République Algérienne Démocratique et Populaire.
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université A. Mira Béjaïa
Faculté des Sciences Exactes
Département Informatique

Mémoire de Master

En Informatique
Option :
Réseaux et Systèmes Distribués

Thème :

Extraction d'information à base d'ontologie pour la construction d'un profil utilisateur.

Présenté par :
M^{elle} **AIT RADI** ***Taklit*** **et** M^{elle} **IGGUI** ***Taous.***

Soutenu le 24\06\2013 devant le jury composé de :

Présidente	**M^{elle} N. KHOULALEN**	**M.A.A**	**U. A/Mira Béjaïa.**
Promotrice	**D^{r} H. NACER**	**M.C.B**	**U. A/Mira Béjaïa.**
Co-Promoteur	**M^{r} Y. SKLAB**	**Magister**	**U. A/Mira Béjaïa.**
Examinatrice	**M^{me} Z. TAHAKOURT**	**M.A.A**	**U. A/Mira Béjaïa.**
Examinateur	**M^{r} R. OUZEGANE**	**M.A.A**	**U. A/Mira Béjaïa.**

Béjaïa, juin 2013.

En premier lieu nous remercions Dieu le tout puissant pour toute la volonté et le courage qu'il nous a donné pour l'achèvement de ce travail.

Nous tenons à remercier très chaleureusement, D^r **H. NACER** pour la qualité de l'encadrement, dont elle nous a fait bénéficier, pour avoir mieux guidé et structurer ce travail en conjuguant habilement, disponibilité, conseils et critiques constructives. Ainsi que de nous avoir fait profiter de son expérience.

Nous ne saurons trouver les termes qu'il faut pour exprimer notre profonde gratitude et la reconnaissance que nous devions à notre Co-promoteur **M. Y.SKLAB** et son aide judicieuse, sa disponibilité, ses orientations et ses précieux conseils. L'aboutissement de ce travail doit beaucoup à sa contribution.

Toutes nos sincères gratitudes et notre profond respect à M^{elle} **N.KHOULALEN** qui nous a honoré en acceptant de présider notre soutenance.

Nous remercions vivement M^e **Z. TAHAKOURT** et **M. R.OUZEGANE** qui ont accepté d'examiner et de valoriser notre travail.

Nous remercions également **M. M.Omar** qui nous a honoré en acceptant d'assister à la présentation de ce travail.

Nous tenons à remercier tous ceux qui ont contribué à ce travail parfois sans le savoir ou du moins sans mesurer la portée de leurs influences.

Sans oublier de remercier l'ensemble des enseignants du département informatique ayant contribués à notre formation durant notre cycle d'étude.

TAKLIT et TAOUS

A mes chers parents, aucune dédicace ne saurait être assez éloquente pour vous exprimer ce que vous méritez pour tous les efforts et les sacrifices que vous n'avez jamais cessé de consentir pour mon instruction et mon bien-être. Je vous rends hommage par ce modeste travail en guise de ma reconnaissance éternelle et de mon infini amour.

Que Dieu tout puissant vous garde et vous procure santé, bonheur et longue vie pour que vous demeuriez le flambeau illuminant le chemin de vos enfants.

A mon grand père et grandes mères, que dieu vous protège. A mes soeurs **H, O, L**, *frères* **M, N**, *oncles et leurs femmes, cousins et cousines ... Les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je vous porte.*

A ma meilleur amie et binôme TAOUS, *en témoignage de l'amitié qui nous uni et des souvenirs de tous les moments que nous avons passé ensemble, je te souhaite des lendemains épiques, un avenir glorieux et magique, j'espère que le fruit de nos efforts fournis, jours et nuits, te mènera vers le bonheur fleuri. On finit toujours par y arriver.*

A toi **MASSI** *, ton soutien et tes conseils m'ont été d'un grand secours, je te remercie pour tous.*

A toutes mes amies : HANNANE, HAYET, LYNDA, NASSIMA, OUIDAD, SABRINA, SORIA et ceux qui me sont chers, la liste est bien longue.

TAKLIT

Toutes les lettres ne sauraient trouver les mots qu'il faut, tous les mots ne sauraient exprimer ma gratitude, mon amour, mon respect ainsi que ma reconnaissance aux deux symboles de sacrifice, et de tendresse, à vous mes très chers parents, je vous dédie ce modeste travail. Quoi que je fasse, je ne pourrais jamais vous récompenser, que Dieu, vous procure santé, bonheur et longue vie.

A toute ma famille : oncles, tentes, cousins et cousines, à tous ceux qui ont contribué de près ou de loin à l'accomplissement de ce travail, à ma très chère Lili, en témoignage de tous ces moments et souvenirs partagés, que bonheur, santé, succès et réussite soient au rendez vous.

On dit souvent que le trajet est aussi important que la destination. Ces cinq années m'ont permis de bien comprendre la signification de cette phrase toute simple. Ce parcours, en effet, ne s'est pas réalisé sans défis et sans soulever de nombreux obstacles. J'en remercie Dieu le tout puissant, de m'avoir donné la foi et de m'avoir permis d'en arriver là, merci à tous mes ami(e)s, ces personnes adorables, chaleureuses, compréhensives, un peux fous mais inoubliables, qui m'ont soutenus, encouragé et cru en moi, merci pour tous ces souvenirs qui seront, jusqu'à l'éternité, gravés dans ma mémoire.

" La vie s'écrit avec un I, c'est le I d'un amI "

TAOUS

Table des matières

Table des figures

Liste des tableaux

Liste des algorithmes

Liste des Abréviations

DTD	**D**ocument **T**ype **D**efinition
EI	**E**xtraction d' **Information**
EN	**E**ntité **N**ommée
GC	**G**raphe **C**onceptuel
HTML	**H**yper**T**ext **M**arkup **L**anguage
IDF	**I**nverse **D**ocument **F**requency
IHM	**I**nterface **H**omme **M**achine
Q-R	**Q**ustion **R**éponse
REG	**RE**sumeur à base de **G**raphes
RI	**R**echerche d' **I**nformation
SRI	**S**ystème de **R**echerche d' **I**nformation
TDT	**T**opic **D**etection **T**raking
TF	**T**erm **F**requency
TAL	**T**raitement **A**utomatique des **L**angues
WS	**W**eb **S**émantique
XML	e**X**tensible **M**arkup **L**anguage

Introduction générale

DE plus en plus, les systèmes d'information sont accessibles à travers Internet ou Intranet. Ils permettent aux utilisateurs d'accéder à une masse énorme d'information provenant d'une ou de plusieurs sources. Cette augmentation rapide a engendré le problème de comment retrouver une information qui nous intéresse dans cette grande masse de données. Effectivement, le besoin en information est primordial dans de nombreux domaines, comme celui de la recherche ou celui de la veille scientifique et technique, mais avec la croissance exponentielle de l'information électronique sur le Web, la satisfaction des besoins en informations d'un utilisateur demeure un but de plus en plus difficile à atteindre pour les systèmes actuels de recherche et d'accès à l'information [27].

Afin de traiter ce problème, une discipline toute entière est née, appelée Extraction d'Information (EI). Elle s'intéresse à l'acquisition, l'organisation, le stockage et la recherche des informations, ainsi au développement des techniques, méthodes et outils permettant de retrouver les informations pertinentes qui assurent la satisfaction des besoins de l'utilisateur.

Les utilisateurs, de nos jours sont devenus plus favorables à l'utilisation des systèmes d'informations, avec la caractéristique de ne pas être intéressés tous par les mêmes informations, et en se présentant avec différents centres d'intérêt, différents besoins et différentes activités, ce qui freine le rendement des systèmes d'information sur le Web vis-à-vis des utilisateurs.

En effet, un SRI (Système de Recherche d'Information) typique retourne la même liste des résultats pour une même requête soumise par des utilisateurs ayant des besoins en information pourtant différents. Par exemple, pour la requête "Chat", les utilisateurs vétérinaires s'intéressent à retrouver des résultats concernant le squelette anatomique "Chat", tandis que les archéologues s'intéressent à retrouver des résultats concernant les chats et la civilisation Egyptienne antique, alors que les utilisateurs physiciens s'attendent à des résultats concernant le théorème du chat de Schrödinger. La faille de ces systèmes

réside en partie dans le fait de ne pas tenir en compte des préférences de l'utilisateur.

D'où l'importance de la notion du " *profil utilisateur* " qui regroupe toutes les informations pouvant caractériser un individu, et pouvant ainsi orienter les résultats d'un système de traitement d'information en fonction d'un utilisateur donné. Le profilage informatique est une nouvelle discipline qui vient de s'intégrer dans plusieurs domaines ; militaire, économique, criminel, la recherche d'information (personnalisée), la sécurité informatique... etc.

Le Web a rigoureusement changé la disponibilité en ligne des données et la quantité de l'information électronique échangée. Il a révolutionné l'accès à l'information et la gestion des connaissances dans les grands organismes. Le recrutement électronique est l'une des applications typiques d'une telle approche de gestion de la connaissance à travers le Web. Internet a prouvé que les méthodes de recrutement classiques qui utilisent les annonces d'emploi dans les journaux et les magasins, qui sollicitent les agences de recrutement et qui passent par l'inscription dans les sociétés de recherche, sont trop lentes, chères et insuffisantes dans leurs capacités de fournir des candidats de haute qualité dans les plus brefs délais [91].

Le Web a permis une évolution importante du marché du recrutement électronique, il s'est transformé rapidement en un outil de recrutement performant. Afin de satisfaire les besoins des recruteurs et de pouvoir assurer un rapprochement automatique entre les offres et demandes d'emplois, nous proposons un système de construction de profil utilisateur à partir des documents textuels, collectées à partir des e-mails électroniques. Il s'agit de l'extraction de toutes les informations concernant ses préférences, ses centres d'intérêt ainsi que ses données personnelles, en utilisant un ensemble de méthodes statistiques ainsi que la sémantique des ontologies du domaine. L'idée générale étant de retrouver et extraire ces connaissances à partir des e-mails électroniques concernant le recrutement dans le domaine informatique.

Dans le but d'atteindre notre objectif, nous avons commencé par la détermination des informations pertinentes à notre système, en utilisant un ensemble de techniques d'extraction d'information. Nous avons utilisé les ontologies du domaine, afin de fournir un cadre unificateur pour réduire et éliminer les confusions conceptuelles et terminologiques, et afin d'assurer une extraction d'information orientée domaine. Nous avons également exploité un ensemble de relations grammaticales et de règles de correspondance afin d'associer à chaque attribut du profil utilisateur une information, détectée pertinente, adéquate.

Nous avons organisé ce mémoire en quatre chapitres :

Le premier chapitre intitulé " *Généralités sur l'extraction d'information* ", nous

présentons la discipline d'Extraction d'Information (EI) à travers plusieurs définitions, son principe et son processus. Ensuite, nous intégrons la notion de l'ontologie dans l'EI avec sa définition, ses rôles ainsi que ses différents types. Nous terminons le chapitre avec la représentation des différents domaines d'application de l'EI.

Dans le deuxième chapitre nommé " *Profil utilisateur* ", nous nous focaliserons sur la notion du profil utilisateur qui est le résultat du processus d'extraction d'information pertinente, avec la présentation des différentes approches de représentation et de modélisation du profil utilisateur.

Le troisième chapitre qui est "*État de l'art*", synthétise les travaux les plus importants ainsi que les solutions standards existantes dans le domaine de l'extraction d'information pour la construction d'un profil utilisateur.

Pour le dernier chapitre " *Proposition d'un système de construction d'un profil utilisateur*", nous décrivons notre système de construction du profil utilisateur dans le domaine de recrutement informatique. Nous présentons l'architecture globale du système proposé qui comporte cinq modules, en combinant entre les approches statistiques, les méthodes des résumés et l'utilisation des ontologies pour l'extraction de toutes les informations pertinentes pouvant représenter le profil utilisateur.

