde la définition du courrier électronique et ses principales caractéristiques.

II. PARTIE I - CLASSIFICATION AUTOMATIQUE DU TEXTE

1. INTRODUCTION

La classification automatique de textes consiste à regrouper de manière automatisée des documents qui se ressemblent suivant certains critères a savoir les critères observables tels que le type du document, l'année, la discipline, l'édition, etc.… ou le critère du contenu.

Elle connait ces derniers temps un fort regain d'internet. Cela est du essentiellement a la forte croissance des documents numériques disponibles et a la nécessité de les organiser de façon rapide.

La classification de textes est une tache générique qui consiste à assigner une ou plusieurs catégories, parmi une liste prédéfinie, ou non a un document.

Actuellement, la classification de textes est un domaine de recherche très actif et l'automatisation de cette opération est devenue un enjeu pour la communauté scientifique, les travaux évoluent considérablement depuis une vingtaine d'année et plusieurs modèles ont vu le jour.

2. APPROCHES DE CLASSIFICATION AUTOMATIQUE

L'objectif de la classification de textes est de rassembler les textes similaires selon un certain critère, au sein d'une même classe.

Deux type d'approches de classification automatique peuvent être distingues :

La classification supervisée et la classification non supervisée. Ces deux méthodes différentes sur la façon dont les classes sont générées. En effet dans le cas de la classification non supervisée, les classes sont calculées automatiquement par la machine, par contre, dans l'approche supervisée, la classification de textes consiste a rattacher un texte a une ou plusieurs catégories prédéfinies par un expert, ces catégories pouvant être par exemple le sujet du texte, son thème, l'opinion qui y est exprimée, …etc.

- ***Dans ce qui suit, notre travail va être concentré sur la classification supervisée de textes (la catégorisation).***

3. CLASSIFICATION SUPERVISEE

3.1. INTRODUCTION

La classification supervisée ou la *catégorisation* de textes correspond à la procédure d'affectation d'une ou de plusieurs catégories ou classes prédéfinies à un texte. Elle correspond à la *classification supervisée*

Cette problématique a par ailleurs dernièrement trouvé de nouvelles applications dans les domaines du traitement du langage tels que : l'affectation de sujets en recherche d'information, l'aide de l'utilisateur pour l'indexation de documents, etc.

Aujourd'hui, cette problématique utilise largement des méthodes issues de l'apprentissage automatique et beaucoup d'algorithmes d'apprentissage supervisé lui ont été appliqués (Naïve bayes, K-plus proches voisins, arbres de décision, machines à vecteurs support, réseaux de neurones, etc.)

3.2. DEFINITIONS

DEFINITION 1 : CATEGORISATION DE TEXTES

Dans sa forme la plus simple, La catégorisation de Textes (C.T) est le processus qui consiste à assigner une ou plusieurs catégories parmi une liste prédéfinie à un document. L'objectif du processus est d'être capable d'effectuer automatiquement les classes d'un ensemble de nouveaux textes.

DEFINITION 2 : CLASSE

La notion de classe pour un système de classification a été habituellement synonyme de «thème ». Dans ce contexte, classer les documents revient à les organiser par différentes thématiques. (Par exemple : *Earn, Ship, Trade,* correspondent à des thèmes dans le corpus Reuters).

Un système de classification d'emails est représenté sur la figure 1.1, où les classes peuvent être de différentes natures (thèmes, messages provenant de certaines personnes, messages d'un certain type, etc…).

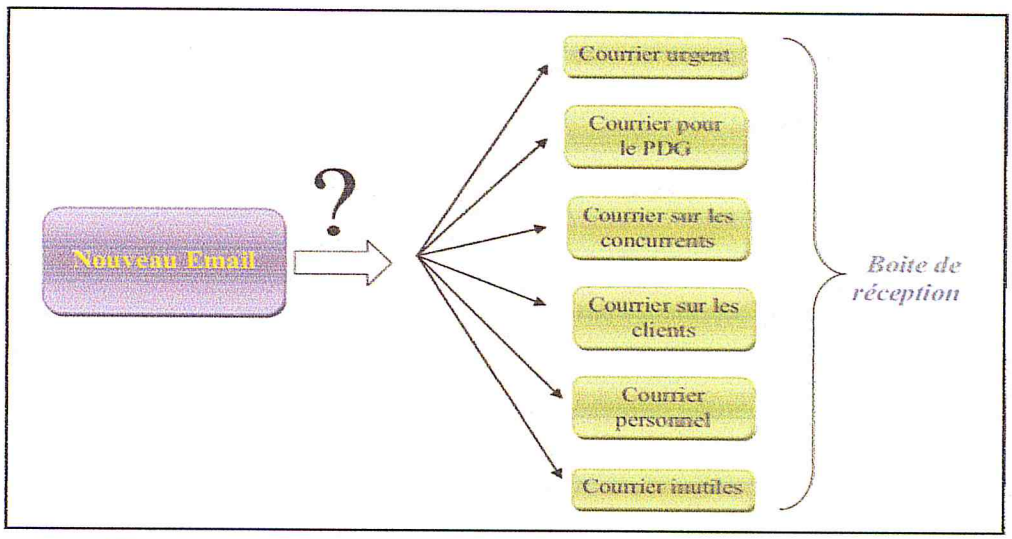


Figure 1.1 : Exemple de système de classification d'emails

Ce système organise des emails dans des boîtes aux lettres qui correspondent chacune à une classe qui sont de différentes natures (« mails urgents », « Mails du Directeur général », etc.…).

3.2 PROCESSUS DE CATEGORISATION DE TEXTES

Pour réaliser l'opération de catégorisation automatique de textes comme nous l'avons défini, la démarche commune est la suivante : la première phase consiste donc à formaliser les textes afin qu'ils soient compréhensibles par la machine et utilisables par les algorithmes.

La catégorisation des documents est la deuxième phase, qui peut être réalisé comme suit :

- Prétraitement.
- Analyseur linguistique
- Définition des descripteurs.
- Classification

3.2.1 PRETRAITEMENT

Le prétraitement des textes est une phase capitale du processus de classification, puisque la connaissance imprécise de la population peut faire échouer l'opération. Il consiste à expurger le plus possible les informations inutiles des documents afin que les connaissances gardées soient aussi pertinentes qu'il se peut.

Le prétraitement est également effectue en quatre étapes séquentielles :

1- La segmentation.

2- Suppression des mots fréquents.

3- Suppression des mots rares.

4- Traitement morphologique.

- ***Dans certain travaux, le prétraitement comporte, en plus des quatre étapes citées ci-dessus, le traitement syntaxique et sémantique, qui nécessite un travail laborieux, et qui n'est pas aujourd'hui bien maitrisé.***

  Par ailleurs, vu les objectifs visés par ce travail, et vu les contraintes de temps imposées, ces deux étapes ne seront pas pries en compte dans ce qui suit.

3.2.1.1. SEGMENTATION

La première opération que doit effectuer un système de classification est la reconnaissance des termes utilisés. La segmentation consiste à découper la séquence des caractères afin de regrouper les caractères formant un même lot.

Habituellement, cette étape permet d'isoler les ponctuations (reconnaissance des fins de phrase ou de paragraphe), ensuite découper les séquences de caractères en fonction de la présence ou l'absence de caractères de séparation (de type « espace », « tabulation »ou « retour a la ligne »), puis regrouper les chiffres pour former des nombres (reconnaissance éventuelle des dates), de reconnaitre les mots composes.

Eventuellement, nous pouvons unifier les écritures en lettre majuscules ou en lettres minuscules avant ou âpres les opérations déjà indiquées.

3.2.1.2. SUPPRESSIONS DES MOTS FREQUENTS OU ELIMINATION DES « MOTS OUTILS »

Les mots qui apparaissent le plus souvent dans un corpus sont généralement les mots grammaticaux, mots vides (empty words) ou mots outils (stop words) : les articles, les

prépositions, les mots de liaisons, les déterminants, les adverbes, les adjectifs indéfinis,…etc, qui constituent une grande part des mots d'un texte, mais malheureusement sont faiblement informatifs, sur le sens d'un texte puisqu'ils sont présents sur l'ensemble des textes. Ces termes très fréquents peuvent être écartés du corpus pour réduire la dimension.
L'élimination systématique du corpus des mots vides peut se faire par l'intermédiaire d'une liste prédéfinie de mots pour chacune des langues étudiées.

3.2.1.3. SUPPRESSION DES MOTS RARES

En général, les auteurs cherchent également à supprimer les mots rares, qui n'apparaissent qu'une ou deux fois sur corpus, afin de réduire de façon appréciable la dimension des vecteurs utilises pour représenter les textes.

3.2.1.4. TRAITEMENT MORPHOLOGIQUE

Consiste à effectuer un traitement au niveau de chacun des mots en fonction de leurs variations morphologiques : flexion, dérivation, composition afin de rassembler les mots de sens identiques. Plusieurs traitements morphologiques existent :

- **Le stemming** ou **la desuffixation** regroupe sous un même terme (stem) les mots qui ont la même racine.
- **Lemmatisation** conserve, non pas les mots eux-mêmes, mais leur racine ou lemme.

Toutes les études montrent que les performances des systèmes de classification, après lemmatisation, sont plus nettement supérieures à celles avant lemmatisation.

3.2.2. DEFINITION DE DESCRIPTEURS

La définition ou l'extraction de caractéristiques au sein d'un texte est une phase décisive puisque la représentation déduite doit conserver au mieux l'information contenue dans le texte. Ces caractéristiques constituent les éléments informationnels composant le document.

Différentes méthodes sont proposées pour le choix des termes et les poids attribues à ces termes, des auteurs utilisent les mots comme descripteurs, d'autre utilisent les groupes de mots comme les mots composes, les expressions ou les collocations.

- ***Dans le cadre de ce mémoire, on a choisi de travailler avec la méthode « sac de mots », les deux raisons de ce choix sont :***
  - ✓ ***d'une part, le fait que cette méthode est largement utilisée, supportée par divers outils et sur laquelle, divers travaux se basent,***
  - ✓ ***et d'autre part, le fait que sont implémentation est simple, et adaptable aux documents de petite taille (cas des courriers électroniques).***

3.2.2.1. REPRESENTATION EN « SAC DE MOTS »

La représentation des textes la plus simple a été introduite dans le cadre du modèle vectoriel et porte le nom de "sac de mots" '' Bag-of-words ''. Les textes sont transformés simplement en vecteurs dont chaque composante représente un terme et son poids (*une fonction de l'occurrence des mots dans le texte*).

La figure 2, donne un exemple de représentation d'un texte en sac de mots.

'' On appelle "langage informatique" un langage destiné à décrire l'ensemble des actions consécutives qu'un ordinateur doit exécuter. Les langages naturels (par exemple l'anglais ou le français) représentent l'ensemble des possibilités d'expression partagé par un groupe d'individus. Les langages servant aux ordinateurs à communiquer n'ont rien à voir avec des langages informatiques, on parle dans ce cas de protocoles de communication, ce sont deux notions totalement différentes. Un langage informatique est une façon pratique pour nous (humains) de donner des instructions à un ordinateur.''

Terme	fréquence	Terme	fréquence
Langage	6	Ordinateur	3
informatique	3	Protocole	1
…	…	…	…

Figure 1.2 - Exemple de la représentation d'un texte en « sac de mots »

3.2.2.2. PONDERATION DES TERMES

Le poids d'un terme représente le degré de son importance dans le document.

TF-IDF est la méthode de pondération qui a été la plus étudiée en recherche documentaire, où l'importance d'un terme est proportionnelle à la fréquence d'apparition de ce terme dans le document et inversement proportionnelle à la fréquence en documents (nombre de documents où le terme apparaît) [SAL 83].

- La mesure de Jaccard donne :

$$coeff_{jaccard}\ (X,Y) = \frac{\| X \cap Y \|}{\| X \| + \| Y \| - \| X \cap Y \|} = \frac{\sum x_i * y_i}{\sum x_i^2 + \sum y_j^2 - \sum x_i * y_j}$$

x_i est le poids du terme i dans le document x ; y_i est le poids du terme i dans le document y ;

- Une autre mesure, appelée mesure du cosinus est donnée par la formule suivante :

$$cos\ (X,Y) = \frac{\| X \cap Y \|}{\sqrt{\| X \| * \| Y \|}} = \frac{\sum x_i * y_i}{\sqrt{\sum x_i^2 * \sum y_j^2}}$$

Si le cosinus est proche de 1 alors les éléments sont sémantiquement proches, si par contre le cosinus est éloigné de 1 alors les mots sont sémantiquement différents. Plein d'autres mesures existent dans la littérature telles que la distance euclidienne, Inner Product **[JAC 04],[LAN 97]**.

4. CONCLUSION

Les différentes approches de représentation informatique et de classification du texte ont été exposées dans ce chapitre.

Ainsi avant la codification, un ensemble d'opérations préliminaires doivent être faites pour épurer le texte de tous les mots inutiles et conserver seulement ceux qui sont porteurs d'informations et utiles pour le processus de classification.

Cependant, force est de constater que les prétraitements appliqués sur le courrier, reste peu efficace, vu sa nature déviante par rapport à l'écrit classique.

Ainsi, la deuxième partie de ce chapitre, sera consacrée aux particularités techniques et linguistiques du courrier électronique, et plus particulièrement, aux difficultés qu'elles impliquent pour le traitement et la classification automatique.

III. PARTIE II - COURRIER ELECTRONIQUE

1. INTRODUCTION

Le courrier électronique (*e-mail, messagerie*) est l'une des applications d'internet les plus anciennes : les premières messageries électroniques datent des années 60. Aujourd'hui, l'e-mail est l'application d'internet la plus populaire et la plus répondue au monde.

Les sections suivantes décrivent les principales caractéristiques des langages électroniques qui différent du texte conventionnel.

2. LANGAGE ELECTRONIQUE

L'e-mail appartient à une taxinomie d'outils de communication, qui génèrent une nouvelle forme de langage qu'est le langage *" texto "*. Plusieurs appellations ont été proposées pour représenter ces langages, la plus connue qui englobe les différentes formes d'écrit issues des Technologies de l'Information et de la Communication (TIC) est proposée par *Guimier de Neef et* Veronis **[GUI, 04]**, à savoir *"les Nouvelles Formes de Communication Ecrite "* (NFCE).