Enfin, nous terminons par une conclusion et des perspectives.

Chapitre 1

Généralités sur l'Extraction d'Information (EI)

1.1 Introduction

Le développement et l'évolution rapide des technologies, ainsi le progrès des outils de production d'informations tels que les éditeurs de textes, ont permis la production quotidienne d'une énorme masse d'information.

Cette augmentation rapide du volume d'information a engendré le problème de comment retrouver et extraire une information qui nous interesse, et d'où la naissance de la discipline d' **Extraction d'Information** (EI).

1.2 Définitions de l'Extraction d'Information

Dans la littérature, on trouve plusieurs définitions pour l'extraction d'information dans les textes en langue naturelle :

Définition 1.2.1. *L'Extraction d'Informations (EI)* est le processus qui permet d'identifier l'information pertinente, où les critères de pertinence sont définis sous forme de patrons (templates) à remplir [90].

Définition 1.2.2. L'*EI* consiste à remplir automatiquement des formulaires ou une banque de données à partir de textes écrits en langue naturelle [63].

Définition 1.2.3. L'*EI* est l'activité qui consiste à remplir une source de données structurées (base de données) à partir d'une source de données non structurées (texte libre) [74].

Définition 1.2.4. L'*EI* consiste à identifier l'information bien précise d'un texte en langue naturelle mais aussi à pouvoir la représenter sous forme structurée [85].

Définition 1.2.5. Pour notre définition, l'EI consiste à analyser des textes écrits en langage naturel dans le but d'obtenir des informations en vue d'une application précise.

1.3 Principe de l'Extraction d'Information

L'extraction met en oeuvre une analyse du texte pour interpréter et construire une représentation formelle qui permettra d'apporter automatiquement des réponses précises à l'utilisateur. Il ne s'agit donc pas simplement de sélectionner un fragment brut du texte, mais de mettre des éléments en relation pour restituer une information complète et structurée. Sauf dans les cas très simples, c'est une tâche difficile qui requiert une part de compréhension et nécessite des connaissances, des ressources lexicales, sémantiques et conceptuelles adaptées aux documents et au domaine à traiter [3].

Les techniques d'extraction d'information, basées sur des technologies du traitement automatique des langages (TAL) permettent de structurer une information textuelle qui est au départ dépourvue de toute structure logique. Le champ de l'EI est souvent décomposé en plusieurs sous problèmes qui sont :

- L'extraction d'entités nommées ;
- L'extraction de descripteurs thématiques (libres ou normalisés) ;
- L'extraction de phrases importantes sous un point de vue donné ;
- L'extraction d'attributs ;
- L'extraction d'associations entre entités nommées et descripteurs ;
- L'extraction de correspondances multilingues.

1.4 Le processus d'Extraction d'Information

Le processus d'extraction d'information est utilisé dans des applications de compagnies d'assurance, compagnies bancaires (crédit, prédiction du marché, détection de fraudes), marketing (comportement des consommateurs, " mailing " personnalisé), recherche médicale (aide au diagnostic, au traitement, surveillance de population sensible), réseaux de communication (détection de situations alarmantes, prédiction d'incidents), analyse de données spatiales,... etc.

Ce processus, décrit dans la Figure 1.1, comprend globalement 4 phases [9] :

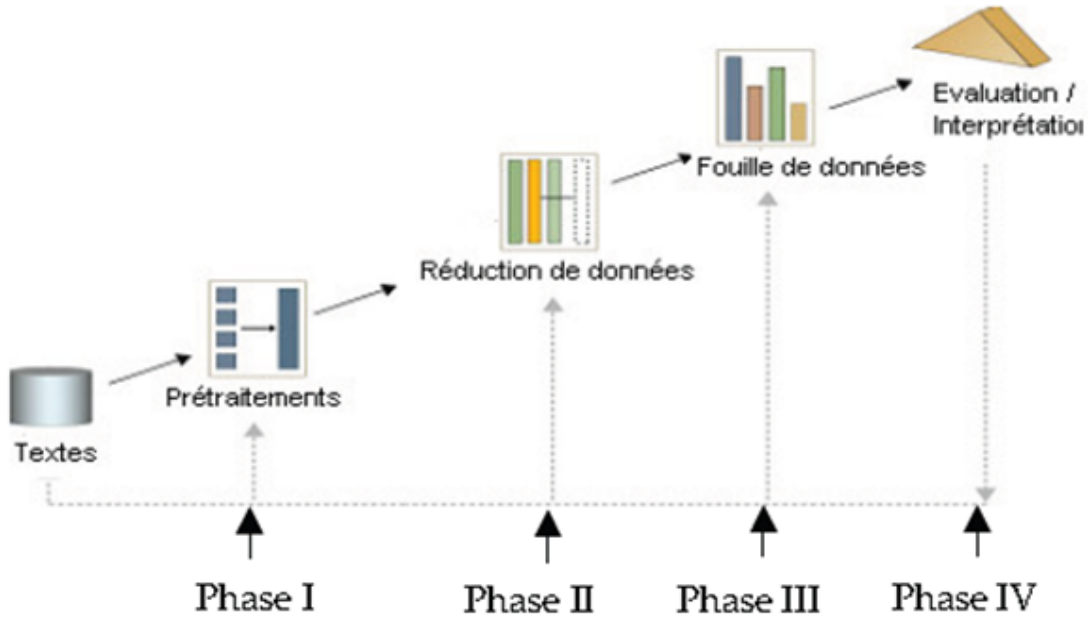


FIGURE 1.1 – Le processus d'extraction de connaissance à partir d'un texte [9] .

• **Phase I :** Pré-traitements et transformations des textes.

Cette phase fait appel à des techniques de pré-traitements des textes : nettoyage des textes, suppression des mots peu informatifs, et/ou normalisation. En outre, étant donné qu'il n'est pas possible de traiter les documents dans leur globalité, la transformation offre une représentation réduite des différents documents.

Effectivement, l'objectif de ce pré-traitement est de minimiser l'espace de recherche. En effet, même si les capacités des ordinateurs évoluent constamment, il n'est malheureusement pas possible de traiter les documents dans leur intégralité. En fonction des différentes thématiques de recherche, cet objectif est mené de différentes manières, par exemple, les approches issues du traitement automatique du langage naturel offriront des techniques pour obtenir les différents composants d'une phrase, pour associer du sens aux termes manipulés, etc.

• **Phase II :** Sélection et réduction des données.

Cette phase est généralement regroupée avec la première et a pour but soit de réduire réellement le volume des contenus textuels, soit de minimiser l'espace de recherche.

Telle l'extraction des termes, qui est basée sur des techniques d'analyse statistique de mots clés. L'idée principale consiste à analyser le contenu des documents utilisateur et d'en extraire des mots clés significatifs qui décrivent son contenu. Ces termes constituent les données d'entrée pour l'algorithme d'apprentissage du profil.

Dans le cas où le profil contient simplement que des mots-clés, ces termes vont être

regroupés en paquets selon leur degré de similarité pour former les centres d'intérêts. Dans le cadre d'une approche vectorielle, les termes vont être pondérés pour former des vecteurs de termes représentant les centres d'intérêts. Le poids attribué à chaque mot clé permet de traduire son degré d'importance dans le profil [69].

• **Phase III :** En appliquant des techniques et algorithmes de fouille de données (algorithme clustering, K-means, EM [57], les réseaux bayésiens, ect), l'objectif est de mettre en évidence des caractéristiques ou des modèles contenus implicitement dans les données.

• **Phase IV :** Analyse, interprétation et validation des résultats. Le but de cette dernière phase est d'interpréter la connaissance extraite lors de l'étape précédente, pour la rendre lisible et compréhensible par l'utilisateur et permettre ainsi de l'intégrer dans le processus de décision.

1.4.1 Processus d'indexation

L'indexation des documents consiste à représenter les objets (documents) par des descripteurs généralement représentés sous forme d'une liste de mots clés et de poids qui leur sont associés.

Les modes d'indexation sont les suivants :

1.4.1.1 Indexation manuelle :

Réalisée par un spécialiste du domaine qui analyse le contenu du texte pour identifier les termes représentatifs du document, c'est une indexation très précise mais très coûteuse.

1.4.1.2 Indexation automatique :

Le processus d'indexation est entièrement informatisé, deux approches sont utilisées pour extraire les termes représentatifs des documents et des requêtes : statistiques [16] et linguistiques [72]. L'approche statistique se base sur la distribution statistique des termes dans le document. L'approche linguistique se base sur les techniques de traitement du langage naturel, telles que l'analyse lexicale, syntaxique et sémantique.

L'indexation automatique s'agit d'extraire les termes des documents grâce à un processus automatique. Cependant, le choix final reste au spécialiste du domaine pour établir les relations entre les mots clés et choisir les termes significatifs.

De manière générale, elle est réalisée selon les étapes suivantes :

• **Analyse lexicale :** l'analyse lexicale (tokenization) est le processus qui permet de convertir le texte d'un document en un ensemble de termes. Un terme est un groupe de caractère constituant un mot significatif.

• **L'élimination des mots vides :** un des problèmes majeurs de l'indexation consiste à extraire les termes significatifs des mots vides (pronoms personnels, prépositions,...). On distingue deux techniques pour éliminer les mots vides :

- L'utilisation d'une liste de mots vides (stop words).
- L'élimination des mots dépassant un certain nombre d'occurrences dans la collection.

• **Lemmatisation :** un mot donné peut avoir différentes formes dans un texte. Pour résoudre le problème, une substitution des termes par leur racine ou lemme est utilisée. Cinq stratégies de lemmatisation sont distinguées : la table de consultation (dictionnaire, tel Tree-Tagger), l'élimination des affixes (algorithme de Porter), la troncature, les variétés de successeur et la méthode des n-grammes.

1.4.2 Pondération des termes

La pondération permet de mesurer l'importance d'un terme dans chaque document de collection, cette pondération est utilisées pour le calcul de la pertinence d'un document en réponse à une requête utilisateur. Les travaux de Zipf [35]et Luhn [71] ont montré que la fréquence d'un terme aussi bien dans un document que dans une collection de documents est un bon indicateur de son importance.

Les méthodes de pondérations existantes reposent sur la pondération locale et la pondération globale.

1.4.2.1 Pondération locale :

Cette mesure est proportionnelle à la fréquence du terme dans le document. L'idée sous-jacente est que plus un terme est fréquent dans un document, plus il est important dans la description de ce document, elle est notée ***TF*** (Terme Frequency) et qui est souvent exprimée selon l'une des des formules suivantes [35], [71] :

$$TF_{t,d} = valeurbrute. \tag{1.1}$$

$$Ntf_{t,d} = \frac{TF_{t,d}}{Max(TF_{t,d})} \tag{1.2}$$

- $TF_{t,d}$: Fréquence du terme t dans le document d.
- $Ntf_{t,d}$: Fréquence normalisée du terme t dans le document d.
- **Max**($TF_{t,d}$) : Fréquence maximum du terme t dans la collection.

1.4.2.2 Pondération globale :

mesure l'importance du terme par rapport à la collection, ce facteur dépend de la fréquence inverse dans le document, souvent désigné par ***IDF*** (Inverse Document Frequency). L'idée sous-jacente est que les termes qui apparaissent dans peu de documents de la collection sont plus représentatifs du contenu de ces documents que ceux qui apparaissent dans tous les documents de la collection.

Cette mesure est exprimée selon l'une des formules suivantes [35], [71] :

$$IDF(t) = \log \frac{N}{df(t)} \tag{1.3}$$

$$IDF(t) = \log \frac{N - df(t)}{df(t)} \tag{1.4}$$

- **IDF(t)** : Fréquence inverse du termes t dans la collection.
- **N** : Nombre de documents de la collection.
- **df(t)** : Nombre de documents contenant le terme t.

Une bonne approximation de l'importance du terme dans le document est donnée par la combinaison des deux facteurs ***(TF * IDF)***, qui est la fonction de pondération la plus répondue.

La formule finale de la norme TF * IDF qui désigne le poids d'un terme est :

$$W(t,d) = TF(t,d) * IDF(t) \tag{1.5}$$

1.5 Extraction à base d'ontologie

La réalisation du Web Sémantique (WS) à grande échelle implique l'annotation généralisée de documents Web à l'aide de bases de connaissances ontologiques. Depuis plusieurs années, de nombreux travaux de recherche démontrent clairement que l'Extraction d'Information (EI) est essentielle à l'automatisation de ce processus.

L' "*Extraction d'Information Basée sur les Ontologies* " (EIBO) est un sous-domaine de l'EI.

Définition 1.5.1. L'ontologie

Cette notion a été reprise par les chercheurs dans le domaine de l'intelligence artificielle et utilisée dans le cadre de construction des systèmes à base de connaissances. L'idée était de séparer, d'un côté, la modélisation des connaissances d'un domaine, et d'un autre côté, l'utilisation de ces connaissances (i.e. le raisonnement).

Dans ce contexte, plusieurs définitions des ontologies ont été proposées. La première a été proposée par [78] : " *Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui permettent de combiner les termes et les relations afin de pouvoir étendre le vocabulaire* ".

Cette définition descriptive donne un premier aperçu sur la manière de construire une ontologie, à savoir l'identification des termes et des relations d'un domaine ainsi que les règles pouvant s'appliquer sur ces derniers.

Deux années plus tard, l'auteur [83], donne la définition qui est devenue la plus utilisée dans la littérature : " *Une ontologie est une spécification explicite d'une conceptualisation* ". La conceptualisation se réfère ici à l'élaboration d'un modèle abstrait d'un domaine du monde réel en identifiant et en classant les concepts pertinents décrivant ce domaine. La formalisation consiste à rendre cette conceptualisation exploitable par des machines.

Dans cette même logique, les auteurs dans [67], proposent leur définition : " *Une ontologie est une théorie logique proposant une vue explicite et partielle d'une conceptualisation*".

Depuis, de nombreuses définitions, à la fois complémentaires et précises, ont vu le jour. Les auteurs dans [64], soulignent la dépendance entre la formalisation de l'ontologie et l'application dans laquelle elle va être utilisée : " Une ontologie organise dans un réseau des concepts représentant un domaine. Son contenu et son degré de formalisation sont choisis en fonction d'une application ".

1.5.1 Rôles des ontologies

Historiquement, la notion d'ontologie est apparue pour satisfaire des besoins d'interopérabilité dans les systèmes informatiques et de réutilisation. On attend d'elles qu'elles améliorent la communication non seulement entre machines, mais aussi entre humains et machines ou encore entre humains par le biais de logiciels. Les propriétés de ce type de

structure de données ont permis de diversifier leur utilisation à différentes applications, en particulier la gestion des connaissances et le Web sémantique [53]. Elles sont utilisées pour :

- Résoudre des problèmes de compréhension et faciliter le partage des connaissances entre personnes de spécialités différentes ;
- Assurer l'interopérabilité entre applications à base de connaissances ;
- Accéder à des ressources hétérogènes ;
- Permettre la réutilisation de modèles de connaissances ;
- Faciliter la communication entre agents logiciels ;
- Améliorer les processus de recherche d'informations.

1.5.2 Types d'ontologies

Les auteurs dans [36] définissent deux grandes typologies d'ontologies : une typologie fondée sur la structure de la conceptualisation et l'autre fondée sur le sujet de la conceptualisation.

Dans la première typologie, ils distinguent trois catégories à savoir :

- Les ontologies terminologiques (lexiques, glossaires...) ;
- Les ontologies d'information (schéma d'une BD) ;
- Les ontologies des modèles de connaissances.

Dans la deuxième typologie, qui est la plus citée, ils distinguent quatre catégories :

• **Les ontologies d'application :** elles contiennent toutes les informations nécessaires pour modéliser les connaissances pour une application particulière.

• **Les ontologies de domaine :** elles fournissent un ensemble de concepts et de relations décrivant les connaissances d'un domaine spécifique.

• **Les ontologies génériques :** elles sont similaires aux ontologies de domaine, mais les concepts qui y sont définis sont plus génériques et décrivent des connaissances tels que l'état, l'action, l'espace et les composants.

Généralement, les concepts d'une ontologie de domaine sont des spécialisations des concepts d'une ontologie de haut niveau.

• **Les ontologies de représentation :** (méta-ontologies) elles fournissent des primitives de formalisation pour la représentation des connaissances. Elles sont généralement utilisées pour écrire les ontologies de domaine et les ontologies de haut niveau. Exemples : Frame Ontology [83].

1.6 Domaines d'application de l'Extraction d'Information

Les domaines d'application qui ont recours à l'extraction de l'information sont nombreux, mais leurs besoins ne sont pas toujours identiques. Selon les domaines, l'extraction d'information consiste en l'une ou plusieurs des tâches suivantes : filtrer un flux d'informations entrant pour éliminer le bruit, guider la navigation dans un espace d'information trop vaste, effectuer un résumé pour un document en obtenant que les phrases pertinentes, ajuster le résultat d'une requête, ect. Parmi les domaines qui ont le plus souvent recours à l'extraction d'information, on peut citer les suivants :

1.6.1 Analyse des citations dans les publications scientifiques

Une des applications intéressantes des systèmes d'EI est l'analyse des citations dans les publications scientifiques. L'idée sous-jacente est de collecter avec un robot des publications sur des sites académiques. Ces publications ayant une structure relativement homogène, un système d'EI extrait des informations typées, telles que le titre, l'auteur, le résumé, la bibliographie, ainsi que les contextes de citations de ces références. Des analyses de liens de citation peuvent être effectuées sur les informations extraites [39].

1.6.2 Utilisation dans des systèmes de question-réponse et de suivis d'actualités

L'extraction d'information est une composante nécessaire aux systèmes question-réponse (Q-R). En effet, les systèmes de Q-R doivent fournir la réponse à une question et non pas les documents susceptibles de contenir cette réponse. Les systèmes d'EI leur sont donc utiles pour la reconnaissance et le typage des entités nommées. Lorsque le système de Q-R détermine le type de question et le type de réponse à fournir, il peut chercher une phrase contenant ce type grâce aux annotations d'un système d'EI et fournir la réponse à l'utilisateur.

Les systèmes de détection et de suivi d'actualités (Topic Detection Traking ou TdT) ont également à bénéficier des résultats d'un système d'EI, notamment les résultats du module de repérage des entités nommées. L'objectif des systèmes de TdT est de détecter le cycle de vie des événements dans un flux entrant de texte d'actualités. Ces systèmes doivent donc détecter la première apparition d'une nouvelle, son évolution au cours d'une période, sa diminution et sa disparition[39].

1.6.3 Application au droit et à la veille sur la criminalité

Les systèmes d'EI ont trouvé des applications en droit pénal et en veille sur la criminalité. Dans la première application, les documents saisis lors des raids policiers sont scannés, puis convertis en formulaires avec la reconnaissance de certaines entités telles que les noms des personnes, les lieux et les organisations.

Ces formulaires sont intégrés dans une base de données qui peut être soumise à tous les traitements statistiques habituels [39].

1.6.4 Gestion des ressources humaines

Cela concerne l'analyse automatique des sondages internes sur la motivation du personnel, l'analyse automatique des curriculum vitae pour la recherche de compétences particulières [39].

1.6.5 Veille concurrentielle

Il peut s'agir de l'analyse systématique des articles de presse, des dépêches et de sites de concurrents pour la surveillance des concurrents, recherche de clients potentiels, surveillance de l'image de marque de la société.

1.6.6 Découverte scientifique et bio-informatique

Ce domaine connaît actuellement une explosion de publications dont l'exploration manuelle est devenue impossible. Afin de ne pas passer à côté des découvertes faites par d'autres chercheurs, les outils de la fouille de texte peuvent "découvrir" certaines relations de causalité, de dépendance entre les objets biologiques (gènes, séquence ADN)[39].

1.6.7 Commerce électronique

Ce domaine d'application recouvre la vente des produits de toute nature. La gestion de la relation client est un domaine d'application de prédilection de l'EI. Cependant, ici ce sont les textes des clients qui sont les cibles des méthodes de l'EI. L'objectif est soit la catégorisation de ces textes, soit le routage automatique des courriers. Les méthodes développées combinent en général des connaissances linguistiques (extraction de mots clés, termes) et des techniques statistiques pour la classification.

1.6.8 Publication d'offres d'emploi sur Internet

La publication d'offres d'emploi sur Internet a fait naître par son abondance le besoin d'un accès intelligent à ces documents. En pratique, cela équivaut à une indexation selon des descripteurs (clefs lexicales menant au document) plus riches.

Le projet SIRE (Sémantique, Internet, Recrutement et Emploi) développé par [75], a pour objectif de fabriquer des outils d'extraction automatique des termes les plus pertinents afin de faciliter l'étude statistique du marché du travail et la recherche d'emploi.

1.7 Conclusion

Nous avons présenté dans ce chapitre les principales notions de bases concernant le domaine de l'extraction d'information, ainsi que l'essentiel des méthodes et techniques utilisées afin de pouvoir présenter à l'utilisateur l'essentiel de l'information véhiculée par les documents textuels.

L'extraction d'information est à l'intersection de plusieurs domaines : économique, politique, juridique, social etc.

Dans le chapitre suivant, nous nous intéressons au domaine du profilage informatique, où le processus d'EI permet à partir d'un large volume d'information dynamique, d'extraire et de présenter les seules informations pertinentes (données personnelles, centres d'intérêts, préférences, ...), pour obtenir en résultat un profil utilisateur.

Chapitre 2

Profil Utilisateur

2.1 Introduction

La notion du profil utilisateur est apparue vers les années 80 avec les assistants et les agents d'interface, dû principalement au besoin de créer des applications personnalisées, capables de s'adapter à l'utilisateur, défini comme " une source de connaissance qui contient des acquisitions sur tous les aspects de l'utilisateur qui peuvent être utiles pour le comportement du système " [14].

Chacun des travaux traitant la question propose sa propre définition du profil utilisateur. Si tous s'accordent à dire qu'un profil doit représenter l'ensemble des centres d'intérêts, des préférences, des connaissances ou des habitudes de l'utilisateur, les concepts et les formalismes varient selon les systèmes ou les applications.

2.2 Définitions du Profil utilisateur

Définition 2.2.1. Par profil utilisateur, on désigne habituellement l'ensemble des informations permettant de personnaliser le fonctionnement du système (agent de recherche d'information ou autre) de façon à l'adapter à un utilisateur spécifique, soit pour fournir une meilleure qualité de services, soit pour améliorer la productivité de cet utilisateur. Pour ne pas obliger l'utilisateur à effectuer un effort d'adaptation permanent, cette personnalisation doit être persistante et ses effets doivent se reproduire de façon quasi identique à chaque utilisation du système par un même utilisateur (même si une évolution lente est acceptable voire souhaitable). En conséquence, le profil est généralement construit autour de données ayant une durée de validité relativement longue.

Définition 2.2.2. Un profil utilisateur est l'ensemble d'information et caractéristiques (nom, age, compétence,...), pouvant déduire la distinction entre deux utilisateurs.

2.3 Modélisation du profil utilisateur

Le modèle du profil consiste à spécifier sous quelle forme les données du profil doivent être représentées. Les modèles de représentations peuvent être simples basés sur des mots clés ou complexes basées sur des ontologies de domaines ou des hiérarchies de concepts. La construction du profil utilisateur consiste à collecter et exploiter les données et sources d'information pertinentes pour les représenter. La collecte de ces sources d'information intègre la spécification du type des données pertinentes à collecter, le mode d'acquisition des données (explicite ou implicite)[59].