NFCE[1] est développé grâce à deux technologies, Internet (les chats, les forums de discussion, l'e-mail, les blogs, etc.) et le téléphone portable (les SMS). Ils ne sont ni ''verbaux'' ni ''écrits'' au sens conventionnel de ces deux termes **[COL, 96].** Il est difficile de dire qu'ils sont verbaux parce que les gens impliques ne se parlent pas de vive voix. D'un autre cote, on ne peut les considérer comme strictement écrits parce qu'ils sont souvent composés sur le champ, sans planification préalable et en ne suivant aucunement les règles de base de la rédaction.

3. PRINCIPALES CARACTERISTIQUES DU COURRIER ELECTRONIQUE

Le courrier électronique a apporté avec lui son lot de casse-tête pour les linguistes. Nous proposons à la suite une analyse de ses principales caractéristiques, issues des travaux effectués à ce jour, d'un point de vue linguistique, mais aussi technique.

[1] http://sites.univ-provence.fr/~veronis/je-nfce/

3.1. ORGANISATION

L'organisation des courriers représente bien le caractère hybride des communications électroniques et la sensation de vitesse qui se dégage de l'informatique. Au lieu de reprendre le même concept d'introduction, de corps et de conclusion typiques des autres documents écrits, les courriers vont directement au but.

3.2. PONCTUATION

L'utilisation de la ponctuation est un autre indice que les courriers sont souvent rédigés différemment des autres documents écrits. En suivant le modèle de l'oral, les phrases sont courtes et simples. Cela a mène à la quasi-disparition de certains signes de ponctuation dont le point-virgule (;), le deux-points (:) et le tiret(-).

3.3. VOCABULAIRE

Dans un courrier, les gens écrivent ce qu'ils diraient verbalement, et de la manière dont ils le diraient.

Cependant ce langage est différent de l'écrit classique, il s'agit de langage ''*abrégé*'' avec une forte tendance de '' *l'oralisassions de la langue''*, qui doit essentiellement son origine à la rapidité de composition du message. Le traitement automatique du langage '' texto'' est une tache titanesque et a peu prés impossible avec des outils standard. Plusieurs travaux se sont penchés sur cette problématique comme ceux de S.VIENNEY **[VIE, 04]**. Dans ce qui suit nous présenterons une typologie des principales particularités linguistiques rencontrées dans la littérature :

3.3.1. NEOGRAPHIE : Consiste à attribuer une nouvelle orthographe pour des mots existants. Par exemple : *pkoi* au lieu de pourquoi, et p-e à la place de peut-être, ou *mdr* veut dire mort de rire.

3.3.2. ERREURS D'ORTHOGRAPHE : Les erreurs d'orthographe sont de deux types :

1. les erreurs de performance
2. les erreurs de compétence

3.3.3. NEOLOGISME : Consiste en l'utilisation de nouveaux mots. Ces néologismes sont généralement des francisations de mots anglophones. A titre d'exemple citons le mot ''*chat''* qui vient du verbe anglais ''*to chat''*,'' *bavarder''* en français.

3.3.4. EMOTICONS : Les émoticons ne font pas partie intégrante du texte, mais sont de petits dessins formes de caractères alphanumériques qui servent à indiquer les émotions de l'émetteur. Voici quelque exemple d'émoticons (figure 3) :

;)	un clin d'œil	: (	la tristesse
:-p	une grimace	:-o	la surprise
| -0	un bâillement	> : (	la colère
:~(	une larme	:-7	le sarcasme

Figure 1.3 : exemple d'émoticon

3.3.5. MARQUES D'EMAILS

Nous regroupons sous ce titre toutes les données que peut contenir un e-mail, et qui ajoutent du bruit risquant de gêner la classification, comme :

- Les adresses d'e-mails, les url, etc.
- La micro publicité ajoutée au bas des e-mails par les fournisseurs de service de messagerie électronique comme illustré ci-après.

  ''Télécharger le nouveau Windows Live Messenger ! Télécharger Messenger, c'est gratuit !''

3.4. SPECIFICATIONS TECHNIQUES

L'évolution du courrier électronique est gérée par l'Internet Mail Consortium (IMC)[2] et par l'Internet Engineering Task Force(IETF)[3]. L'IMC est une organisation international vouée au développement, à la promotion et à la facilite d'utilisation du courrier électronique.

L'IETF est une organisation mondiale regroupant tous les gens concernés par l'évolution et le bon fonctionnement de l'Internet, comme les chercheurs, les programmeurs, les entreprises, etc. ils se focalisent sur la présentation de la structure de base d'un courrier **[PAL, 97], [RES,01].**

3.4.1. EN TETE : L'entête contient les informations reliées au transport du courrier : la date d'envoi, l'émetteur, le ou les récepteurs, les adresses IP rencontrées lors du trajet, etc. elle est

[2] http://www.imc.org/
[3] http://www.ietf.org/

constituée de plusieurs champs ayant tous la même syntaxe de base, Sauf l'information indiquée dans « Objet », car elle peut nous informer sur le contenu de l'e-mail.

La figure 1.4 illustre un exemple d'en-tête

De : chahnez.zakaria@yahoo.fr
A: nabila.bousbia@yahoo.fr
Cc: seffadj@yahoo.fr
Objet : CURSUS
Date : Sun, 13Jan 2008 13 :36 :36+0000
Size : 3360 Bytes
XPriority : 3

Figure 1.4 : Exemple d'entête d'un e-mail

3.4.2. CORPS : Celui-ci est seulement une suite de caractères formant le message transmis.

3.4.3. FORMAT D'ENCODAGE DE MESSAGERIE : MIME (Multipurpose Internet Mail Extensions) **([FRE, 96])** est un standard qui a été proposé par les laboratoires *Bell Communications* en 1991 afin d'étendre les possibilités limitées du courrier électronique et notamment de permettre d'insérer des documents (images, sons, texte,...) dans un courrier. Il est défini à l' origine par les RFC 1341 et 1342 datant de juin 1992.

La table1 présente quelques exemples des types MIME.

Type MIME	Type de fichier	Extension associée
Application/msword	Fichiers bureautique au format Microsoft Word	Doc
Application/pdf	Fichiers Adobe Acrobat	Pdf

Table 1 : Exemple de type MIME

3.4.4. SERVEUR DE MESSAGERIE : Un serveur de messagerie est un ordinateur-serveur dédie (c.-à-d. une machine et des logiciels, offrant des services aux utilisateurs). Ce serveur de messagerie offre des espaces disques pour le stockage des comptes de messagerie personnels et assure la transmission et la réception des messages.

3.4.5. ADRESSE ELECTRONIQUE : Le courrier électronique repose sur les adresses électroniques des utilisateurs : adresse d'une personne, possédant une boite aux lettres électroniques. Les adresses de courrier électronique se présentent toujours de la manière suivante :

''Nom de l'utilisateur@organisation.domaine''

Le signe @ (arobase) signifie at (chez), pour indiquer la notion d'hébergement sur un serveur hôte.

3.4.6. PROTOCOLES : Deux grands types de protocoles et de serveurs sont utilisés pour le courrier électronique :

- **Les protocoles « sortants »,** permettant de gérer la transmission du courrier entre les serveurs. Le principal protocole sortant est SMTP (Simple Mail Transfer Protocol, ou Protocole Simple de Transfert de courrier) est un protocole de communication pour le courrier permettant d'établir l'interface entre un réseau local et internet.

- **Les protocoles « entrants »,** gérer la communication entre l'utilisateur et le serveur de messagerie et permettre aux utilisateurs d'aller récupérer leurs messages.
  Deux protocoles entrants sont utilisés, au choix, dans les systèmes de messagerie : POP ou IMAP, Ce sont des protocoles de réception et de distribution du courrier.

4. CONCLUSION

Dans cette partie, nous avons exposé les principales caractéristiques du courrier électronique, qui se définit comme un outil de communication universelle, qui à engendré des nouvelles pratiques et modalité écrites codifiées a typiquement par opposition à l'usage dit ''standard''.

Par ailleurs, nous avons pu conclure de cette étude, que le courrier électronique constitue un défi, qui remet en cause les principes algorithmiques généralement acceptés dans le domaine.

Le but du chapitre suivant, n'est évidemment pas d'esquisser des solutions au traitement NFCE, mais plutôt de proposer une architecture globale, permettant de combiner les techniques et outils du traitement automatique des NFCE, du traitement automatique du texte et ceux de la classification automatique du texte.

CHAPITRE II :

ANALYSE DES BESOINS ET CONCEPTION DE L'APPLICATION

CHAPITRE II : ANALYSE DES BESOINS ET CONCEPTION DE L'APPLICATION

I. INTRODUCTION

Nous exposons dans cette partie du mémoire, notre travail de développement d'une application, ayant la capacité de transformer le contenu d'un courriel en un texte (texto vers texte) pour rendre possible l'application des différents traitements automatique du texte et par la suite la classification de ces courriels.

Une estimation de la taille et la durée d'accomplissement du projet, nous a mené à choisir un cycle de développement en cascade, qui nous semblait adapté à un projet de cette dimension. Ce dernier commencent par une phase « d'analyse et de spécification des exigences » suivit par une phase de conception repartie sur deux sous phases « conception générale » et «conception détaillé », suivie d'une phase « d'implémentation » et enfin d'une phase « d'intégration » et « de mise en production », chacune de ces phases fournit un produit en sortie qui sera utilisé dans la phase qui la succède.

La figure 2.1 montre l'architecture du cycle de développement adopté.

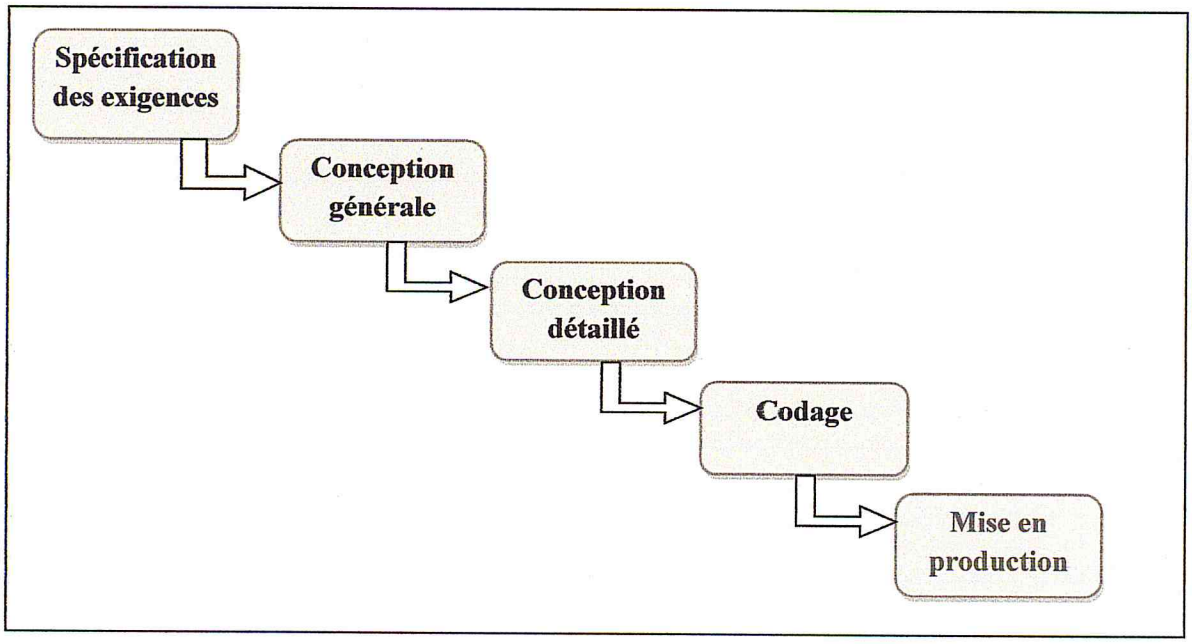


Figure 2.1 : Cycle de vie adopté pour le développement de l'application

Ce chapitre expose les phases spécification des exigences, conception générale et détaillée et décrit les objectifs de notre système, il présente ensuite l'architecture du système proposé avec ses différentes fonctionnalités et ses différents modules.

II. ANALYSE ET SPECIFICATION DES EXIGENCES

1. ANALYSE DES BESOINS

- **BESOIN 1 :** Un analyseur d'email, qui offre une représentation normalisé des courriels reçu sous forme NFCE, afin d'être exploitable par les algorithmes existants (*traitement du texte, classification*).
- **BESOIN 2 :** Une interface graphique pour le paramétrage du processus d'analyse et de classification (de bout en bout), offrant ainsi une grande flexibilité.
- **BESOIN 3 :** Un gestionnaire de corpus, qui offre la possibilité de manipuler des outils pour la gestion, la recherche et la classification des courriels.
- **BESOIN 4 :** Un outil pour l'évaluation et la simulation des résultats de classification. Ce module est principalement dédié à des projets de recherche.

2. ARCHITECTURE DU SYSTEME

L'objectif de notre système est d'assurer l'analyse et la classification automatique des courriers électroniques. Pour atteindre cet objectif, le système se décompose principalement, en trois modules, illustrés dans la figure.2.2. :

1- Analyse
2- Prétraitement
3- Classification.
4- Gestion du corpus

Nous allons détailler dans les parties qui suivent, le rôle des différents modules du système, nous proposons de modéliser leurs fonctionnements ainsi que les fonctionnalités qui doivent être fournies à l'utilisateur grâce aux diagrammes UML.

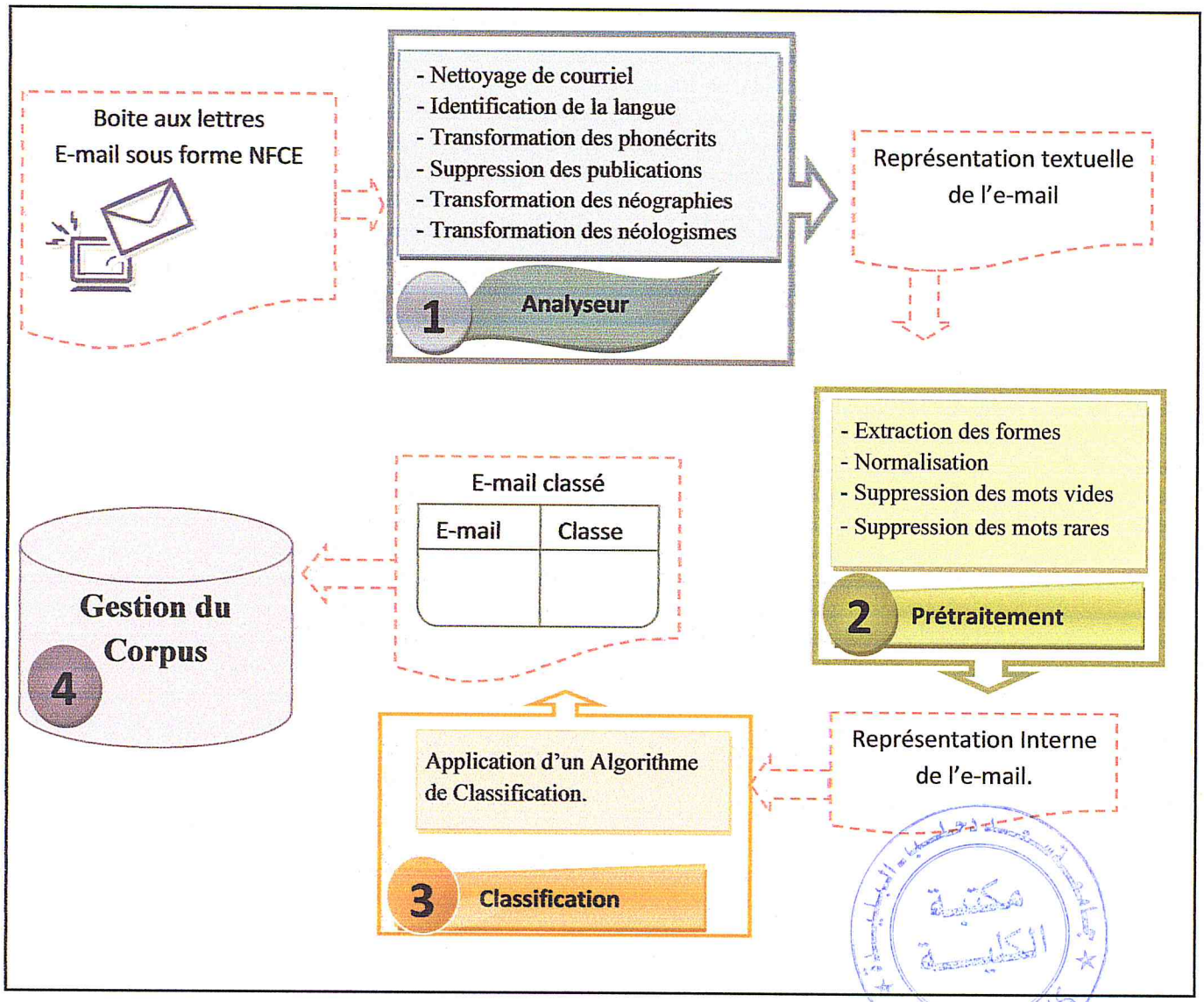


Figure 2.2 : Architecture globale du système

2.1. ANALYSEUR D'EMAIL

En premier, un module de l'analyse de courriel est lancé pour préparer les messages récupérés de la boîte de courriel, aux différentes étapes ultérieures de l'analyse. Il consiste à :

1- Isoler les différents champs (entête, expéditeur, objet, corps, pièces jointes ….).
2- Identifier la langue de chaque message parmi deux actuellement modélisées (français, anglais).
3- Suppression des publications.
4- Transformation des émoticônes.
5- Transformation des néographies.
6- Transformation des néologismes.

Le schéma dans la figure.2.3 résume le fonctionnement de ce module.

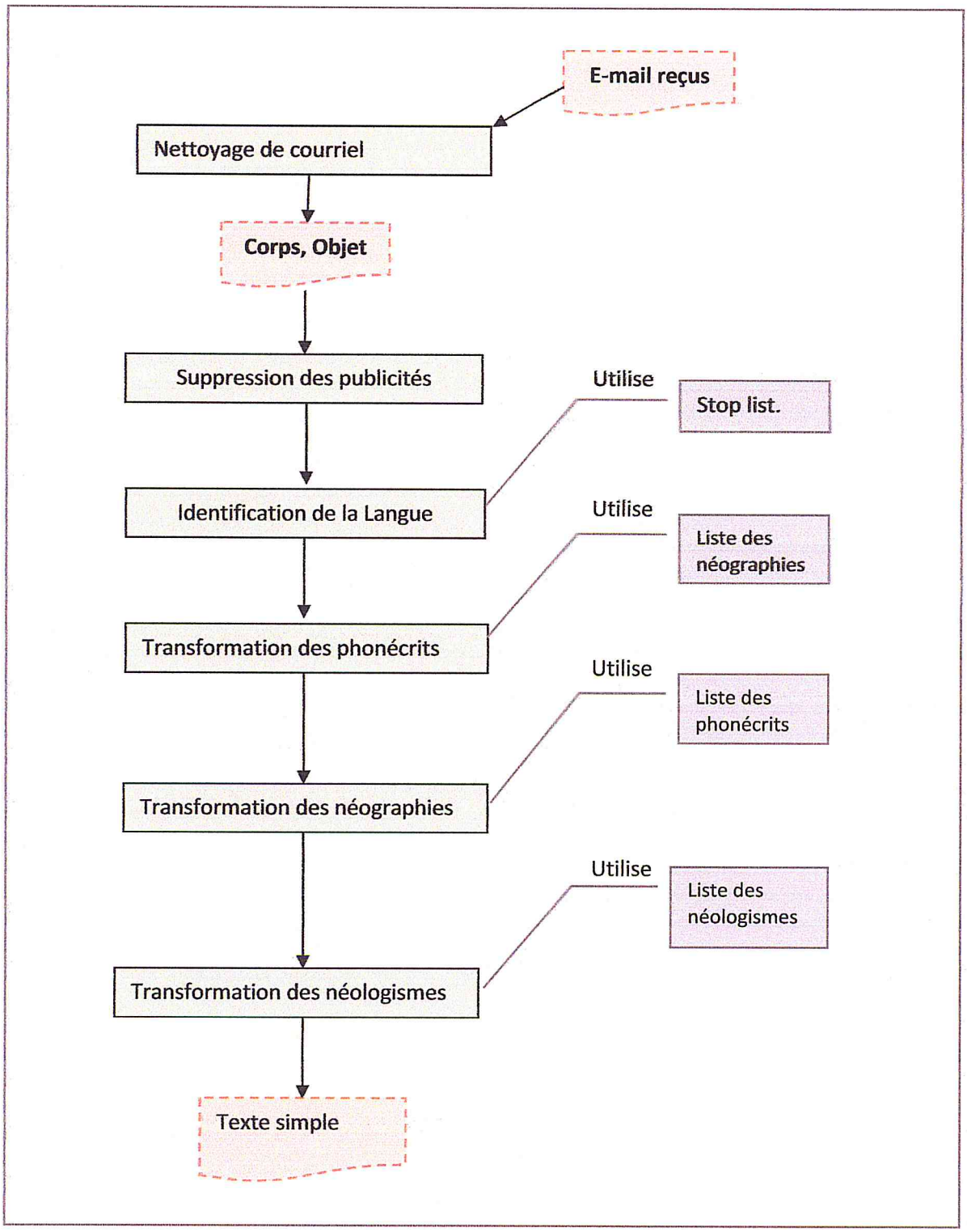


Figure 2.3 : Architecture de l'analyseur

2.1.1. NETTOYAGE DE MAIL

La première partie de l'analyseur, sépare le corps, l'entête et les pièces jointes (cf. figure 2.4). Cette première étape génère un fichier XML, contenant les informations récupérées du message.

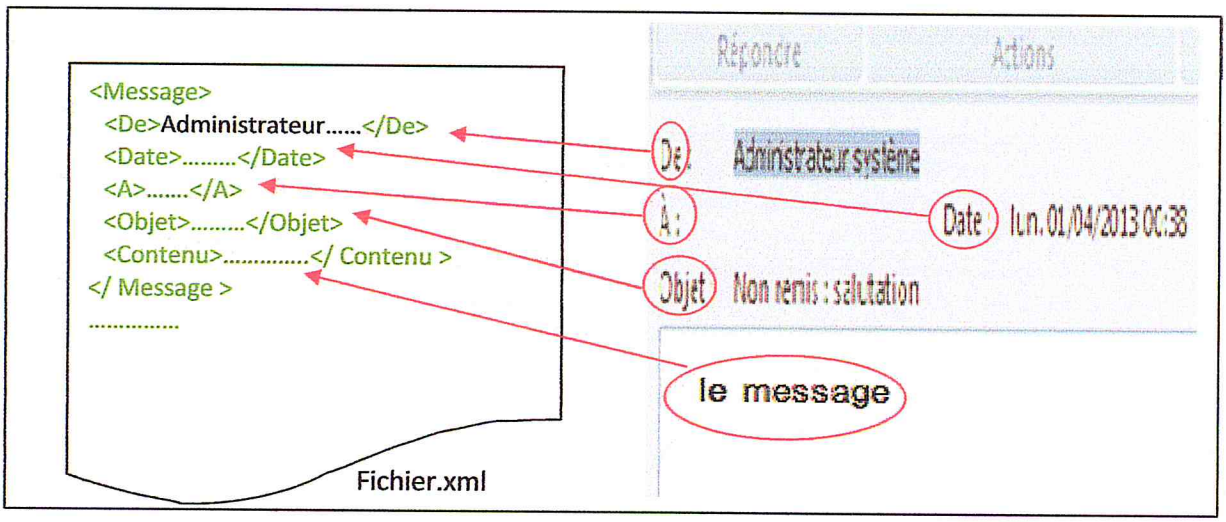


Figure 2.4 : Exemple de nettoyage d'un message

2.1.2. SUPPRESSION DES PUBLICITES

La suppression des micros publicité (microspams) qui n'apporte aucune information permettant de catégoriser le courriel mais, au contraire, ajoute du bruit risquant de gêner cette catégorisation. Il s'agit en général de publicités ajoutées au bas des courriels par les fournisseurs de service de messagerie électronique comme le montre l'exemple suivant dans figure 2.5 :

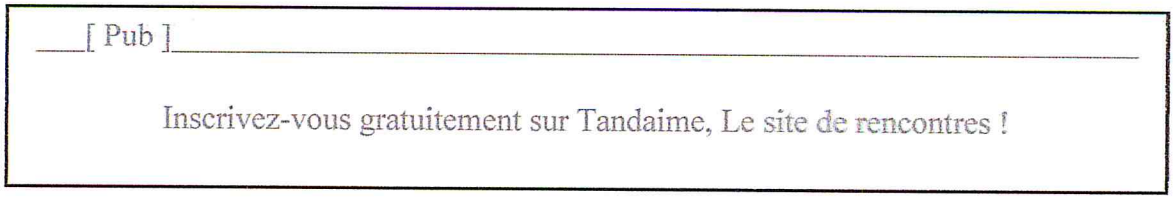


Figure 2.5 : Exemples de publicités

Nous avons supprimé la publicité générique des courriels, celle-ci étant généralement précédée d'une ligne composée de la façon suivante `____[ Pub ]_____`, ou encore **********`. La particularité de ces lignes a permis de les enlever sans risque de perte d'informations au niveau du corps du message.

2.1.3. IDENTIFICATION DE LA LANGUE

La méthode d'identification de la langue est simple : elle utilise des anti-dictionnaires (ou stoplist) propres à chaque langue. Il s'agit de compter, pour chaque message, le nombre de mots outils (articles, prépositions…).Par ailleurs, le système est incrémental et permet facilement la prise en compte de nouvelles langues (ajouter un anti-dictionnaire propre à chaque nouvelle langue).

2.1.4. TRANSFORMATION DE L'EMOTECONE

À l'aide d'un dictionnaire constitué à partir de sites[4] et décrivant les divers émoticônes, nous remplaçons ceux-ci par leurs équivalents en langue. Cette étape de "traduction" est réalisée avant la suppression de la ponctuation car les émoticônes sont essentiellement composés à l'aide de ponctuation (:) → sourire, :) triste).

2.1.5. TRANSFORMATION DE LA NEOGRAPHIE

C'est un processus qui se charge d'effectuer les traitements de transformation des néographies. À l'aide d'un dictionnaire constitué à partir de [REB, 10] et décrivant les divers termes de néographie, par exemple mdr → mort de rire, pkoi → pour quoi.

2.1.6. TRANSFORMATION DE LA NEOLOGISMES

À l'aide d'un dictionnaire constitué à partir de sites[5] et décrivant les divers termes de néologismes, nous avons pu transformer certain terme parmi les plus fréquents.

Par exemple : Chatter →parler, facebouker → utiliser facebook.

2.2. MODULE DE PRETRAITEMENT

Après le nettoyage et la transformation de texto vers texte, le module de prétraitement est lancé pour transformer le message reçu en un ensemble de concepts en passant par les tâches principales suivantes :

- Extraction simple des termes.
- Suppression des mots vides.
- Suppression des mots rares.
- Normalisation.