2.3.1 Données personnelles

Pour que l'utilisateur d'un système personnalisable puisse en bénéficier, il est nécessaire qu'il soit identifié dans le système. Les données personnelles ont un double objectif, d'une part la gestion de l'identification de l'utilisateur (Nom, prénom,...) et d'autre part, elles permettent de catégoriser l'utilisateur en fonction des caractéristiques telles que les attributs d'authentification (Login, Mot de passe, etc.), les facteurs démographiques (Âge, Sexe, Première langue, Lieu de naissance, Particularités sociales et culturelles, etc.), les contacts personnels et professionnels et d'autres informations comme le groupe sanguin, le numéro du compte bancaire, etc. Ces données sont stables, rarement mises à jour et renseignées par l'utilisateur [18].

2.3.2 Domaine d'intérêt

Le domaine d'intérêt est l'un des éléments les plus présents dans un profil utilisateur, pour exprimer son domaine d'expertise ou son périmètre d'exploration. Il peut être défini par un ensemble de mots clés ou peut être vu comme une présélection virtuelle qui réduit la masse d'informations à prendre en compte. Ces domaines d'intérêts seront ensuite utiles pour filtrer, organiser et optimiser le comportement des systèmes [56],[59].

2.3.3 Préférences

Aussi comme le domaine d'intérêts, les préférences est l'un des éléments les plus importants dans un profil utilisateur. Tous les utilisateurs ne sont pas intéressés par les mêmes informations, ou par la même présentation de l'information. Les préférences permettent la personnalisation des comportements des systèmes envers les utilisateurs [18].

- Le choix des méthodes de personnalisation et de services offerts ;
- Le choix des composants graphiques à (ou ne pas) afficher ;
- Les préférences concernant la présentation des informations ;
- Les modalités d'exécution, décrivant le moment d'exécution d'une requête ;
- Le niveau de détails souhaités...etc.

2.3.4 Expérience et compétences

Selon [52], l'expérience de l'utilisateur représente son savoir-faire, la familiarité et l'aisance qu'il possède avec le type de système qui lui est présenté. Les compétences possédées par l'utilisateur correspondent aux connaissances qui ne relèvent ni du domaine, ni de l'expérience mais qui sont néanmoins considérées comme pertinentes dans le fonctionnement du système. La compétence est en relation avec d'autres concepts décrivant la capacité, l'habilité et le niveau d'expertise d'une personne.

2.3.5 Sécurité

Certains auteurs [18], considèrent que la sécurité fait partie du profil utilisateur. Selon eux, la sécurité peut concerner les données que l'on interroge ou modifie, les informations que l'on calcule, les requêtes utilisateurs elles-mêmes ou les autres éléments du profil. La sécurité des données peut être exprimée par des niveaux de sécurité predefines qui dépendent de la hiérarchie des vues autorisées.

2.4 Approches de représentation de profil utilisateur

La représentation de l'utilisateur à travers un profil permet de cibler ses besoins spécifiques et à prendre en considérations ses préférences à travers l'amélioration du comportement des systèmes informatiques. Un modèle de représentation de profil permet d'organiser ces éléments afin de faciliter leur exploitation par les applications. On distingue quatre principales approches de représentation de profil utilisateur : Historique, Ensembliste, Connexionniste, Multi-dimensionnelle et Conceptuelle.

2.4.1 Représentation par l'historique

Cette représentation est la plus répondue dans le domaine de la recherche d'informations. L'historique de recherche consiste en un ensemble des requêtes, des pages Web

visitées ou clickés ou des résumés textuels des résultats associés accumulés au cours des sessions de recherche de l'utilisateur.

Nous citons dans cette catégorie, les travaux des auteurs [87] qui représentent le profil utilisateur par son historique de recherche en se servant comme une base de données des requêtes soumises précédemment ainsi que leurs résultats associés. Cette base de données est utilisée pour sélectionner les requêtes les plus similaires à une requête en cours d'évaluation. Aussi l'approche proposée par l'article [13], où le profil utilisateur est représenté par l'historique des requêtes passées et l'historique des clicks sur les résultats sélectionnés par l'utilisateur.

2.4.2 Représentation ensembliste

La représentation ensembliste du profil utilisateur est basée sur un ensemble de mots clés (ou vecteurs de mots clés) pondérés représentés souvent selon le modèle vectoriel [34]. Ce type de représentation est le premier conçu pour modéliser le profil utilisateur.

Les paquets de termes traduisent les centres d'intérêts de l'utilisateur. Le poids d'un terme est souvent calculé selon le schéma TF*IDF communément utilisé en RI et représente le degré d'intérêt de l'utilisateur dans le profil. Dans [59], l'auteur divise la représentation ensembliste en trois sous-modèles de représentation :

- Un ensemble de termes pondérés où chaque terme représente un centre d'intérêt possible de l'utilisateur,
- Un vecteur de termes pondérés représentant un centre d'intérêt [37],
- Un ensemble des vecteurs de termes pondérés dont chacun représente un centre d'intérêt [43].

La figure 2.1 montre un exemple de profil utilisateur représenté par un ensemble de vecteurs de mots clés.

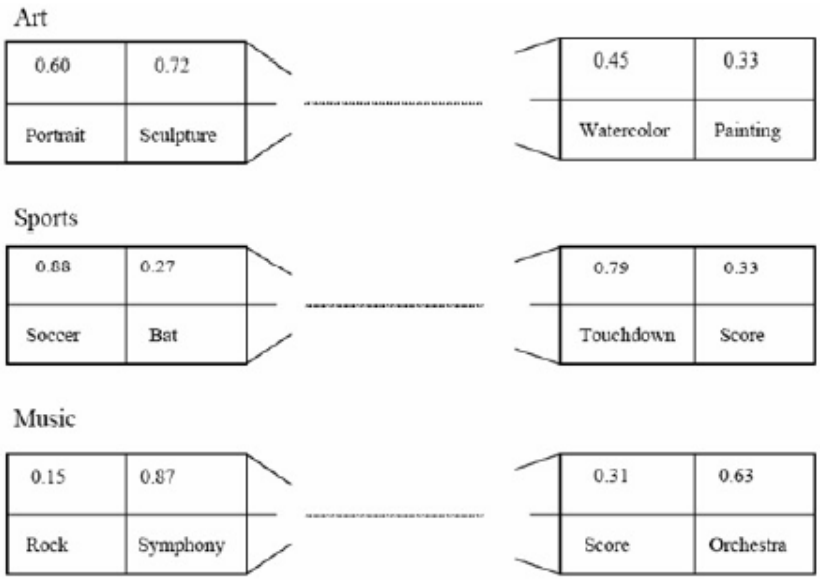


FIGURE 2.1 – Exemple de profil représenté par des vecteurs de mots clés [59].

La construction d'un profil ensembliste repose sur des techniques d'extraction des termes à partir des documents pertinents, jugés implicitement ou explicitement par l'utilisateur. Plusieurs systèmes d'accès personnalisé à l'information adoptent ce type de représentation. Tel est le cas des portails web tels que MyYahoo, InfoQuest, Anatagonomy qui est un système personnalisé de consultation de nouvelles et de journaux en ligne [38].

2.4.3 Représentation connexionniste

La représentation connexionniste du profil utilisateur consiste à représenter les centres d'intérêts de l'utilisateur par un réseau de noeuds pondérés dont chaque noeud représente un concept traduisant un centre d'intérêt de l'utilisateur. Cette représentation permet de résoudre les failles de la représentation ensembliste par la mise en place des relations de corrélation sémantiques entre les centres d'intérêts du profil. En effet, la richesse sémantique dans cette représentation permet de résoudre le problème de la polysémie des termes inhérents à la représentation ensembliste, l'incohérence possible entre les centres d'intérêts et l'identification d'un profil adéquat au sujet de la requête via les relations sémantiques [59].

Certains auteurs tels que [10] et [32], se sont intéressés à la représentation du profil utilisateur par un réseau de nœuds pondérés dans lequel chaque nœuds représente un concept traduisant un centre d'intérêt utilisateur. Ce type de représentation offre le double avantage de la structuration et de la représentation associative permettant de considérer

l'ensemble des aspects représentatifs du profil. Les concepts composant le profil sont souvent représentés par des relations de paires de nœuds.

L'auteur dans [69], suggère de créer les arcs reliant deux nœuds sur la base des co-occurrences entre ses termes.La figure 2.2 représente un extrait d'un profil utilisateur sémantique.

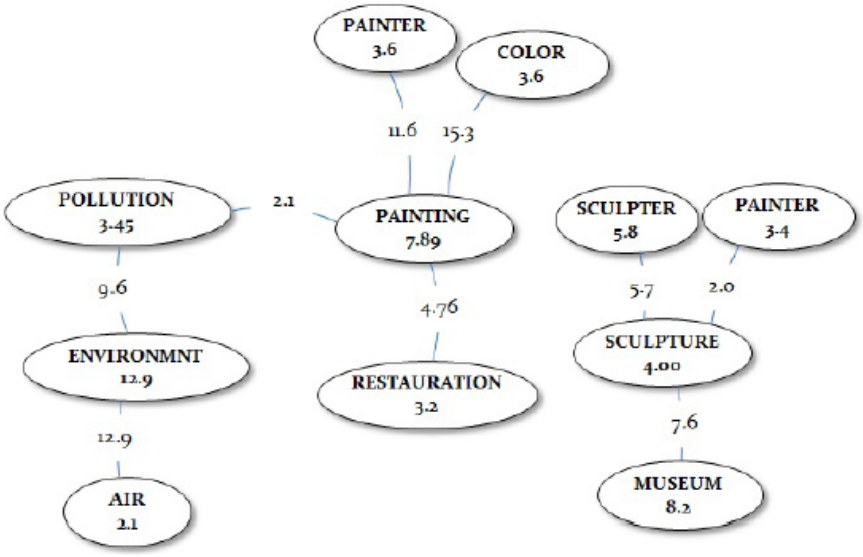


FIGURE 2.2 – Un extrait d'un profil utilisateur sémantique [69].

2.4.4 Représentation multidimensionnelle

Les auteurs dans [18, 29] ont adopté ce type de représentation pour le profil utilisateur. Il consiste en un ensemble de dimensions représentant chacune un aspect particulier (comme par exemple les données personnelles, le domaine d'intérêt). Son contenu est représenté par un modèle structuré de dimensions (ou catégories) pré-définies dans [29] ou bien par une structure générique [18] (2.3) où chaque dimension est constituée d'un ensemble d'attributs éventuellement organisés en sous dimensions offrant ainsi, la possibilité d'élargir l'ensemble des données représentées en fonction de domaine d'application et des besoins utilisateur. Les attributs peuvent être simples ou composés. Une sous dimension regroupe un ensemble d'attributs simples qui sont liés sémantiquement (par exemple l'adresse est composée du numéro de la rue, du nom de la rue, du code postal etc...).