[4] www.codes-des-emoticones.com
[5] www.neologisme-wikipedia.com

Le schéma suivant dans figure.2.6 résume le fonctionnement de ce module :

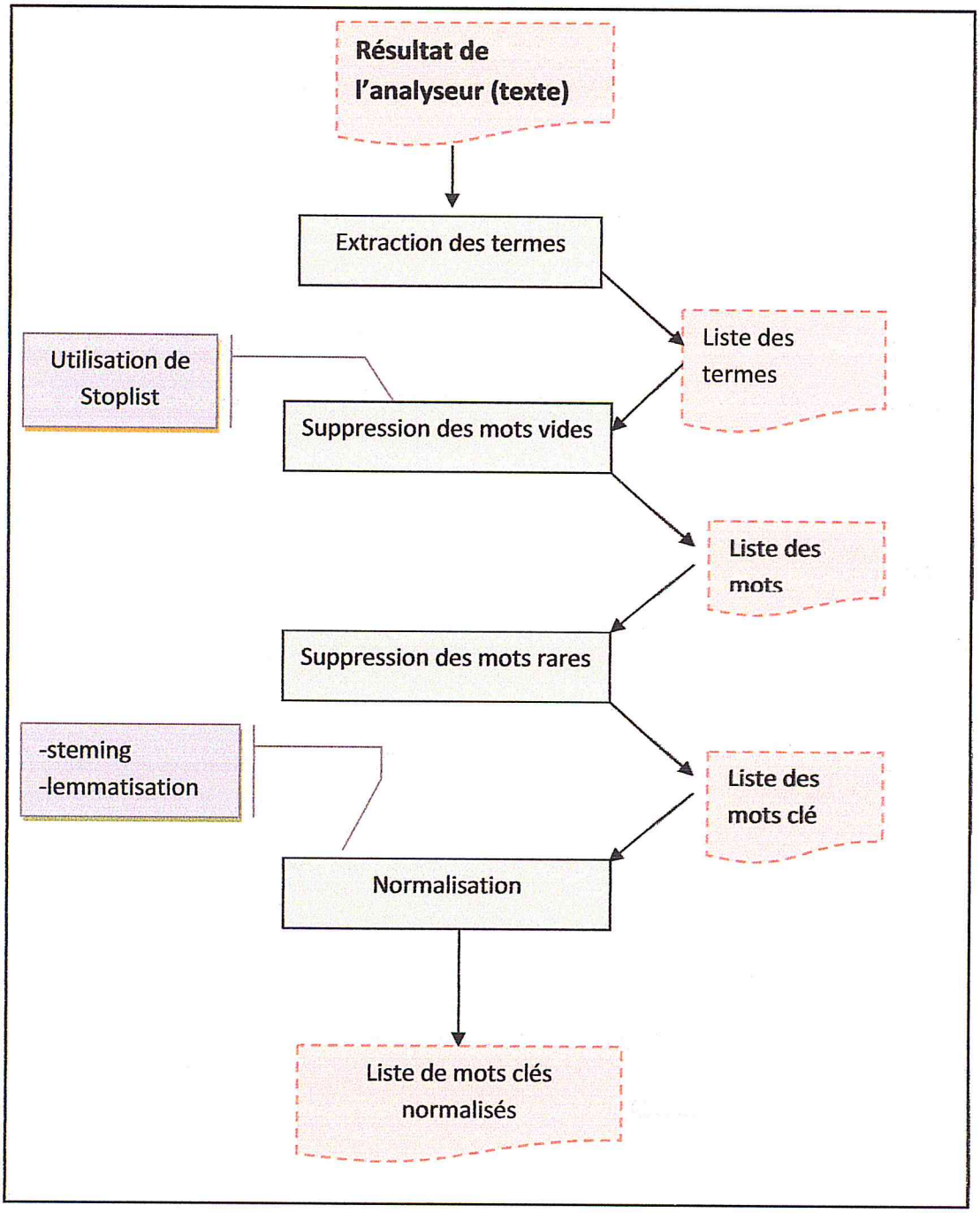


Figure 2.6 : Architecture du module de prétraitement

2.2.1. EXTRACTION SIMPLE

Consiste à extraire la liste des termes simples à partir d'un texte, qui se base sur les délimiteurs, ces derniers changent d'une langue à une autre.

2.2.2. SUPPRESSION DES MOTS VIDES

Les mots vides (ou outils) font partie des mots non porteurs de sens, qu'il faut filtrer. Ils comprennent des mots de différents types grammaticaux, à savoir : article, pronom, préposition, etc. [ZAK, 09].

La suppression des mots vides revient à supprimer les mots qui n'ont pas d'importance informationnelle, autrement dit, ils n'ont pas un poids sémantique. Les mots vides différents d'une langue à une autre, la liste des mots vides à charger est choisie grâce à la langue du texte.

La figure 2.7 donne un exemple de liste des « stops words[6] » anglais utilisée par les moteurs de recherche, en particulier le moteur de recherche Google.

I, a, about, an, are, as, at, be, by, com, de, en, for, from, how, in, is, it, la, of, on, or, that, the, this, to, was, what, when, where, who, will, with, und, the, www.

Figure 2.7 : La liste des stops words anglais

La liste suivante (figure 2.8) présente ceux de la langue française[7] :

Alors, au, aucuns, aussi, autre, avant, avec, avoir, bon, car, ce, cela, ces, ceux, chaque, ci, comme, comment, dans, des, du, dedans, dehors, depuis, deux, devrait, doit, donc,

dos, droite, début, elle, elles, en, encore, essai, est, et, eu, fait, faites, fois, font, force , haut, hors, ici, il, ils, je, juste, la, le, les, leur, là, ma, maintenant, mais, mes, mine, moins, mon, mot, même, ni, nommés, notre, Nous, nouveaux, ou, où, par, parce, parole, pas, personnes,

peut, peu, pièce, plupart, pour, pour, quoi, quand, que, quel, quelle, quelles, quels, qui, sa, sans, ses, seulement, si, sien, son, sont, sous, soyez, sujet, sur, ta, tandis, tellement, tels, tes,

Figure 2.8: la liste des stops words Français

Après l'extraction des termes, une vérification des stops words est lancée pour la suppression des mots vides.

[6] http://www.okkiweb.fr/?Liste-des-stop-words-mots-vides
[7] http://www.ranks.nl/stopwords/french.html

2.2.3. SUPPRESSION DES MOTS RARES

Les mots rares sont considérés comme des mots non porteurs de sens, du temps qu'ils ne sont employés que très rarement dans le corpus.

Ainsi, tous les mots dont la fréquence d'apparition dans le corpus est inférieure à un certain seuil, sont enlevés.

2.2.4. NORMALISATION

L'objectif de la normalisation est de regrouper les termes dans des familles pour avoir une pondération plus précise, il est évident que, à titre d'exemple, « former » et « formation » appartiennent à la même famille, présentée par le radical «form». Un steming doit donc être effectué.

Plusieurs normalisations existent : steming, lemmatisation, etc.

- Les algorithmes de stemming sont très nombreux, mais l'un d'entre eux est le plus connu et le plus utilisé, c'est l'algorithme de Porter **[POR, 80]**.
- Lemmatisation (définition des formes canoniques) consiste à utiliser l'analyse grammaticale pour transformer chaque terme en sa forme canonique. Il s'agit donc de remplacer les verbes par leur forme infinitive, les noms par leur forme au singulier, et transformer au masculin tous les mots qui sont au féminin.

Chaque langue a ses propres règles de lemmatisation qui suivent les règles de grammaire et de syntaxe. Dans ce travail, nous nous basons sur deux langues : l'anglais et le français.

La lemmatisation nécessite l'utilisation d'algorithmes très compliqués avec des fichiers d'informations selon la langue, pour cela nous avons utilisé un algorithme très efficace, nommé TreeTagger[8] [STR, 00], qui a été développé pour les langues anglaise, française, allemande et italienne. Cet algorithme utilise des arbres de décision pour effectuer l'analyse grammaticale, puis des fichiers de paramètres spécifiques à chaque langue. Nous avons utilisé ce lemmatiseur au cours de cette étape. La figure 2.9 présente un exemple d'une phrase lemmatisée avec TreeTagger.

[8] Les publications relatives à cet algorithme ainsi que les codes source sont disponibles sur le site : http://www.ims.uni-stuttgart.de/projekte/corplex/DecisionTreeTagger.html

```
TreeTagger          NAM          <unknown>
permet              VER:pres     permettre
d'                  PRP          de
annoter             VER:infi     annoter
plusieurs           PRO:IND      plusieurs
langues             NOM          langue
.                   SENT         .
```

Avec
ABR : abréviation
ADJ : adjectif
VER : verbe
DET:ART :article

Figure 2.9: Exemple d'une phrase lemmatisé avec TreeTagger.

Le traitement porte sur une liste de termes. Le steming traite les termes l'un après l'autre, par contre la lemmatisation prend toute une liste de termes, l'appel de TreeTagger se fait via ligne de commande (exécution des commande système).

2.2.5. REPRESENTATION DE COURRIEL

En sortie du module de prétraitement, un courriel est représenté par une structure exploitable par le système.

Nous avons implémenté dans la version actuelle du système la structure vectorielle « Sac de mots », toutefois le système offre la possibilité d'ajout d'autre représentation.

Le courriel est représenté conceptuellement par un vecteur :

$$M = \{(T1, W1)\}.$$

T1 représente la valeur (le terme), W1 le poids.

Cette représentation constitue l'entrée du module de classification.

2.3. MODULE DE CLASSIFICATION

La classification de texte est généralement associée à la séparation des documents selon leur contenu. La figure 2.10 résume le fonctionnement de ce module.

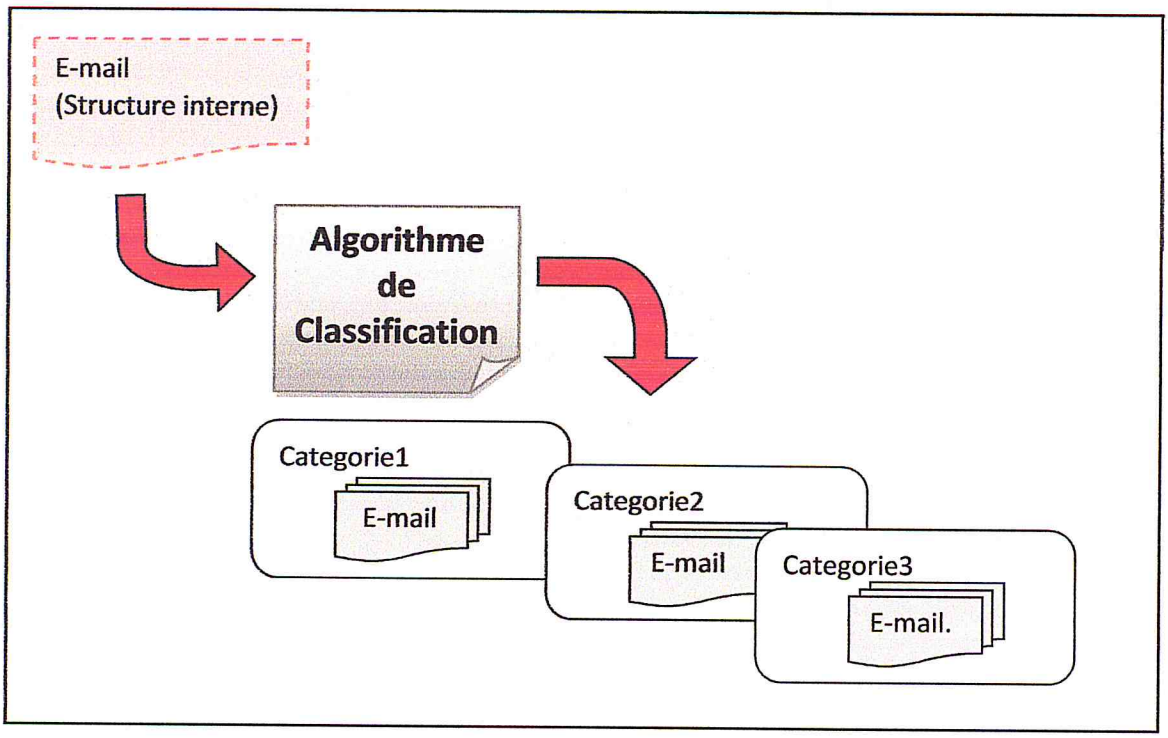


Figure 2.10 : Architecture du module de classification

La classification principale sera faite en fonction du contenu du message, le système permet de faire un choix de l'algorithme de classification à appliquer.

Dans ce travail, par contrainte de temps, nous avons implémenté une méthode de classification classique l'algorithme KPPV :

2.3.1. PRINCIPE DE L'ALGORITHME KPPV

C'est un algorithme qui permet la classification supervisée, son but est de déterminer la classe pour le nouvel individu en entrée, en utilisant les individus déjà classés.

2.3.2. DESCRIPTION DE L'ALGORITHME KPPV

- Choisir un entier K.
- Calculer des distances.
- Retenir le k observations pour lesquelles ces distances sont le plus petite.
- Compter le nombre de fois ou ces k observations appariassent dans chacune des classes.
- Choisir la classe la plus représente.

2.4. MODULE DE GESTION DE CORPUS

Ce module donne la possibilité de choisir un corpus de travail, de plus, il offre une panoplie d'outils qui permettent essentiellement :

- L'analyse, la simulation et l'évaluation des résultats.
- La validation des démarches suivies.

III. SPECIFICATION SEMI-FORMELLE DES BESOINS

Une étude approfondie des besoins fonctionnels s'avère indispensable avant d'entamer la conception, afin d'obtenir de manière plus formelle une vue globale sur les exigences de l'application. Cette partie présente alors une modélisation de ces besoins en faisant recours aux concepts fondamentaux du langage de la référence UML.

1. MODELISATION PAR UNE METHODE CONCEPTUELLE

1.1. DEFINITION D'UML

UML (*Unified Modeling Language*) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement.

1.2. AVANTAGES D'UML

✓ UML est un langage formel et normalisé : il permet un gain de précision et de stabilité.
✓ UML est un support de communication performant : il permet grâce à sa représentation graphique, d'exprimer visuellement une solution objet, de faciliter la comparaison et l'évolution de solution.
✓ Son caractère polyvalent et sa souplesse en font un langage universel.

1.3. DIAGRAMMES D'UML

Un diagramme UML est une représentation graphique, qui permet de modéliser un aspect bien précis du système, chaque type de diagramme UML possède une structure et des concepts prédéfinis.

Un diagramme donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation.

1.3.1. DIAGRAMME DE CAS D'UTILISATION (USE CASES)

Représente les fonctions du système du point de vue des utilisateurs.

- Il s'agit de la solution UML pour représenter le modèle conceptuel.
- Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.
- Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés.
- Ils identifient les utilisateurs du système (acteurs) et leur interaction avec le système.
- Ils servent de base à la traçabilité des exigences d'un système dans un processus de développement intégrant UML.

La figure 2.11 schématise le diagramme de cas d'utilisation relatif à notre système.

Les différents cas d'utilisations exprimés dans la figure 2.11, sont détaillés dans la section suivante (du tableau 2.1 au tableau 2.12) :

Le cas d'utilisation « *gestion d'analyse*» et ses extensions sont décrits dans le tableau 2.1

Item	**Description**
Nom	Gérer l'analyse
Description	Grace à ce cas d'utilisation et ses extensions, l'utilisateur peut effectuer divers types de traitements concernant la gestion d'analyse du message.
Etend	/

Tableau 2.1- Description du cas d'utilisation « Gérer l'analyse».

Le cas d'utilisation « *gestion de prétraitement* » est décrit dans le tableau 2.2

Item	**Description**
Nom	Gestion de prétraitement
Description	Grace à ce cas d'utilisation et ses extensions, l'utilisateur peut effectuer divers types de traitements concernant la gestion de prétraitement du message.
Etend	/

Tableau 2.2- Description du cas d'utilisation « Gestion de prétraitement ».

Le tableau 2.3 décrit le cas d'utilisation « *gestion de classification* » et ses extensions.

Item	**Description**
Nom	Gestion de classification
Description	Grace à ce cas d'utilisation et ses extensions, l'utilisateur peut gérer la classification, il peut choisir un algorithme de classification parmi les divers algorithmes existants.
Etend	« Choisir l'algorithme »

Tableau 2.3- Description du cas d'utilisation « Gestion de classification ».

Le cas d'utilisation « *consulter l'aide* » et ses extensions sont décrits dans le tableau 2.4

Item	**Description**
Nom	Consulter l'aide
Description	Grace à ce cas d'utilisation et ses extensions, l'utilisateur peut consulter divers options d'aide, à savoir une rubrique d'aide qui lui donnera des informations sur les fonctionnalités de l'application ou des informations sur l'application elle même.
Etend	/

Tableau 2.4- Description du cas d'utilisation « Consulter l'aide ».

Le cas d'utilisation « *choisir la langue*» et ses extensions sont décrits dans le tableau 2.5

Item	**Description**
Nom	Choisir la langue
Description	L'administrateur choisi la langue de la gestion d'analyse et de prétraitement des courriels.
Etend	/

Tableau 2.5- Description du cas d'utilisation « Choisir la langue ».

Le cas d'utilisation « *gestion de corpus*» et ses extensions sont décrits dans le tableau 2.6

Item	**Description**
Nom	Gérer le corpus
Description	Grace à ce cas d'utilisation, l'utilisateur peut consulter les déférents messages existants, mettre à jour le corpus et effectuer des évaluations et des analyses sur les démarches suivies.
Etend	/

Tableau 2.6- Description du cas d'utilisation « Gérer le corpus ».

Le cas d'utilisation « *M-à-j des mots vides*» et ses extensions sont décrits dans le tableau 2.7

Item	**Description**
Nom	Mise à jour des mots vides
Description	Grâce à ce cas d'utilisation l'administrateur peut gérer le prétraitement. Il peut consulter, ajouter, modifier ou supprimer les mots vides.
Etend	<ajouter>, <supprimer >, <modifier>,<consulter >.

Tableau 2.7- Description du cas d'utilisation « M-à-j des mots vides».

Le cas d'utilisation« *M-à-j des mots rares*» et ses extensions sont décrits dans le tableau 2.8

Item	**Description**
Nom	M-à-j des mots rares
Description	Grâce à ce cas d'utilisation l'administrateur peut gérer le prétraitement. Il peut consulter, ajouter, modifier ou supprimer les mots rares.
Etend	<ajouter>, <supprimer >, <modifier>, <consulter >.

Tableau 2.8- Description du cas d'utilisation « M-à-j des mots rares».

Le cas d'utilisation« *M-à-j néographies*» et ses extensions sont décrits dans le tableau 2.9

Item	**Description**
Nom	M-à-j néographies
Description	Grâce à ce cas d'utilisation l'administrateur peut consulter, ajouter, modifier ou supprimer des néographies.
Etend	<ajouter>, <supprimer >, <modifier>, <consulter >.

Tableau 2.9- Description du cas d'utilisation « M-à-j néographies».

Le cas d'utilisation« *M-à-j des néologismes*» et ses extensions sont décrits dans le tableau 2.10

Item	**Description**
Nom	M-à-j des néologismes
Description	Grâce à ce cas d'utilisation l'administrateur peut consulter, ajouter, modifier ou supprimer des néologismes.
Etend	<ajouter>, <supprimer >, <modifier>, <consulter >.

Tableau 2.10- Description du cas d'utilisation « néologismes».

Le cas d'utilisation« *M-à-j des émoticônes* » et ses extensions sont décrits dans le tableau 2.11

Item	**Description**
Nom	M-à-j des émoticônes
Description	Grâce à ce cas d'utilisation l'administrateur peut consulter, ajouter, modifier ou supprimer des émoticônes.
Etend	<ajouter>, <supprimer >, <modifier>, <consulter >.

Tableau 2.11- Description du cas d'utilisation « émoticônes».

Le cas d'utilisation« *Validation* » et ses extensions sont décrits dans le tableau 2.12

Item	**Description**
Nom	Validation
Description	Grâce à ce cas d'utilisation l'administrateur peut consulter l'évolution de traitement de son message. Qui passera par l'analyse et le prétraitement et en finalité qui sera classifie. Il peut aussi effectuer divers type d'analyse, de simulation et de validation du processus de classification.
Etend	<analyse>, <prétraitement >, <classification >.

Tableau 2.12- Description du cas d'utilisation « validation».

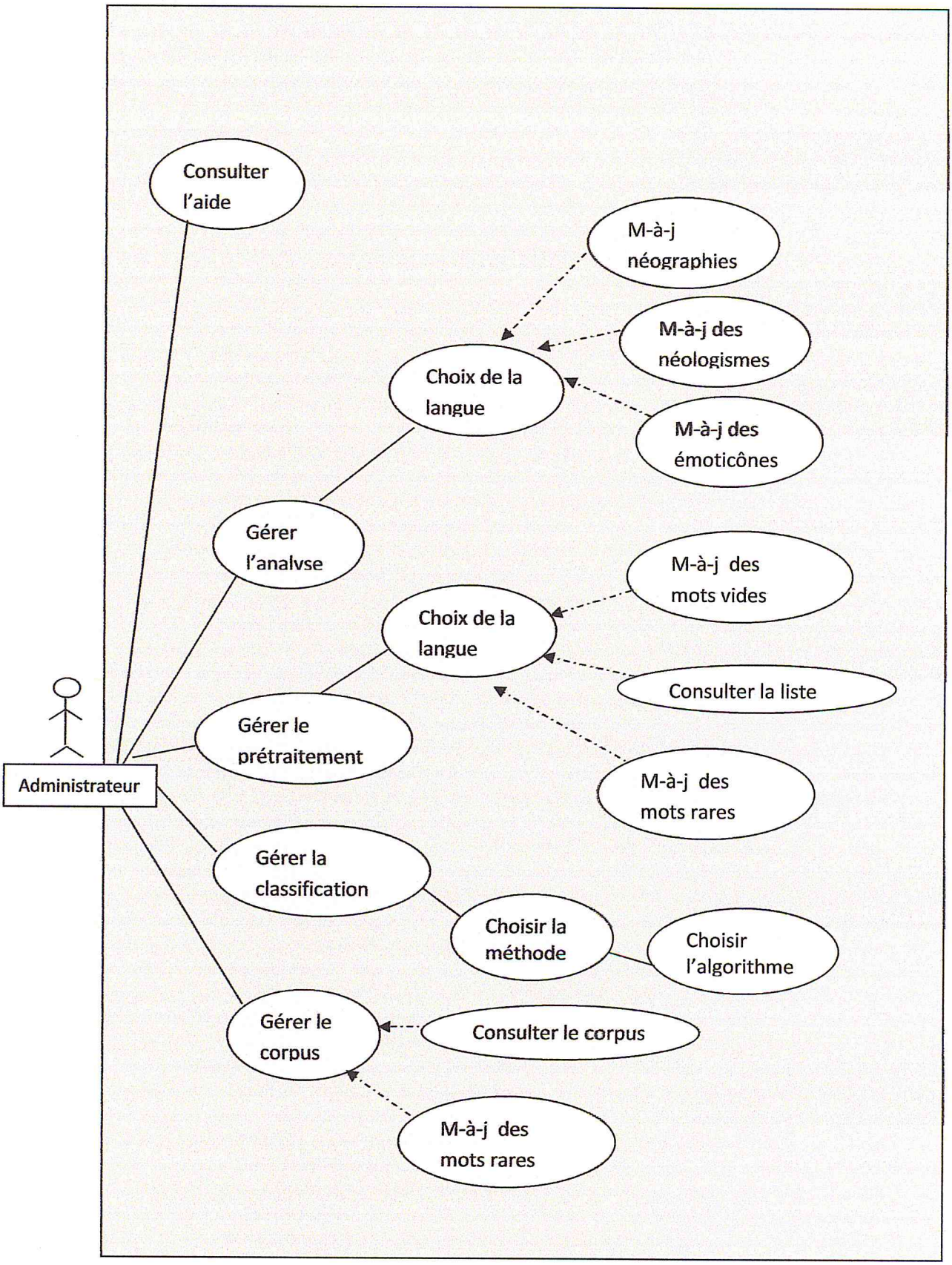


Figure 2.11 : diagramme de cas d'utilisation globale

IV. CONCEPTION GENERALE

Il s'agit dans cette phase d'apporter des solutions conceptuelles aux besoins exprimés dans le cahier de charges élaboré ci-dessus, pour cela nous allons épuiser dans les outils offerts par UML afin d'apporter une vision modulaire de notre future application.

1. DIAGRAMME DE CLASSES

Un diagramme de montre une collection d'éléments statiques (classes), leur contenu (attributs, opérations, types) et les relations entre eux (associations). Il permet de décrire la structure statique d'un système. Néanmoins, on constate souvent qu'un diagramme de classes proprement réalisé permet de structurer le travail de développement de manière très efficace ; il permet aussi, dans le cas de travaux réalises en groupe, de séparer les composantes de manière à pouvoir répartir le travail de développement entre les membres des groupe. Enfin, il permet de construire le système de manière correcte.

1.1. DESCRIPTION DES CLASSES

Avant d'exposer notre diagramme de classe nous allons décrire dans cette partie (cf. tableau 2.13) les classes objets constituant notre application.

PACKAGE	CLASSE	DESCRIPTION
ANALYSE	Analyse	- Classe abstraite. - Permet à l'administrateur de paramétrer la méthode d'Analyse.
	ParamAnalyse	- Classe abstraite. - Contient des méthodes qui permettent de vérifier si le terme est un mot néologisme, néographie ou un émoticon. Ainsi que leur mise à jour.
	Neologisme	- Représente un objet. - Contient les néologies.
	Neographie	-Représente un objet. -contient les néographies.
	Emoticone	-Représente un objet. -Contient des informations sur :{Valeur=l'emoticon ;Syno=son synonyme }

	Publicite	-Représente un objet. -permet à l'administrateur de vérifier l'existence de la Pub.
PRETRAITEMENT	**Pretraitment**	- Classe abstraite. - Permet à l'administrateur de paramétrer la méthode de Prétraitement.
	ParamPretrait	- Classe abstraite. -Permet au Use de vérifier si le mot est un mot vide (StopListe) ou mot rare. -contient des méthodes permettant de mettre à jour les mot vide et StopListe.
	StopListe	- Représente un objet. -Contient la liste des mots Vides.
	MotRare	- Représente un objet. -contient la liste des mots rares.
	Stemer	- C'et un composant de la classe prétraitement. -Permet au Use de consulter le radicale de chaque terme.
	MesOutils	-C'est un composant de la classe traitement. -Contient des méthodes qui permettent de faire passer un texte dans un String(ou l'inverse). -Administrateur l'utilise pour la lemmatisation.
CLASSIFICATION	**Classification**	- classe abstraite. -Permet à l'administrateur de paramétrer la méthode de classification.
	Kpp	-Un Composant de la classe ParamClass.
	Distance	- Représente un objet. -Un composant de la classe ParamClass. -permet au Use de utiliser les déférentes méthodes de calcule de distance.
	DistanceEuc	Représente un objet. -Permet de calculer la Distance Euclidienne pour les termes, lemmes et stemmes.
	DistanceMan	- Représente un objet. -Permet de calculer la Distance de Manhattan pour les termes, lemmes et stemmes.

CORPUS	Corpus	-Représente un document. -contient l'ensemble de messages reçus. -Use peut consulter les messages. Lancer le traitement et aussi mettre à jours les mots rares.
	Categorie	-Représente un Objet. -Permet au Use de classifier les messages. -c'est un composant de la Classe Corpus.
	Message	-Représente le contenu de l'e-mail. -Use peut consulter des informations sur le message (l'entête, Corp.,).
	Term	-Représente un Objet. -Contient un ensemble d'information sur le terme {valeur, nature, lemme, stem, poid} - permet au use de consulter les termes. -c'est un composant de la classe Message.