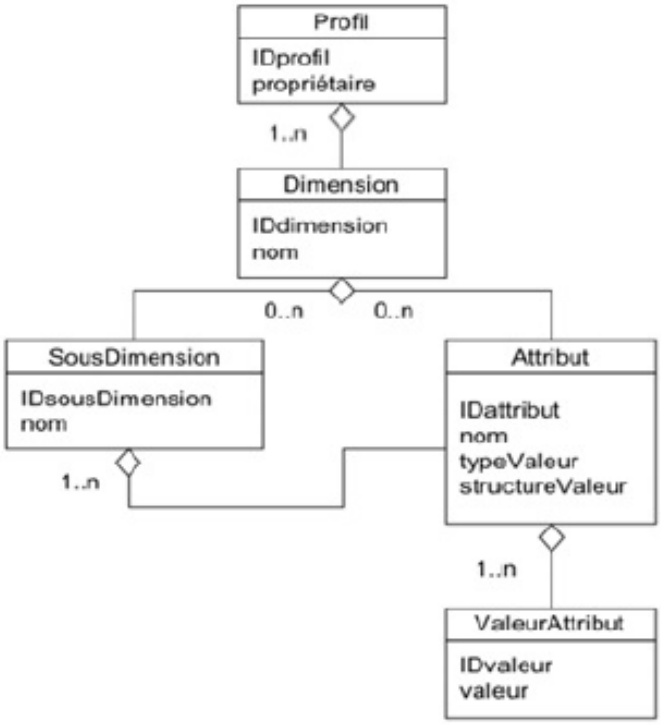


FIGURE 2.3 – Meta-modèle de profil utilisateur [18].

2.5 Construction du profil utilisateur

La construction du profil traduit un processus qui permet d'instancier sa représentation. Ce processus peut être explicite ou implicite. La construction explicite est basée sur une collection d'informations directement fournies par l'utilisateur via l'interface du système. La construction implicite, largement motivée par les travaux actuels dans le domaine, repose sur un procédé d'inference du contexte et préférences de l'utilisateur via son comportement lors de l'utilisation du système ou d'autres applications quotidiennes.

Les informations exploitées pour la construction sont généralement issues [55] :

- **Directement de l'utilisateur :**
  - Jugement explicite sur la pertinence des termes, documents,..
  - Définition de différents attributs : domaine d'intérêts, niveau, langue, ect...
  - Sélection des thèmes, sites favoris,...
- **Indirectement de l'application :**
  - Contenu des documents crées, consultés ,
  - Liens explorés,
  - Durée de lecture des documents,
  - Type d'application.

2.6 Evolution du profil utilisateur

La gestion de l'évolution du profil utilisateur est un processus complémentaire à la construction d'un profil utilisateur et désigne leur adaptation à la variation des centres d'intérêt des utilisateurs au cours du temps. L'évolution du profil utilisateur se fait souvent selon un processus incremental basé sur l'addition de nouvelles informations dans la représentation du profil.

La gestion de l'évolution du profil utilisateur consiste principalement à capturer les changements des centres d'intérêts de l'utilisateur dans une première phase et propager ces changements au niveau de la représentation du profil. Autrement dit, L'évolution de la représentation du profil utilisateur implique un changement des degrés d'intérêts dans certains domaines qui se traduit par une mise à jour de la structure/contenu des centres d'intérêts préalablement appris ou alors l'apparition d'un nouveau besoin en information qui se traduit par un ajout d'un nouveau centre d'intérêt au profil utilisateur [59].

2.7 Conclusion

L'approche commune à tous les systèmes d'accès à l'information, consiste en premier à modéliser le profil de l'utilisateur, puis à l'intégrer dans le processus d'accès à l'information.

Ce chapitre consacré au profil utilisateur, présente une large variété de technique de modélisation, comme nous avons fait ressortir les phases communes dans les différentes approches : la phase de représentation, la phase de construction (acquisition des données utilisateurs) ainsi que la phase d'évolution.

Le processus de construction consiste à organiser et extraire les éléments qui constituent le profil à partir des données de l'utilisateur collectées, en utilisant les différentes techniques d'extraction d'information, que nous allons détailler dans le chapitre suivant à travers l'étude de l'existant.

Chapitre 3

État de l'art

3.1 Introduction

La construction du profil traduit un processus qui permet d'instancier sa représentation. L'approche de construction dépend fortement de la représentation choisie pour le profil utilisateur (représentation par un (des) vecteur(s) de termes ou par des classes hiérarchiques ou pas). Cependant la démarche de construction commune à tous les systèmes est la suivante : on commence par collecter des informations sur l'utilisateur à partir de sources d'informations diverses, puis on applique des techniques et des algorithmes pour apprendre à partir de ces informations le profil classification de celles-ci.

3.2 Le domaine du résumé automatique

3.2.1 Résumé à base des métriques statistiques

3.2.1.1 Présentation

Les cadres d'application du résumé automatique sont multiples, et suivent la demande industrielle. Les systèmes de résumé automatique ont longtemps été focalisés sur les dépêches de presse, mais également sur les pages Web et les articles scientifiques.

Les techniques de production de résumé automatique ont évolué au fur et à mesure des problématiques que les chercheurs tentaient de résoudre. Ainsi, depuis les premiers travaux des années 50, les tâches du résumé automatique ont dérivé vers des problématiques plus complexes, satisfaisant les réelles attentes des utilisateurs.

Dans les travaux de [24], nous y trouvons les différentes étapes des approches statistiques de résumé. Ces étapes étaient à la base du développement du système de résumé automatique YACHS *(Yet Another Chemistry Summarizer)* et le système *CORTEX* qui

résulte de la combinaison de deux algorithmes : une méthode statistique de pondération des phrases couplées à une stratégie de sélection basée sur un algorithme de vote).

3.2.1.2 Principe

La méthode proposée par [24] est réalisé dans le domaine de la Chimie Organique, le document électronique à résumer, subit d'abord un pré-traitement qui consiste en la segmentation du document en phrases, la suppression de la ponctuation, la suppression de la casse (passage au minuscule), la lemmatisation des mots. Ensuite, vient la détermination des substances (En appliquant les différents critères des substances chimiques ainsi chaque mot qui sera nom d'une substance aura un score), une fois déterminées, le document sera traité sous forme d'une matrice.

$$M = [ax, y]x = 1...m; y = 1...n; \tag{3.1}$$

où m est le nombre de phrases du document et n le nombre de termes différents. Dans cette interprétation, chaque ligne de M est un vecteur (V_x) correspondant à la phrase x du document où chacune des composantes de ce vecteur est la fréquence d'un terme dans la phrase. C'est à partir de cette matrice que sont calculées des métriques (la somme des fréquences des mots d'une phrase, somme des poids des mots d'une phrase, la position de la phrase dans le document, similarité avec le titre, l'interaction entre les phrases. . .etc.) qui une fois assemblées vont permettre d'attribuer un score de pertinence à chaque phrase du document. Les phrases ayant les scores les plus élevées seront sélectionnées pour le résumé (le nombre de phrase dépend de la taille du résumé demandé).

Les méthodes de résumé automatique à base de graphe ont récemment bénéficié des avancées de la recherche dans l'étude des réseaux sociaux. En effet, si l'on considère un corpus comme un réseau social où les différents acteurs du réseau sont les phrases du corpus, extraire les phrases centrales revient à extraire d'un réseau social les nœuds les plus influents, donc ceux qui entretiennent le plus de liens avec les autres. Ce type d'analyse permet de rendre compte de la centralité d'une phrase en tirant parti du contenu global des documents, et non plus uniquement de la requête et de listes de termes, forcément limitatifs.

Dans [2], l'auteur présente quelques autres méthodes de résumé automatique, et parmi l'algorithme *LexRank*, développé par Radev [31], qui se base sur l'observation citée dans la figure 3.1. Le but du système est d'extraire les phrases considérées comme " centrales " dans un corpus, en se basant sur la notion de " prestige " des réseaux sociaux. Un

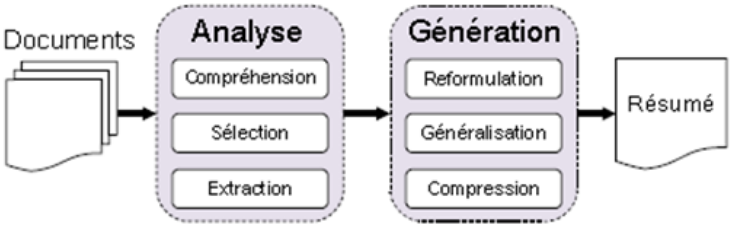


FIGURE 3.1 – Méthodologie de production de résumé [24].

graphe du corpus est établi, chaque nd étant une phrase et chaque arête recevant comme poids la similarité des deux phrases qu'elle lie. Cette similarité est calculée à l'aide de la mesure *cosinus*. Cette mesure établit une similarité entre deux vecteurs en calculant le cosinus de l'angle de ces deux vecteurs. Pour calculer la similarité entre phrases, il faut établir un vecteur pour chaque phrase. Le vecteur représentant une phrase est généralement rempli avec les *tf.idf* des constituants de cette dernière. Les phrases sélectionnées sont celles considérées comme centrales dans ce graphe : les phrases qui ont les plus fortes composantes.

3.2.1.3 Avantages

✓ L'utilisation de la technique de *tf.idf* qui a déjà fait ses preuves dans le domaine de l'extraction de l'information.

3.2.1.4 Inconvénients

– La détermination d'un seuil peut causer une perte d'informations pertinentes.

3.2.2 Résumé à base des graphes

3.2.2.1 Présentation

Le résumé est une technique de fouille de texte qui consiste à compresser un texte tout en répondant aux besoins informationnels de l'utilisateur, le résumé est un problème d'optimisation, plusieurs méthodes ont étaient proposé, nous y trouvons dans ce cadre les travaux de [47], qui se sont basés sur les méthodes statistiques afin d'en proposer un algorithme glouton.

L'objectif de la méthode proposée est de concevoir un système générique pour les résumés automatiques afin de pouvoir s'adapter à un document de trois langues (français, anglais et espagnol).

3.2.2.2 Principe

Dans l'approche [47], les auteurs utilisent un modèle graphique, tel exploités dans les travaux de [31], qui considère le résumé comme l'identification des segments les plus prestigieux dans un graphe, tel les algorithmes de PageRank [79], dans les réseaux sociaux, qui décide de l'importance d'un sommet non pas de son contenu mais de son emplacement, appliqué au résumé, cela signifie que le document sera représenté sous forme d'un graphe d'unités textuelles (phrases) liées entre elles par des relations issues de calculs de similarité. Les phrases sont ensuite sélectionnées par le critère de centralité dans le graphe puis assemblées pour construire des extraits. Il est à noter que les méthodes de classement dépendent de la bonne construction du graphe sensé représenter le document, puisque le graphe est généré à partir des mesures de similarité inter-phrases.

Dans les travaux de [77], [31], les auteurs ont utilisé une représentation vectorielle de façon à avoir chaque phrase sous forme d'un vecteur de taille N (un vecteur de N mots) où V_i(j) représente le poids *tf*idf* du mot j dans la phrase i combiné aux valeurs de similarités entre phrases calculés avec la mesure cosinus. Le point faible de toute méthode utilisant les mots comme unité est qu'elles sont tributaires du vocabulaire, ce qui cause la chute de la mesure cosinus dans le cas de deux mots morphologiquement différents. Dans [24], l'auteur remédie à ce problème en proposant une mesure dérivée d'un calcul de similarité entre chaînes de caractères pour la détection des entités redondantes, cette mesure permet d'avoir des relations entre des segments ne partageant aucun mot. Les auteurs [26] ont montré que la mesure mixte entre les mots et les caractères améliore l'extraction des segments.

Les auteurs, dans [47], exposent le problème du résumé automatique comme étant un problème d'optimisation. Ainsi, le texte est représenté sous forme d'un graphe non orienté qui peut être assimilé comme un problème de coloration ou à une variante de celui du voyage du commerce. Le problème ainsi posé est de l'ordre de P !, étant P le nombre de phrases d'un document, cela fait un problème NP-complet, ce pour, les auteurs se sont tournés vers les approches gloutonnes. En s'inspirant de l'algorithme de Kruskal, les auteurs ont développés l'algorithme REG (REsumeur à base de Graphe) qui réalise l'extraction des m phrases les plus pertinentes qui constitueront le résumé.