Tableau 2.13- Description des class

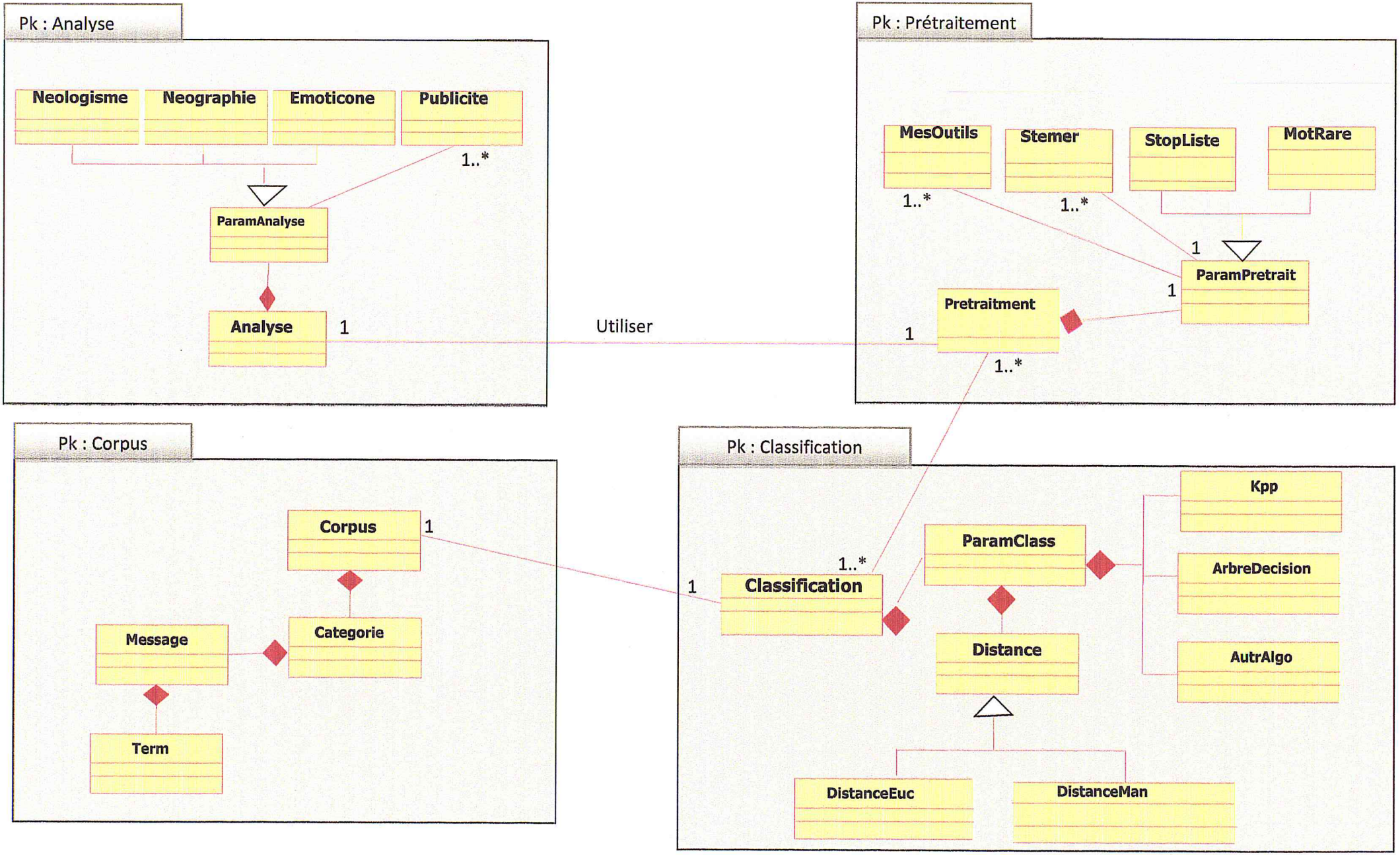


Figure 2.11 : Diagramme de classe globale

V. CONCLUSION

Dans ce chapitre, nous avons abordé l'architecture générale de notre système projeté, ainsi que la description et le fonctionnement des différents modules. Nous avons exposé dans ce chapitre notre travail d'analyse et de conception de l'application, en faisant appel aux divers outils de modélisation UML.

Le prochain chapitre est réservé au reste des phases du cycle de vie de développement.

CHAPITRE III :

MISE EN OUVRE

CHAPITRE III : MISE EN OUVRE

I. INTRODUCTION

Nous avons vu dans la partie précédente la conception détaillée ainsi que la description de ses différents modules. Dans ce chapitre, nous allons présenter les différents aspects techniques liés à l'implémentation et au déploiement de notre système, en suite nous présentons notre application baptisé ''ClassEmail''.

I. OUTILS DE DEVELOPPEMENT

Dans cette partie, on va citer les outils de développement pour l'implémentation de prototype tout en mentionnant les raisons qui nous ont amenés à les utiliser.

1. LANGAGE JAVA

Pour le langage de programmation notre choix s'est porté sur le langage JAVA et cela parce que JAVA :

- JAVA est un langage orienté objet simple ce qui réduit les risques d'incohérence.
- JAVA est portable. Il peut être utilisé sous Windows, sur Macintosh et sur d'autres plates formes sans aucune modification. JAVA est donc un langage multi-plateformes, ce qui permet aux développeurs d'écrire un code qu'ils peuvent exécuter dans tous les environnements.
- JAVA possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards, le système de gestion de fichiers, les fonctions multimédia et beaucoup d'autres fonctionnalités.

2. JSP

JavaServer Pages, une technologie relativement récente dans le monde de J2EE, a été conçue pour simplifier encore plus le développement des applications Web. Il est

facile de mettre en œuvre des pages Web en permettant une connexion adaptée entre ces composants d'interface et la logique métier donnée.

Plusieurs raisons ont motivé notre choix, citons principalement :

- Une séparation nette entre la couche de présentation et les autres couches(le modèle MVC).
- Une gestion de l'état de l'interface entre les différentes requêtes.
- Une liaison simple entre les actions côté client de l'utilisateur et le code Java correspondant côté serveur.

3. NETBEANS

NetBeans est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web).

NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OSX.

4. SERVEUR D'APPLICATION

Le serveur d'application Apache Tomcat joue le rôle de conteneur JSP qui permet à sa connexion avec un serveur web de délivrer du contenu dynamique aux clients. La version du serveur utilisée dans le projet est Apache Tomcat 7.0.

Les raisons ayant motivé ce choix sont :

- Apache Tomcat est un logiciel open source.
- Apache Tomcat permet l'implémentation rapide des dernières spécifications des JSP/Servlet.
- Apache Tomcat est connu pour ses larges utilisations dans la communauté JAVA/J2EE en prototypage, il est ouvert et portable.
- Apache Tomcat est considéré comme stable et sécurisé.

5. TREETAGGER

Le treetagger est un outil pour annoter le texte à la partie du discours et le lemme de l'information. Il a été développé par Helmut Schmid dans un projet à l'Institut de linguistique computationnelle de l'Université de Stuttgart. Le treetagger a été utilisé avec succès au tag allemand, anglais, français, italien, néerlandais, espagnol, bulgare, russe, grec, portugais, chinois, swahili et les anciens textes français et est adaptable à d'autres langues, si un lexique et un corpus d'apprentissage marqué manuellement sont disponibles, L'utilisation de TreeTagger se fait via ligne de commande, les paramètres nécessaires sont :

- Le fichier de paramètre de la langue désirée.
- Le fichier en entrée.
- Le fichier résultant.

L'utilisation de Tree Tagger est très facile via java du moment que son exécution se fait à travers la ligne de commande donc il suffit de passer les paramètres de la ligne de commande à java dans un objet Runtime.

TreeTagger a besoin de fichier paramètre de la langue en question, dans notre cas, deux fichiers étaient nécessaires : « french.par » et « english.par ».

6. API UTILISEES

6.1. JDOM API

- JDOM permet un traitement direct des documents XML.
- La sortie est du XML.
- Puissantes et efficaces.
- JDOM exploite de nombreuses caractéristiques de Java (collections, surcharge de méthodes, etc..).
- JDOM est une API open-source : www.jdom.org

7. LANGAGE XML

7.1. DEFINITION

Le **XML**, acronyme de eXtensible Markup Language (qui signifie: langage de balisage extensible), est un langage informatique qui sert à enregistrer des données textuelles. Ce langage a été standardisé par le W3C en février 1998 et est maintenant très populaire. Ce langage, grosso-modo similaire à l'HTML de par son système de balisage, permet de faciliter l'échange d'information sur l'internet. Contrairement à l'HTML qui présente un nombre finit de balises, le XML donne la possibilité de créer de nouvelles balises à volonté.

7.2 AVANTAGES DU XML

- **Lisibilité:** il est facile pour un humain de lire un fichier XML car le code est structuré et facile à comprendre. En principe, il est même possible de dire qu'aucune connaissance spécifique n'est nécessaire pour comprendre les données comprises à l'intérieur d'un document XML.
- **Disponibilité** : ce langage est libre et un fichier XML peut être créé à partir d'un simple logiciel de traitement de texte (un simple bloc-note suffit).
- **Interopérabilité**: Quelques soit le système d'exploitation ou les autres technologies, il n'y a pas de problème particulier pour lire ce langage.
- **Extensibilité**: De nouvelles balises peuvent être ajoutée à souhait.
- Plusieurs parseurs XML différent doivent en principe (s'ils sont bien codés) produire le même résultat.
- Tous les navigateurs internet récents intègrent un parseur XML, pour lire les documents de ce langage informatique.

II. DEPLOIEMENT DE L'APPLICATION

Nous avons développé notre application Web avec la technologie JSP pour qu'elle soit fonctionnelle, elle a été déployée avec le serveur Apache Tomcat qui est à la fois serveur HTTP (serveur Apache) et conteneur JSP, ce qui fait de lui un candidat idéal pour le déploiement de notre application.

Le déploiement de l'application sous Apache Tomcat 7.0 s'effectue en plaçant le répertoire contenant les fichiers de l'application dans le répertoire webapps de Tomcat.

III. IMPLEMENTATION DE L'APPLICATION

Après avoir justifié nos choix du serveur et outils de développement, nous allons illustrer les fonctionnalités de notre système en effectuant quelques prises d'écran qui nous montrent les étapes de l'application d'analyse et de classification automatique des courriels électroniques.

Notre système propose à l'utilisateur, des interfaces qui lui permettront de configurer les différents paramètres utilisés dans l'analyse et la classification.

1. INTERFACES

Les interfaces utilisateur permettent d'accédé au différents modules de l'application précitées :

- Analyse
- Prétraitement
- Classification
- Gestion du corpus
- Validation

Dans ce qui suit, nous allons présenter les différentes interfaces (page web) de ClassEmail.

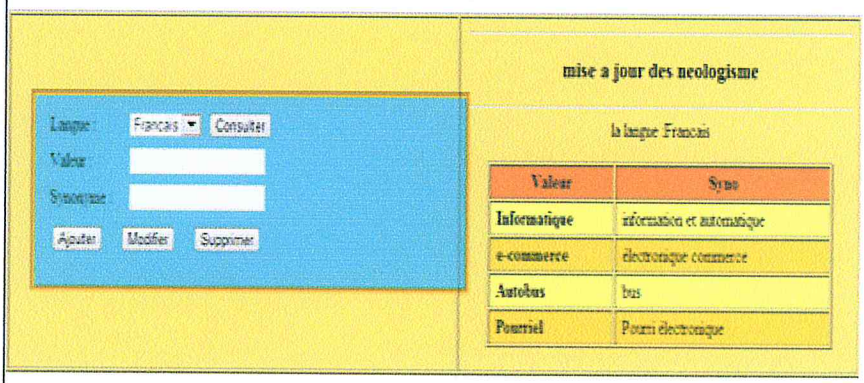
1.1. ACCUEIL

Cette fenêtre permettra d'accomplir toutes les opérations précitées.

Figure 3.1 : Page d'accueil de ClassEmail.

1.2. ANALYSE

La principale tâche de cette rubrique est le paramétrage de l'étape d'analyse. Elle se décompose en quatre sous rubrique principales, décrites dans le tableau 3.1 :

	Objectifs & services offerts	interfaces
GESTION DES NEOLOGISMES	Ajouter, supprimer, modifier, et consulter les néologismes selon la langue choisie	

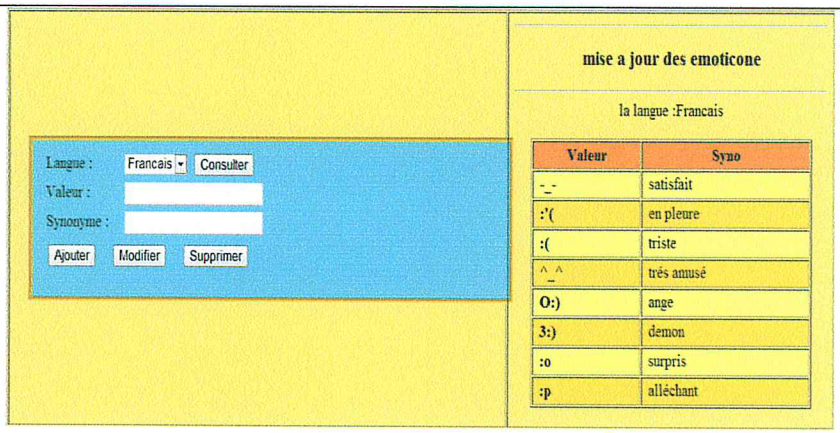
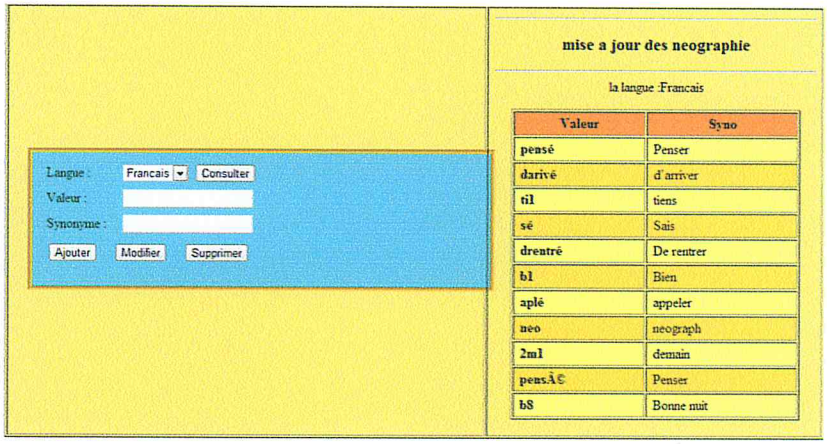
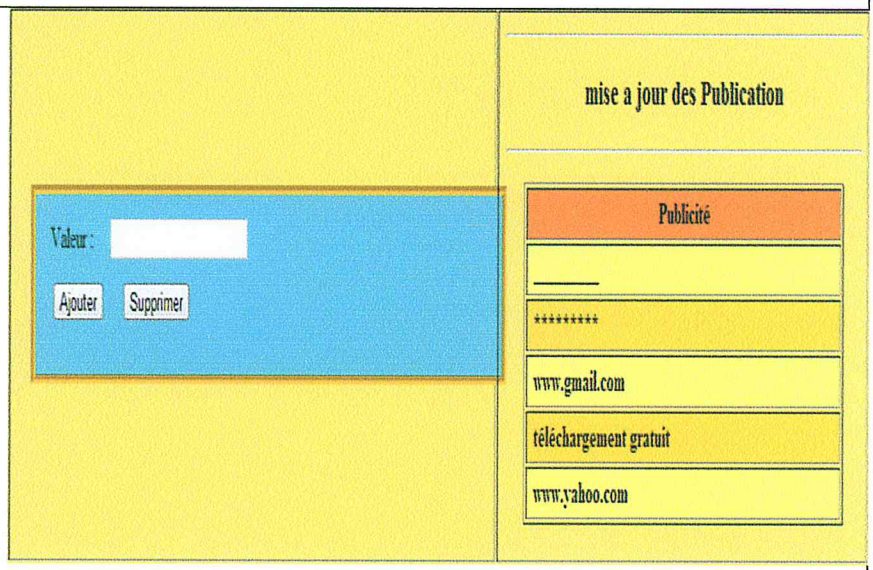
GESTION DES ÉMOTICONE	Ajouter, supprimer, modifier, et consulter les Emoticônes selon la langue choisie	
GESTION DES NEOGRAPHIES	Ajouter, supprimer, modifier, et consulter les néographies selon la langue choisie	
GESTION DES PUBLICITES	Ajouter, supprimer, des publicités.	