L'algorithme REG du résumé automatique comprend trois modules :

- **Le premier module :** réalise la transformation vectorielle du texte avec des processus de filtrages classiques, de lemmatisation et de normalisation, afin de réduire la dimensionnalité du texte. Une représentation en sac de mots produit une matrice S de P phrases et N mots S[ij]= TF_i.

- **Le second module :** applique l'algorithme glouton et réalise le calcul de la matrice d'adjacence, pour avoir une pondération de la phrase v directement de l'algorithme, ainsi les phrases pertinentes sont sélectionnées comme ayant les plus grandes pondérations.

✍ ***Solution Gloutonne :***

A partir de la matrice, les auteurs créent un graphe G(S, A), où A est l'ensemble des arrêtes (il existe une arrête entre phrase (i) et une phrase (j) s'il existe au moins un mot en commun entre les deux).

Pour afficher les phrases les plus lourdes, il faut chercher une variante du problème de l'arbre de poids maximum, où les poids sont sur les sommets, non pas sur les arrêtes, pour ensuite appliquer une recherche gloutonne dont les étapes sont :

1. Choisir le sommet le plus lourd X_0, et le mettre dans T. Il sera appelé racine.
2. La racine sera choisie parmi les nœuds dont le degré est supérieur ou égal à deux.
3. Ajouter à T le voisin de X_0 le plus lourd. Il sera choisi parmi ceux qui ne faisant pas partie de T.
4. Répéter 2 jusqu'à en avoir les K sommets requis.
5. La sortie sera le chemin T.

- **Le troisième module :** génére les résumés par l'affichage et concatenation des phrases pertinentes. Le premier et le dernier module reposent sur le système Cortex [49], [25], qui effectue une extraction non supervisée de phrases pertinentes en utilisant plusieurs métriques pilotées par un algorithme de décision.

3.2.2.3 Avantages

✓ L'algorithme est considéré efficace pour l'extraction des segments pertinents.

✓ Indépendance par rapport aux sujets abordée.

3.2.2.4 Inconvénients

– L'inconvénient de l'approche proposée est qu'elle limite à l'avance le nombre de phrases à obtenir dans le résumé.

3.3 Le domaine de la Recherche d'Information

3.3.1 Construction d'un profil basé sur l'interaction entre dimensions

3.3.1.1 Présentation

L'approche [56], porte sur l'apprentissage d'un profil utilisateur qui reflète ses centres d'intérêts à long terme. De manière sommaire, le profil utilisateur est représenté selon deux dimensions : l'historique de ses interactions et l'ensemble de ses centres d'intérêts à un certain instant. A l'instant s , le profil utilisateur est représenté par $U = (H^s, I^s)$, où H^s représente l'historique des interactions de l'utilisateur avec le SRI (Système de recherche d'information) jusqu'à l'instant s et I^s représente la bibliothèque de ses centres d'intérêt inférés jusqu'à l'instant s. Le procédé de construction du profil consiste en un cycle comportant deux étapes. La première consiste à représenter puis faire évoluer l'historique des interactions de l'utilisateur avec le SRI par agrégation de l'information issue des sessions de recherche successives dont le but d'inférer les contextes d'usages décrits par des mots clés pondérés. La seconde étape consiste à construire puis faire évoluer les centres d'intérêts de l'utilisateur sur la base de l'historique d'interactions. L'évolution est basée sur une mesure de corrélation de rangs qui évalue le degré de changement des centres d'intérêts durant une certaine période de recherche.

3.3.1.2 Principe

Soit q_s la requête soumise par un utilisateur U la session de recherche S^s se déroulant à l'instant s et D^s l'ensemble des documents pertinents pour l'utilisateur durant cette session. Un document est considéré comme pertinent s'il a été ainsi jugé par l'utilisateur de manière explicite ou implicite. Soit R_u^s l'ensemble des documents déjà visités et jugés pertinents par l'utilisateur lors des sessions de recherche passées depuis l'instant s_0 .La méthode propose l'utilisation de matrices pour la représentation d'une session de recherche et de l'historique des interactions. La session de recherche S^s est représentée par une matrice Document-Terme D^sXT^s où T^sest l'ensemble des termes qui indexent les documents de D^s(T^s est une partie de l'ensemble des termes représentatifs des documents préalablement jugés pertinents noté $[R_u^s]$.

Chaque ligne de la matrice S^s représente un document $d \in D^s$, chaque colonne représente un terme $t \in T^s$. Dans le but d'améliorer la précision de la représentation Document-Terme, la méthode propose d'introduire dans le schéma de pondération

terme-document un facteur qui reflète la pertinence relative d'un terme compte tenu des jugements de pertinence que l'utilisateur a émis.

Le coefficient de pertinence d'un terme t dans un document d à l'instant s noté $s(t,d)$ est défini comme suit :

$$CPT^s(t,d) = \frac{w_{td}}{l(d)} * \sum_{t' \neq t, t' \in D^s} cooc(t,t') \tag{3.2}$$

- w_{td} : est le poids du terme t dans le document d calculé selon le schéma classique *tf * idf* ,

- **l(d)** : est la longueur du document d,
  ntt' est la proportion de documents contenus dans R_u^s contenant t et t', nt est la proportion de documents contenus dans R_u^s. contenant t . $S^s(d,t)$est ainsi construite :

$$S^s(d,t) = CPT^s(t,d). \tag{3.3}$$

L'historique des interactions de l'utilisateur est représenté par une matrice noté H^s de dimension R_u^s * TR_u^s),construite de manière incrémentale par agrégation, les informations issues de la matrice Ss en utilisant un opérateur d'agrégation qui combine pour chaque terme son poids classique dans le document calculé selon le schéma *tf * idf* et ses poids atténués par les coefficients de pertinence calculés lors des sessions de recherche passées . Plus précisément l'opérateur d'agrégation est défini comme suit :

$$H^0(d,t) = S^0(d,t). \tag{3.4}$$

Un contexte d'usage est ainsi un vecteur de termes K^s extrait à partir de l'historique des interactions en sommant chaque colonne de la matrice associée. Le poids d'un terme est calculé comme suit :

$$K^s(t) = \sum_{d \in R_u^s} H^s(d,t) \tag{3.5}$$

La maintenance du profil utilisateur est ainsi basée sur la mesure de la corrélation de rangs de termes entre deux contextes d'usages successifs. Ensuite, l'historique des interactions n'est cumulé que si les sessions de recherche sont liées à un même domaine d'intérêt de l'utilisateur, et K^s représente ainsi un cumul de l'ensemble de ces sessions reflétant un seul centre d'intérêt de l'utilisateur.

3.3.1.3 Avantages

✓ Les centres d'intérêt ainsi construits peuvent être réutiliser dans différentes étapes d'un processus d'accès personnalisé à l'information. Plus précisément, la bibliothèque Is constitue alors une ressource pour :

- La réécriture de la requête.
- La mise en œuvre de l'appariement requête-document.
- L'ordonnancement des résultats de recherche.

3.3.1.4 Inconvénients

– La variation des centres d'intérêt de l'utilisateur, décelé à travers les requêtes que l'utilisateur a émis, ne présentent pas forcément des régularités prévisibles.

– La méthode statistique proposée serait confrontée à un risque d'erreur difficilement mesurable.

– La méthode présentée ne permet pas de couvrir une représentation sémantique des centres d'intérêts. En effet, les centres d'intérêts de l'utilisateur sont représentés selon des vecteurs de termes pondérés n'ayant aucune correspondance avec les concepts associés.

– Cette représentation a un impact direct sur la procédure de maintenance du profil utilisateur de l'approche de base. En effet, la détection d'un éventuel changement des centres d'intérêts entre les sessions de recherche est basée sur une mesure de corrélation de rangs des termes entre des contextes d'usages successifs.

3.3.2 Profils utilisateurs à base d'ontologie

3.3.2.1 Présentation

Dans l'article [60], les auteurs ont présenté une extension d'une approche de construction implicite du profil utilisateur développée dans [56], et dans le but de remédier aux limites liées à l'approche, ils ont étendu le processus de construction du profil utilisateur dans l'approche de base afin d'obtenir un profil à base d'une ontologie. La nouvelle définition du profil est basée sur une représentation sémantique des centres d'intérêts de l'utilisateur.

Notation

– K^s : un contexte d'usage associé à la session de recherche S^s.

– C^s :les catégories sémantiques de l'ontologie ainsi extraites forment un vecteur de concepts pondérés noté C^s représentant sémantiquement le centre d'intérêt de l'utilisateur lors de la session de recherche S^s.

3.3.2.2 Principe

L'objectif principal est de construire dans un premier temps un profil utilisateur à base d'une ontologie où les centres d'intérêts sont représentés selon des vecteurs de concepts pondérés de l'ontologie de l'ODP (ontologie de référence permettant de représenter sémantiquement les centres d'intérêts de l'utilisateur.), puis intégrer ce profil dans un processus de RI personnalisé.

Premièrement, l'objectif est de représenter chaque catégorie sémantique de l'ODP selon le modèle vectoriel servant ainsi ultérieurement à la classification sémantique des contextes d'usage qui leur correspond. En effet, afin de mettre en place une telle classification précise, l'auteur a choisi de représenter chaque catégorie en utilisant les données d'apprentissage, soit les 60 premiers titres et descriptions des liens url associés. L'étude dans [20] a montré que l'utilisation des titres et des descriptions composés manuellement dans le répertoire du web "Looksmart" permet d'achever une précision de classification plus élevée que l'utilisation du contenu des pages. Pour cela, la procédure suivie est comme suite :

1. Concatener les titres et descriptions des 60 premiers liens url associées à chacune des catégories de l'ODP dans un super-document sd_j formant ainsi une collection de super-documents, un par catégorie.
2. Lemmatiser (analyse lexical) les super-documents à l'aide de l'algorithme de porter (algorithme de normalisation des mots).
3. Représenter chaque super-document noté sd_j par un vecteur V_j selon le modèle vectoriel où le poids W_{ij} du terme t_i dans le super-document sd_j est calculé comme suit :

$$W_{ij} = P_{ij} * \log \frac{N}{N_i} \tag{3.6}$$

– P_{ij} :le degré de représentativité du terme ti dans le super document sd_j .
– **N** :le nombre de super-documents de la collection.
– N_i :le nombre de super documents contenant le terme t_i.

Le degré de représentativité du terme dans le super-document est égal à la moyenne de la fréquence du terme dans ce super-document et sa fréquence dans les super documents

fils. Chaque catégorie de l'ODP C_j est représentée selon le modèle vectoriel par le vecteur V_j .

Après avoir représenté chaque catégorie sémantique de l'ODP selon le modèle vectoriel, l'étape suivante est l'application d'une méthode de classification supervisée des contextes d'usage selon l'ontologie de référence utilisée. La classification est basée sur une mesure de similarité vectorielle entre le vecteur représentatif V_j d'une catégorie C_j de l'ODP et celui du contexte d'usage K^s . Le contexte d'usage sera classé dans les n premières catégories ayant la similarité vectorielle la plus élevée avec son vecteur représentatif. Le poids $p(C_j)$ d'une catégorie C_j représenté par son vecteur V_j est donné selon la formule suivante :

$$P(C_j) = sim(V_j, K_s) = \sum t_{ij} * t_{ik} \tag{3.7}$$

- t_{ij} : poids du terme t_i dans le vecteur représentatif de la catégorie V_j.
- t_{ik} : poids du terme t_i dans le vecteur représentatif du contexte d'usage K^s.

Enfin, on obtiendra ainsi un vecteur ordonné des catégories sémantiques pondérées de l'ontologie de l'ODP noté C_s. C_s est la représentation sémantique du centre d'intérêt qu'on appelle vecteur contextuel représenté comme suit :

$$C^s = (p(C_1), p(C_2), :: p(C_i) ::, p(C_n)) \quad telque \quad p(C_j) = sim(V_j, K^s). \tag{3.8}$$

3.3.2.3 Avantages

✓ L'un des objectifs de la méthode proposée est d'évaluer l'impact du niveau de profondeur des catégories descriptives des centres d'intérêts de l'utilisateur sur la qualité du profil construit ainsi que sur les performances de recherche.