Tableau 3.1. les Taches de gestion d'Analyse.

1.3. PRETRAITEMENT

La principale tâche de cette rubrique est le paramétrage de l'étape de prétraitement. Elle se décompose en deux sous rubrique principales, décrites dans le tableau 3.2 :

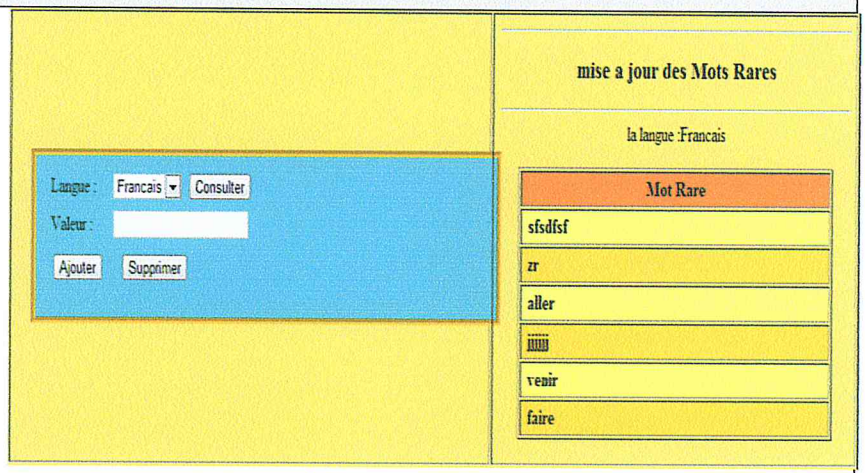
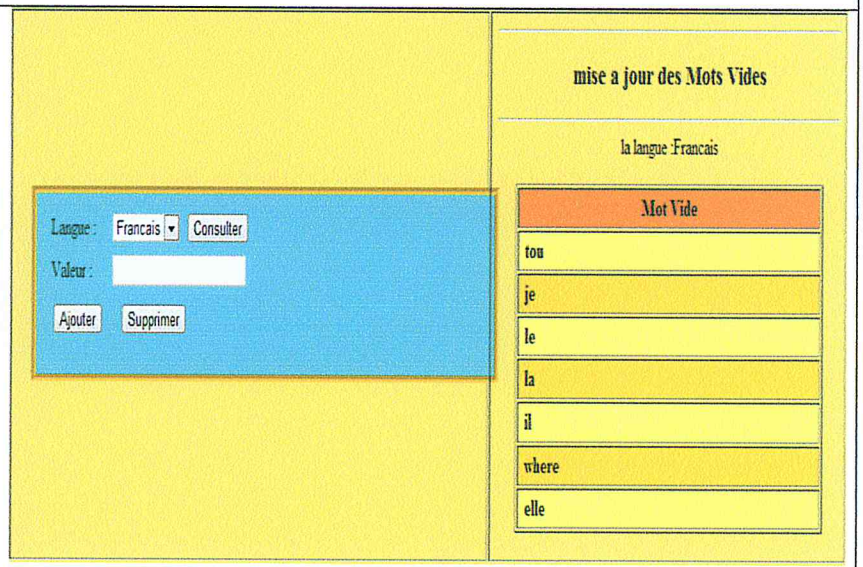
	Objectifs & services offerts	interfaces
GESTION DES MOTS RARES.	Ajouter, supprimer, et consulter les Mots Rares selon la langue choisie	
GESTION DES MOTS VIDES.	Ajouter, supprimer, et consulter les Mots Vides selon la langue choisie	

Tableau 3.2. Les Taches de gestion du Prétraitement.

1.4. CLASSIFICATION

La principale tâche de cette rubrique est le paramétrage de l'algorithme de classification. Elle se décompose en deux sous rubrique principales, décrites dans le tableau 3.3 :

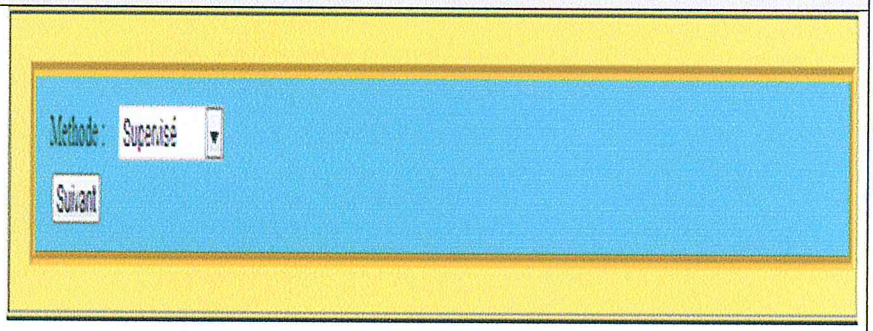
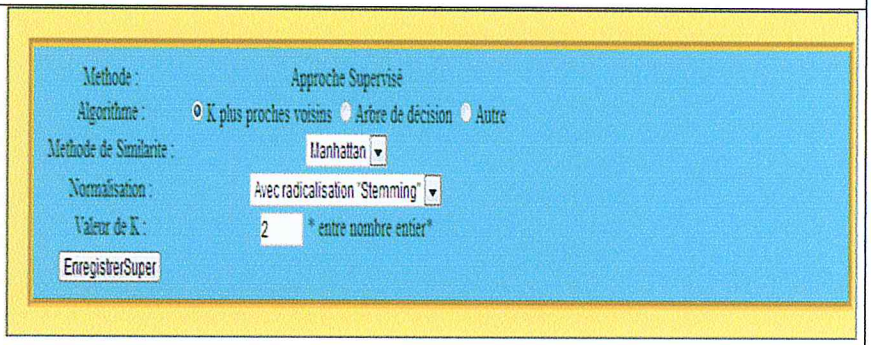
	Objectifs & services offerts	interfaces
GESTION DE CLASSIFICATION	Choisir la Méthode de Classification.	
GESTION DE PARAMETRAGE DE L'ALGORITHME.	Enregistrer le parametrage de l'algorithme choisie, la méthode de similarité et aussi la normalisation	

Tableau 3.3. les Taches de gestion de Classification.

1.5. CORPUS

Cette rubrique donne une vue détaillée sur le contenu du corpus, ses principales fonctions sont décrites dans le tableau 3.4 :

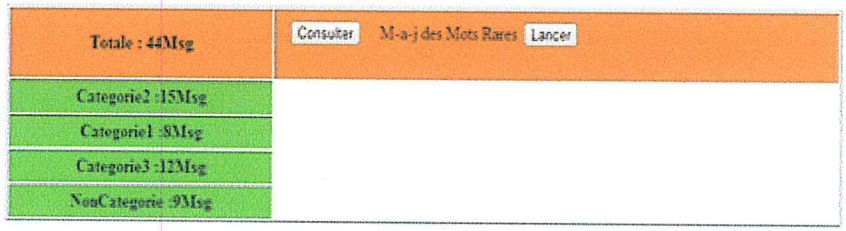
	Objectifs & services offerts	interfaces
GESTION DU CORPUS.	Permet de : Consulter le corpus et ses différentes classes. Afficher des statistiques sur le contenu du corpus. Mettre à jour la liste des mots rares.	

Tableau 3.4. les Taches de gestion du Corpus.

IV. CONCLUSION

Nous avons vu à travers ce chapitre l'implémentation des différentes fonctionnalités de notre application classEmail.

L'interface simple du système permet à l'utilisateur d'exploiter les déférents paramètres des processus du système plus facilement.

Enfin, après avoir effectué des tests sur l'application, les résultats donnent de la satisfaction puisqu'ils répondent parfaitement aux objectifs du travail demandé.

CHAPITRE IV

VALIDATION

CHAPITRE IV- VALIDATION

I. INTRODUCTION

Pour évaluer notre système, nous avons appliqué le processus d'analyse et de classification à un corpus d'étude. Ce processus est constitué comme décrit dans le chapitre 2 de trois étapes, «Analyse du mail», «Prétraitement», et «classification». Une discussion sur les résultats de cette évaluation est donnée à la fin de la section II.

Dans la section III, on dresse une deuxième évaluation, qui consiste à mener des tests pour mesurer les performances du système du point de vue précision et rappel.

Enfin, quelques observations sur les résultats obtenus concluent ce chapitre.

II. EXEMPLE D'APPLICATION DE LA DEMARCHE

Dans ce qui suit, nous allons illustrer à travers des exemples, les résultats correspondants aux différentes étapes de la démarche.

1. MAIL

Le processus d'analyse et de classification a été appliqué à un prototype de mail illustré dans la figure 4.1.

From :
Sent :
To :
Subject :

Bjr,

Je pouré pa venir te cherhcé ☹ , javé oublié ke je devé faire les assurance pr ma voiture.
Essai de te débrouillé ☺ .

Si je fini je te tein o couran, sinon rentr seul.

..... n'oublie p de chatter ....

A+

http://www.yahoo.com

Figure 4.1. Prototype de mail

2. ANALYSE DU MAIL

2.1. ISOLER LES DIFFERENTS CHAMPS

La figure 4.2. Présente le fichier XML correspondant au mail de la figure 1, et qui décrit ses divers champs.

```
<Message>
  <Id>Id86</Id>
  <Adr>admin</Adr>
  <Objet>gol</Objet>
  <Corps>bjr,:
je pouré pa venir te cherhcé :( , javé oublié ke je devé faire les assurance pr ma voiture. essai de te débrouillé :)
si je fini je te tein o couran, sinon rentr seul
 a+
www.yahoo.com</Corps>
  <Langue>TermFr</Langue>
</Message>
</FACEBOOK>
```

Figure 4.2. Fichier XML correspondant

2.2. IDENTIFIER LA LANGUE

La détermination de la langue se fait grâce au fichier StopList.xml, illustré dans la figure 4.3.

```
<?xml version="1.0" encoding="UTF-8"?>
<StopList>
  <TermFr Valeur="le" />
  <TermFr Valeur="il" />
  <TermFr Valeur="elle" />
  <TermEn Valeur="the" />
  <TermEn Valeur="are" />
  <TermFr Valeur="le" />
  <TermFr Valeur="il" />
  <TermFr Valeur="elle" />
  <TermEn Valeur="the" />
  <TermEn Valeur="are" />
  <TermFr Valeur="la" />
  <TermFr Valeur="je" />
  <TermFr Valeur="le" />
  <TermFr Valeur="tou" />
  <TermEn Valeur="from" />
  <TermEn Valeur="how" />
  <TermEn Valeur="that" />
  <TermEn Valeur="what" />
  <TermEn Valeur="when" />
```

Figure 4.3. Fichier StopList.xml

2.3. SUPPRESSION DES PUBLICITE

Chaque publicité rencontrée dans le mail sera supprimé par l'utilisation d'une liste des publicités dans le fichier Publicite.xml (figure4.4).

```
<?xml version="1.0" encoding="UTF-8"?>
<Publicite>
    <Pub Valeur="http://www.yahoo.com" />
    <Pub Valeur="http://www.gmail.com" />
    <Pub Valeur="téléchargement gratuit" />
    <Pub Valeur="_____" />
    <Pub Valeur="*********" />
    <Pub Valeur="*********" />
</Publicite>
```

Figure4.4. Fichier Publicite.xml

2.4. TRANSFORMATION DES EMOTICONES

Chaque émoticône rencontré dans le mail sera remplacé par son équivalent dans le fichier Emoticone.xml (figure4.5).

```
<?xml version="1.0" encoding="UTF-8"?>
<Emoticone>
  <TermFr Valeur=":o" Syno="surpris" />
  <TermEn Valeur=":v" Syno="pacman" />
  <TermFr Valeur="-_-" Syno="satisfait" />
  <TermFr Valeur="3:)" Syno="demon" />
  <TermEn Valeur="3:)" Syno="devil" />
  <TermEn Valeur=":(" Syno="frown" />
  <TermFr Valeur=":(" Syno="triste" />
  <TermEn Valeur=":'(" Syno="cry" />
  <TermFr Valeur=":'(" Syno="en pleure" />
  <TermEn Valeur="O:)" Syno="angel" />
  <TermFr Valeur="O:)" Syno="ange" />
```

Figure.4.5 Fichier Emoticone.xml

Le résultat de cette étape est illustré dans la figure 4.6

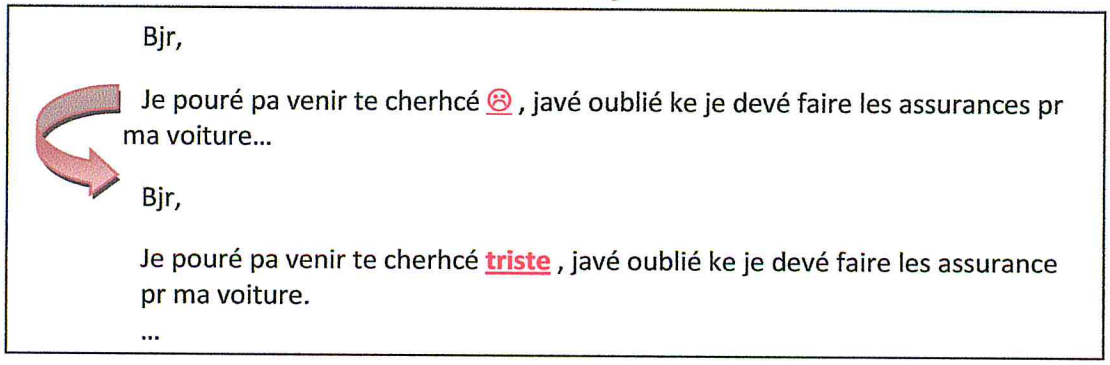


Figure 4.6 . Transformation des émoticônes

2.5.TRANSFORMATION DES NEOGRAPHIES

Chaque néographie rencontrée dans le mail sera remplacé par son équivalent dans le fichier neographie.xml (figure4.7).

```
<?xml version="1.0" encoding="UTF-8"?>
<Neographie>
  <TermFr Valeur="2m1" Syno="demain" />
  <TermFr Valeur="b8" Syno="Bonne nuit" />
  <TermFr Valeur="ti1" Syno="tiens" />
  <TermFr Valeur="b1" Syno="Bien" />
  <TermFr Valeur="aplé" Syno="appeler" />
  <TermFr Valeur="darivé" Syno="d'arriver" />
  <TermFr Valeur="drentré" Syno="De rentrer" />
```

Figure 4.7. Fichier neographie.xml

Le résultat de cette étape est illustré dans la figure 4.8

Bjr,

Je pouré pa venir te cherhcé triste, javé oublié ke je devé faire les assurances pr ma voiture.