✓ La modélisation du profil utilisateur est basée sur l'exploitation de son historique de recherche et d'une ontologie de domaines Web permettant de dégager une représentation sémantique de ses centres d'intérêt.

✓ L'utilisation de l'historique de utilisateur permet d'évoluer son profil à court et à long terme.

3.3.2.4 Inconvénients

– La précision du profil construit à court terme dépend du nombre de sessions de recherche effectuées par l'utilisateur.

3.3.3 Conversation électronique

3.3.3.1 Présentation

La recherche des intérêts des utilisateurs est devenue une question pré-occupante dans la personnalisation du contenu de l'information. Généralement, les gens tendent à parler des sujets par auxquels ils sont intéressés. En fait, les conversations électroniques sont l'une des sources les plus riches pour appliquer l'analyse sémantique. Le développement courant des outils sociaux pour l'interaction numérique fait une sorte d'informations disponibles sur le Web. A travers un système automatique de traitement de textes, des sujets appropriés ont pu être extraits à partir des conversations textuelles afin de créer des profils d'utilisateur avec l'information d'intérêt.

Dans l'article [62], les auteurs visent des textes de sources dynamiques afin d'extraire des informations d'intérêt sur les utilisateurs. Les documents statiques ne reflètent pas habituellement l'intérêt courant d'utilisateur comme le fait d'une source dynamique. En outre, dans ce travail, les auteurs étudient l'utilisation des techniques de filtrage de données, puisqu'ils jouent un rôle important dans l'analyse en travaillant avec des textes de sources non-structurées (chat, forum,...).

Un des travaux les plus appropriés qu'ils ont analysés est *" Wikify !"* [6] et sa prolongation. *" Wikify !"* est un algorithme qui lie des concepts de Wikipédia aux documents utilisant une approche de détection de mot-clé basée sur les titres d'article de Wikipédia.

Dans ce contexte, les auteurs proposent une approche originale appelée TopText. Elle se compose d'une méthode pour la détection des sujets à partir des textes non structurés (chat,forum,..). L'idée principale est d'associer des articles de Wikipedia, considérés comme concepts de la connaissance humaine, aux messages textuels afin d'impliquer les sujets d'une conversation textuelle, comme ils proposent deux stratégies pour l'association des concepts aux messages d'utilisateur : (i) utilisation d'un texte brut des messages pour une recherche dans le dictionnaire de concept, et (ii) identification des entités à partir des messages précédents pour la recherche des concepts.

3.3.3.2 Principe

Le schéma général de la méthode proposée par [62] est présenté par la figure 3.2. Les entrées du système sont principalement des textes de conversations électroniques. Dans la première étape du processus, l'entrée est analysée dans l'ordre pour identifier les utilisateurs impliqués et leurs messages. En conséquence, des messages sont groupés par des utilisateurs. Deuxièmement, les textes sont prétraités en utilisant des techniques de filtrage de données afin de préparer les messages d'utilisateur pour la future analyse.

La troisième étape est l'association de concept, pour ceci, ils emploient un dictionnaire sémantique contenant des concepts de la connaissance humaine. Le résultat de cette étape est un ensemble de concepts associés à chaque message. Cette information donne une idée générale de la signification sémantique des messages.

En conclusion, une hiérarchie de catégorie est établie pour chaque concept. Les informations sur les catégories sont également extraites à partir du dictionnaire. D'abord, ils relient chaque concept à une catégorie correspondante au premier niveau. Après, le processus est répété pour des catégories de plus haut niveau.

Le résultat final du processus est un ensemble de profils d'utilisateur, chacun d'eux contient (i) un rang des concepts les plus appropriés, (ii) un rang des premières catégories de niveau les plus appropriées et (iii) une liste des catégories générales les plus significatives de n'importe quel niveau. Puisque ce profil est basé sur les sujets qui sont régulièrement mentionnées par l'utilisateur, l'information peut être considérée comme un intérêt d'utilisateur.

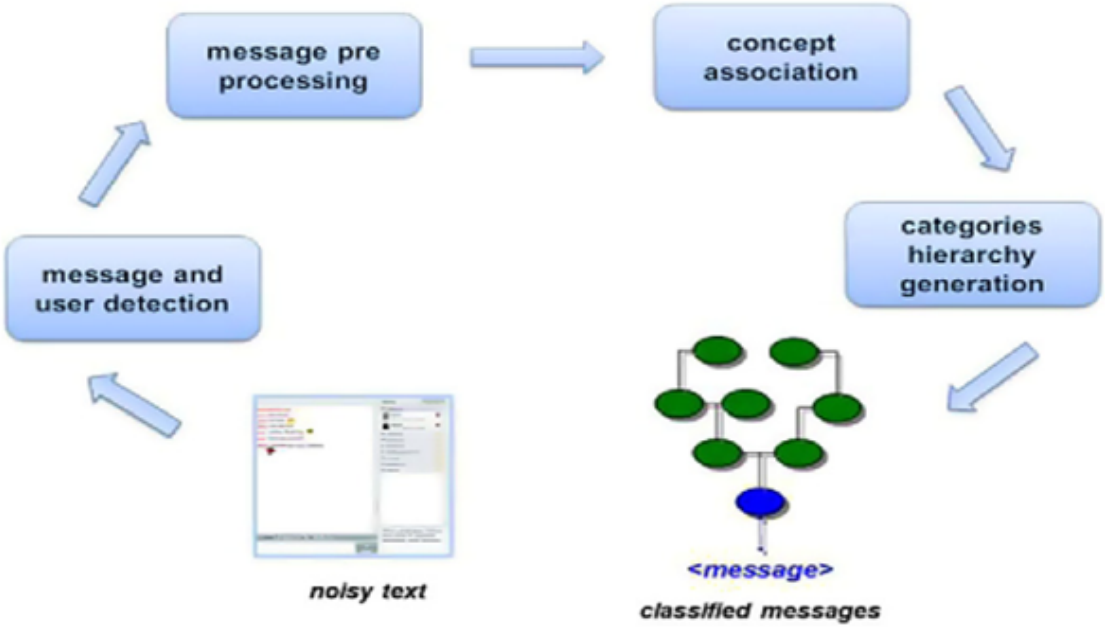


FIGURE 3.2 – Vue de la méthode proposée [62].

La structure générale de la méthode proposée, doit être spécifiée avec des opérations et des techniques concrètes dans chaque étape. La première étape demeure non modifiée, puisque l'objectif est de détecter les utilisateurs et les messages.

Pour la deuxième étape, la méthode définie la stratégie de prétraitement dans le but de traiter des textes de chat. Ceci comporte (i) la suppression des références aux

utilisateurs par leurs noms, (ii) le filtrage des caractères inadmissibles et (iii) l'exécution des opérations d'analyseur.

En ce qui concerne la troisième étape, la méthode définie une stratégie spécifique pour l'association de concept. Dans ce cas, les auteurs ont employé l'index de concept qui établi des articles de Wikipedia. En outre, la position dans l'ensemble de résultat est employée pour définir la valeur de pertinence du concept au message d'utilisateur.

En conclusion, les deux index de catégorie sont employés pour la génération de hiérarchie de catégories. Au commencement, les auteurs ont associé les premières catégories de niveau aux concepts, et puis, ils ont établi des relations avec les catégories de plus haut niveau afin d'établir la structure hiérarchisée.

Comme les catégories sont trés généralisées et la durée de calcul se développe exponentiellement, ils ont décidé de limiter la profondeur d'arbre de hiérarchie à trois niveaux, comme illustrer dans la figure 3.3.

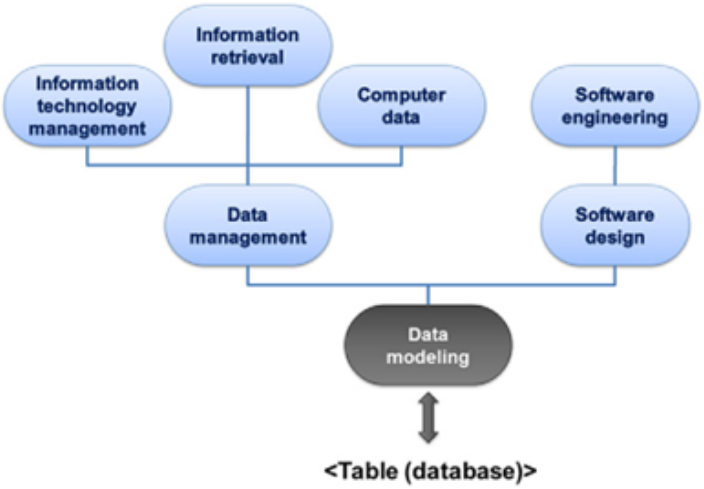


FIGURE 3.3 – Une hiérarchie de catégorie d'échantillon pour le Tableau de concept (base de données)[62].

3.3.3.3 Avantages

✓ Dans ce travail, les auteurs ont présenté une autre idée originale, qui est l'utilisation des comptes-rendus dans l'assortiment des concepts de Wikipedia, alors que tous les travaux relatifs emploient juste le titre des articles. De cette façon, l'association des concepts est reliée non seulement au même nom, mais également aux concepts fortement connexes qui sont mentionnés dans les articles soustraits.

✓ La précision du profil construit à court terme dépend du nombre de sessions de recherche effectuées par l'utilisateur.

✓ La plupart des travaux relatifs visent à identifier des sujets de documents statiques, alors que dans cette méthode vise des textes de source dynamique.

✓ Il y'a aucune précédente recherche sur l'identification de sujets dans des textes de notation chat en utilisant Wikipédia comme source de connaissance.

3.3.3.4 Inconvénients

– L'utilisation des mots-clés nous fait perdre l'information concernant le contexte de l'utilisation des termes, car un concept ne peut être décrit par un seul mot.

3.4 Le domaine de l'Extraction d'Information

3.4.1 E-Recrutement : Traitement des offres d'emplois

3.4.1.1 Présentation

La multiplication du nombre des moyens de recrutement sur Internet a rendu de plus en plus importante l'analyse de la performance des annonces d'emploi et son optimisation. Notamment, celle-ci peut être évaluée à travers le volume de candidatures reçues suite à la publication d'une annonce. Suite à la pertinence des titres des annonces de recrutements ainsi qu'à l'importance du volume d'information qu'ils véhiculent, des méthodes ont étaient développées afin de repérer les mots clés.

3.4.1.2 Principe

Dans [50], les auteurs proposent une méthode d'extraction des mots-clés du titre d'un corpus d'offre d'emploi, ainsi qu'un modèle pour tester l'effet de leur présence sur le rendement. Cette méthode, illustrée par une analyse exploratoire préliminaire, repose sur le codage en indicatrices des mots-clés repérés grâce à l'étude des fréquences d'apparition et des spécificités lexicales associées aux différentes catégories de fonctions des offres d'emploi.

Le pré-traitement consiste à exclure les annonces de test, exclure les annonces rédigées en anglais, la sélection des offres faisant référence à des postes à pourvoir dans les fonctions : Commercial-Vente, Gestion-Comptabilité-Finance, Marketing et Systèmes d'information-Telecom.

L'approche s'intéresse aux titres des annonces qui ont la particularité d'être court et peu chargé (pas de verbes, pas d'expressions courantes), contenant quelques mots outils (de, d', et...) qui seront filtré, car ne présentent pas d'intérêt pour la problématique.