...

Bonjour,

Je ne pourrais pas venir te chercher triste, j'avais oublié que je devais faire les assurances pour ma voiture.

...

Figure 4.8. Transformation des néographies

2.6. TRANSFORMATION DES NEOLOGISMES

Chaque néologisme rencontré dans le mail sera remplacé par son équivalent dans le fichier neologisme.xml (figure 4.9).

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Neologisme>
  <TermFr Valeur="Autobus" Syno="bus" />
  <TermFr Valeur="e-commerce" Syno="électronique commerce" />
  <TermFr Valeur="Informatique" Syno="information et automatique" />
  <TermFr Valeur="Pourriel" Syno="Pourri électronique" />
```

Figure 4.9. Fichier néologisme.xml

Le résultat de cette étape est illustré dans la figure 4.10

Bonjour,

Je ne pourrais pas venir te chercher triste, j'avais oublié que je devais faire les assurances pour ma voiture.
..... N'oublie pas de chatter ....

Bonjour,

Je ne pourrais pas venir te chercher triste, j'avais oublié que je devais faire les assurances pour ma voiture.
...
..... N'oublie pas de bavarder....

Figure 4.10. Transformation des néologismes.

2.3. PRETRAITEMENT

Le résultat de cette étape est la représentation interne des mots et de leurs poids correspondants au mail entrant, illustré dans le tableau **.

Mot	Poids
Bjr	0.1090040216782178
Pourrais	0.1090040216782178
Venir	0.09598105333579247
Chercher	0.17445335966922887
Triste	0.09598105333579247
Avais	0.1090040216782178
Oubli	0.09598105333579247
Devais	0.1090040216782178
Faire	0.09598105333579247
Assurances	0.1090040216782178
Voiture	0.09598105333579247

Tableau 4.1. La représentation interne des mots et leur poids.

2.4. CLASSIFICATION

La démarche décrite dans le chapitre 2, a été appliquée sur l'ensemble des messages appartenant au corpus d'étude.

La figure 4.11donne un exemple de la représentation interne des messages classés.

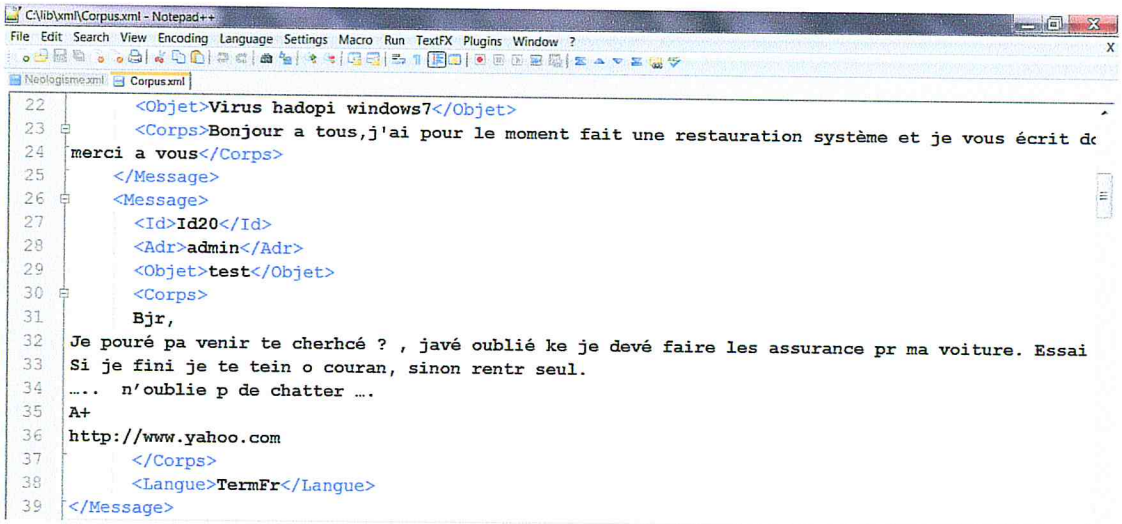


Figure 4.11. Représentation interne du corpus

III. EVALUATION

1. DESCRIPTION DU CORPUS

Pour effectuer nos tests, nous avons travaillé avec un corpus de 20 messages construit à partir d'un ensemble de messages.

Il regroupe une variété de types de messages.

Le corpus a été initialement décomposé en trois classes principales : C1, C2, C3.

C1= « FACEBOOK»

C2= «LINKEDIN »

C3= « GOOGLE+»

2. CRITERES D'EVALUATION

Pour mesurer les performances, nous utilisons les mesures de précision et de rappel. Nous déterminons également la performance globale du système en calculant le pourcentage d'erreur et de succès (tableau 4.2).

Jugement du Système ⇩	**Jugement de l'expert**	
	C	¬C
C	α	B
¬C	γ	Δ

Tableau 4.2. Critères d'évaluation

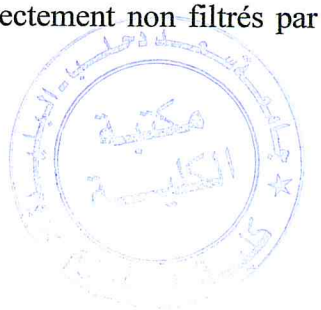
Avec :

α : messages de classe C correctement filtrés (*classés*) par le système ;

β : messages n'appartenant pas à la classe C incorrectement filtrés par le système ;

γ : messages de classe C incorrectement non filtrés (*rejetés*) par le système ;

δ : messages n'appartenant pas à la classe C correctement non filtrés par le système.



Les mesures rappel et précision pour la classe C sont :

$$\text{Rappel} = \frac{\alpha}{a+\gamma} \qquad \text{Precision} = \frac{\alpha}{a+\beta}$$

Les mesures globales erreur et précision du système sont :

$$\text{Erreur\_globale} = \frac{\beta + \gamma}{a + \beta + \gamma + \delta}$$

C'est le rapport entre le nombre total de messages incorrectement classé et incorrectement non classé et le nombre total de messages de la base de test ;

$$\text{Precision\_globale} = \frac{\alpha + \delta}{a + \beta + \gamma + \delta}$$

C'est le rapport entre le nombre total de messages correctement classé et correctement non classé et le nombre total de messages de la base de test

3. EXPERIENCE

Nous présentons, dans ce qui suit, les performances de notre système de classification. Ou nous avons évalué notre système, en comparants les résultats d'une modélisation manuelle (*effectuée par des utilisateurs connaisseur du domaine*) et celle effectuée par notre système.

3.1. RÉSULTATS

C1 « FACEBOOK »

$$\text{rappel} = \frac{5}{5+1} = 0.83 \quad \text{precision} = \frac{5}{5+0} = 1$$

C2 « GOOGLE+ »

$$\text{rappel} = \frac{3}{3+0} = 1 \quad \text{precision} = \frac{3}{3+2} = 0.6$$

C3 « LINKEDIN »

$$\text{rappel} = \frac{4}{4+0} = 1 \quad \text{precision} = \frac{4}{4+1} = 0.8$$

$$\text{erreur\_globale} = \frac{3+1}{12+2+1+2} = 0.2352$$

$$\text{precision\_globale} = \frac{12+2}{12+2+1+2} = 0.8235$$

4. DISCUSSION

A travers l'expérience qu'on a réalisée, nous avons montré l'applicabilité des e-mails au différents processus. Nous avons remarqué que les résultats semblaient plutôt satisfaisants.

L'expérience menée sur notre corpus de messages, très modeste, nous a permis de valider :

- Que notre système a validé une bonne précision de 82% d'e-mail, et avec une erreur de 24% d'e-mail.

Il serait intéressant d'extrapoler l'étude sur d'autres types d'e-mail pour étendre la liste des critères et tester l'adaptabilité.

IV. CONCLUSION

Dans se chapitre nous avons proposé une approche évolutive qui s'adapte à la nature des e-mails au cours du temps et qui exploite le corpus pour classer le courrier électronique.

Ce travail nous a permis de valider :

- Aider à améliorer les résultats de classification, et de relier le message a une classe (catégorie) même s'ils n'ont pas tous les mots en commun, tout en gardant une bonne *précision*.

CONCLUSION GENERALE

CONCLUSION GENERALE

La tâche d'analyse et de classification de courriels est assez difficile en raison des particularités de cette forme de communication. Et aussi, la nature des messages qui se varie au cours du temps, ce qui nécessite une mise à jour fréquente des ses propriétés de bases, ainsi la grande utilisation de se genre d'outil de communication le courriel électronique, notre motivation a été de concevoir un système capable de s'adapter a la nature des courriels dans le temps, ce qui nous conduit a décrire les majeures contributions de ce mémoire qui peuvent se résumer :

- Conception et mise en œuvre du processus d'analyse (*Transformation des émoticônes, néologismes, néographies et suppression des publications*) pour transformer le texto vers texte,
- Conception et mise en œuvre du processus de prétraitement (*Extradions des termes, élimination des mots rares et mots vides, Normalisation*) afin de représenter les courriels dans un modèle vectoriel.
- Etude de différentes méthodes de classification, qui nous a permis de mieux connaître leurs caractéristiques et leur comportement.
- Implémentation d'un algorithme de classification, qui nous a permis de classer les e-mails entrants.
- Validation de la démarche, en exploitant un corpus d'étude.
- Conception et implémentation du système ClassEmail, une solution flexible, qui permet de :
    - Paramétrer les différents processus mise en œuvre (*analyse, prétraitement et classification*).
    - Mettre à jour les diverses bases exploitées dans la démarche (*néologismes, néographies, corpus …*).

On perspectives, nous prévoyons essentiellement :

- Projeter la solution vers d'autres algorithmes de classification.
- Valider la solution sur un corpus fonctionnel.
- Mise en œuvre d'un module de statistique pour l'évaluation des démarches suivies.

BIBLIOGRAPHIE

[STR, 00] M. Stricker. Réseaux de neurones pour le traitement automatique du langage : conception et réalisatiion de filtres d'informations. PhD thesis, Université Pierre et Marie Curie - Paris VI, 2000.

[POR, 80] M. F. Porter. An algorithm of suffix stripping. Program, vol. 14, no.3, juillet 1980.

[ZAK, 09] C. ZAKARIA, Contributions à la détection de conflits relationnels dans les échanges d'e-mails entre personnes en situation de travail coopératif. Une approche fondée sur les modèles statistiques et les ontologies. PhD thesis, Université de Marne-La-Vallée, 2009.

[REB, 10] S. REBIAI. Á travers texto: La néographie dans les pratiques du français en Algérie Cas des SMS des étudiants de la 3ème année du département de langue et littératures françaises, Master thèse, Université Mentouri Constantine, 2010.

[ANI, 02] J. Anis. Communication électronique scripturale et formes langagières :chats et sms. In In Actes des Quatrièmes Rencontres Réseaux Humains /Réseaux technologiques, Poitiers, 2002.

[Arm, 95] R.Armstrong, D.Freitag, T.Joachims, T.Mitchell « WebWatcher : a Learning apprentice for the World Wide Web »,1995.

[Bro, 98] G.Brown, H.A.Chong « The Guru System in TREC-6 »,1998.

[Hay, 90] P.Hayes, S.P.Weinstein « Construe/Tis : A system for content-based,1990.

[GUI, 06] E. Guimier De Neef and J. Veronis. Le traitement des nouvelles formes de communication écrite. Sabah (dir.), Compréhension des langues et interaction, Paris,Hermès-Lavoisier, 2006.

[Lan, 95] K. Lang « NewsWeeder : Learning to Filter Netnews »,indexing of a database of news stories »,1995.

[PAN, 97] R.Panckhurst. La communication médiatisée par ordinateur ou la communication médiée par ordinateur ? Terminologies nouvelles, (17) :56–58, 1997.

[PAN, 07] R. Panckhurst. Discours électronique médié : quelle évolution depuis une décennie ? In Gerbault Jeannine, editor, La langue du cyberespace : de la diversité aux normes, pages p. 121–136. lHarmattan, 2007.

[SAL, 83] G.SALTON, M.MCGILL, Introduction to Modern Information Retrieval, New York: McGraw-Hill, 1983.

[VIE, 04] S. Vienney,C. Melian. La Correction Automatique Du Langage Des Nouvelles Formes De Communication Ecrite", journal="Bulag Correction automatique : bilan et perspectives. (29) :183–196, 2004.

[COL, 96] M. Collot, N. Belmore. Electronic language: a new variety of English.Computer-mediated Communication: Linguistic, Social and Cross-Cultural Perspectives, John Benjamins, pp. 13-28, 1996.

[PAL, 97] J. Palme. Common internet message headers. RFC 2076,1997.

[RES, 01] P. Resnick. Internet message format. RFC 2822, avril 2001.

[FRE, 96] N. Freed et N. Borenstein. MIME part one: format of internet message bodies.RFC 2045, novembre 1996. www.imc.org/rfc2045

[FRE, 96] N. Freed et N. Borenstein. MIME part two: media types. RFC 2046, novembre 1996.www.imc.org/rfc2046

[FRE, 96] N. Freed et N. Borenstein. MIME part five: conformance criteria and examples. RFC 2049, 1996.

[JAC, 04] F. Jacquenet, C. Largeron, et S. Chapaux. Veille technologique assistée par la fouille de textes. In G. Hébrail, L. Lebart, and J.-M. Petit, editors, EGC, volume RNTIE-2 of Revue des Nouvelles Technologies de l'Information, pages 429–440. Cépaduès-E'ditions, 2004.

[LAN, 97] T.Landauer, S.Dumais, A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review, pages 211–240, 1997.