Le texte est donc écrit en minuscule et les accents éliminés. Pour l'extraction des mots clé, ils choisissent tous les mots qui font référence aux qualifications spécifiques à chacune des fonctions étudiée, ç-a-d qu'ils peuvent appartenir aux profils lexicaux des différentes catégories d'offres d'emploi à l'aide d'une analyse de correspondance, qui sera complétée par le calcul des formes spécifiques à l'aide d'un modèle probabiliste. Pour chaque fonction étudiée, les auteurs calculent les spécificités positives, et les formes ainsi obtenues détermineront un premier ensemble de mots-clés. Ensuite, ils relèvent les formes de plus hautes fréquences parmi celles n'étant pas des spécificités lexicales afin d'obtenir

des mots-clés transversaux à toutes les fonctions. Tant qu'à l'évaluation de ces mots-clés sera faite par l'algorithme de CART.

3.4.1.3 Avantages

✓ L'amélioration du rendement aux annonces d'offre d'emploi.

✓ S'intéresser au titre, minimise le temps du traitement, car généralement le titre est un résumé représentatif (pertinent) du corpus.

3.4.1.4 Inconvénients

– Deux annonces ayant le même titre ne peuvent pas être toute deux acceptés alors qu'elles font référence à deux poste de travailles différents, donc le critère du poste recherché doit être pris en considération.

– Les compétences des utilisateurs ne sont pas prises en compte.

3.4.2 E-Recrutement : Application polonaise

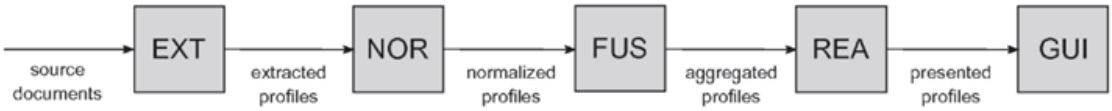
3.4.2.1 Présentation

À l'heure actuelle, l'exigence de la plupart des processus de recrutement est de fournir un curriculum vitae sous forme d'un fichier informatique, qui est ensuite traité par un flux de travail des systèmes internes, comme il faut considérer le développement d'activités sociales sur le Web qui se traduisent par la création de profils professionnels .

Dans l'article [84], le but du projet d'eXtraSpec est de créer les outils qui peuvent aider à automatiser le processus de recrutement et la conclusion de l'expert. L'intérêt spécial est l'analyse et l'extraction d'information des sources d'enchaînement, particulièrement disponibles dans la langue polonaise. Un fait important est que le système est consacré aux documents de processus écrits dans la langue polonaise, qui est une tâche dure due à la complexité de la grammaire polonaise. Ces documents sont traités par le composant d'extraction (EXT.), qui a pour rôle de valider, et créer plus tard les profils extraits qui stockent toutes les informations pertinentes du point de vue du recruteur.

La circulation des documents au sein de l'architecture d'eXtraSpec est présentée dans la figure suivante :

Les auteurs dans [84], ont présenté un composant d'extraction appelé EXT, étant une partie du projet eXtraSpec. EXT est capable de traiter le navigateur des pages écrites dans la langue polonaise afin d'en extraire les informations pertinentes pour les besoins de



constatation d'experts. Pour remplir ces tâches, EXT utilise un algorithme d'extraction de contenu, ce qui est fait conformément à la hiérarchie d'un document HTML. Cela signifie que la structure de la page Web se traduit par un profil extrait et le contenu des champs spécifiques est transformé par XPath ou Regex et se sont installés dans un champ correspondant.

Dans le volet d'extraction décrite dans le système eXtraSpec, les auteurs [84], ont concentré sur le traitement des documents semi-structurés (documents HTML). La principale tâche de cette composante est d'extraire les attributs concrets du profil des documents sources. Pour ce faire, ils ont crée un arbre de règles d'extraction, dans lequel chaque règle est chargé d'extraire les éléments du profil. Le composant EXT exécute une transformation d'arbre de HTML(semi-structuré) à une autre structure arborescente (XML).

3.4.2.2 Principe

Afin de permettre l'utilisation de la méthode d'extraction pour différentes sources d'enchaînement, un algorithme hiérarchique d'extraction a été proposé pour opérer l'extraction des règles représentatives comme expressions de XPath et expression régulière pour les transformations des chaînes avancées.

Le profil extrait a une forme d'un arbre structuré et sa représentation nécessite l'utilisation XML.

Les classes d'information suivantes ont été prises en compte :

- Base de données personnelles (nom, prénom, date de naissance, adresse, e-mail...).
- Mentions : cette catégorie contient une liste de références à des sites en différents documents, où la personne a été mentionnée.
- L'Education, Certificats, Compétences, Expérience de travail, Adhésion.

À l'exception des catégories mentionnées ci-dessus, certaines meta-données sur le profil sont stockées : Numéro d'identification, Date à laquelle il a été créé et Référence au document original. L'algorithme d'extraction fonctionne en pages Web HTML. L'approche d'extraction qui a été choisie est fondée sur deux hypothèses : la première, que la cible et la source de données est un arbre, et, deuxièmement, que la source des données a une structure fixe qui suit l'approche d'un arbre.
Une structure fixe veut dire que :

- Toutes les catégories des nds de l'arbre sont fixes et connus.
- Les relations entre les nds ne changent pas.
- La recursion n'est pas autorisée.

L'approche d'extraction de l'information choisie est basée sur la hiérarchie des règles d'extraction.

Dans chaque hiérarchie des règles, il y'a une règle racine unique, qui a pour but d'extraire la partie de l'arbre du document HTML qui contient l'ensemble des informations nécessaires pour créer un profil. Cette partie, peut être encore traitée par les règles de fils pour extraire des informations plus détaillées.

Dans chaque règle de l'hiérarchie, il y'a une seule règle racine, qui extrait une partie de l'arbre du document HTML qui inclue les informations nécessaires pour construire le profile.

Chaque règle d'extraction peut pointer vers un élément cible de l'extrait du profil. Il peut y avoir des règles qui ne pointent pas vers un élément de profil, ils sont utilisés comme regroupant "facilités" pour les règles de fils.

Chaque règle d'extraction indique, si l'élément vers lequel il pointe est atomique (un résultat attendu est un élément simple feuille - par exemple d'abord nom), le composé (élément cible peut être un élément complexe comme l'éducation entrée), ou une collection d'éléments de même type.

Chaque règle a une condition, que son résultat de l'extraction doit englober toutes les informations nécessaires pour remplir l'objectif de la règle (même si elle peut être faite par les règles de fils). Cela limite les règles du fils pour en extraire une partie seulement de sous-arbre extrait par la règle du parent.

Le schéma XML de la règle d'extraction est présenté dans la figure 3.4 :

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="http://extraspec.org/schema/ExtractionRulesSchema"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="extractionRules">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="extractionRule">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="language" type="xs:string" use="required" />
<xs:attribute name="parentExtractionRuleId" type="xs:string" use="required" />
<xs:attribute name="id" type="xs:unsignedByte" use="required" />
<xs:attribute name="relatedElement" type="xs:string" use="required" />
<xs:attribute name="arity" use="required">
<xs:simpleType>
      <xs:restriction base="xs:string">
      <xs:enumeration value="single">
      </xs:enumeration>
      <xs:enumeration value="compound">
      </xs:enumeration>
      <xs:enumeration value="collective">
      </xs:enumeration>
      </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="sourceName" type="xs:string" use="optional" />
<xs:attribute name="representationMimeType" type="xs:string" use="optional" />
</xs:complexType>
</xs:element>
</xs:schema>
```

FIGURE 3.4 – Schéma de XML de la règle d'extraction [84].

Tout d'abord, les auteurs définissent l'espace des noms utilisé par des règles d'extraction. Ensuite, les attributs suivants sont définis :

- **Langage :** définie le langage de la règle d'extraction. Actuellement, deux options sont possibles : XPath et Regex. Cet attribut est requis.
- **parentExtractionRuleId :** l'identifiant de la règle du parent qui doit être exécuté en premier, cet attribut est requis pour la bonne formation de la règle.
- **id :** numéro d'identification unique de la règle d'extraction. Cet attribut est requis bonne formation de la règle.
- **relatedElement :** désigne le nom du domaine de la PE, dans lequel le résultat de cette règle est stocké, cet attribut est requis pour la bonne formation de la règle.
- **arity :** indique le type de contenu du bloc qui est extrait par cette règle. Cet attribut est requis. Nous avons trois types d'arité :

– **Single :** c'est le type le plus simple de données utilisé dans PE.

– **Collective :** utilisé lorsque la règle extrait plusieurs éléments du même type. Un exemple peut servir l'extraction de bloc avec les employeurs, dans lequel l'élément "expérience professionnelle" est décrit par les mêmes attributs.

– **Compound :** type complexe dans laquelle un processus d'extraction consiste en plusieurs éléments de différents types ou une collection des éléments.

- **SourceName :** le nom de la source des données générées par un ensemble de règles d'extraction. Cet attribut est facultatif.
- **representationMimeType :**type de document décrit, sur lesquels les règles sont exécutées.Cet attribut est facultatif.

L'ensemble des règles d'extraction sont stockées sous le format XML. Les jeux de règles sont conservés sous forme de fichiers XML et stockées dans un dossier dédié, accessible uniquement par composante EXT.

3.4.2.3 Avantages

✓ Les règles dans le cas présenté (contrairement à d'autres scénarios d'extraction d'informations) ne sont pas de nature statistique.

✓ La méthode est basée sur l'hypothèse que les sources des documents ont une structure fixe.

✓ Les règles d'extraction développées font face très bien à l'information structurée.

✓ La traduction d'information semi-structurée à une information structurée, étant sous forme de document de HTML.

3.4.2.4 Inconvénients

– La méthode traite et extrait les informations à partir des sources particulièrement dans la langue polonaise.

– Cette technique serait particulièrement intéressante en cas de traitement de contenu des blocs qui peuvent être librement remplis par des utilisateurs. Mais dans ce cas, il n'est pas possible d'employer des règles pré-définies d'extraction, car quelques méthodes heuristiques sont nécessaires.

3.4.3 Le poids des entités nommées : filtrage des termes pour un domaine donné

3.4.3.1 Présentation

L'extraction automatique des termes est utilisée pour des tâches variées comme l'analyse terminologique, la détection des mots clés pour la recherche d'information et la construction d'ontologies. Les outils de traitement automatique de la langue (TAL) ont la charge d'extraire les termes d'un domaine à partir de corpus spécialisé, mais ces outils n'extraient pas que des termes pertinents. L'objectif de l'article [68] est d'améliorer la sélection des termes pour un domaine donné. Les auteurs proposent des méthodes de filtrage et de pondération de termes qui tiennent compte de la distribution des termes au voisinage des entités nommées et ils montrent qu'elles aident à détecter les termes représentatifs d'un domaine.

Les entités nommées sont des unités textuelles qui ont suscité beaucoup d'intérêt en TAL et elles ont la particularité de renvoyer à des entités du monde. Les outils de reconnaissance d'entités nommées permettent de repérer les entités nommées d'un texte et de leur attribuer un type sémantique qui dépend du domaine considéré. L'article [68] montre comment le voisinage des entités nommées permet de filtrer et pondérer une liste de termes en fonction d'un domaine particulier.

3.4.3.2 Principe

Les outils d'extraction de termes, qui sont généralement indépendants de tout domaine, produisent des résultats bruités et nécessitent souvent un travail manuel pour sélectionner les termes les plus pertinents pour un domaine particulier.

L'article de [68] propose de s'appuyer sur les propriétés de domaine des entités nommées pour améliorer cette sélection.

Étant donné des types sémantiques pertinents pour un domaine donné, ils définissent une relation de voisinage entre des occurrences de terme et d'entités nommées. En prenant la phrase comme contexte, les auteurs proposent qu'une occurrence de terme t figure au voisinage d'une occurrence d'entité nommé e (vois(t , e)) si et seulement si elles figurent dans la même phrase.

La méthode proposée est comme suit : un terme t est pertinent si et seulement si l'une de ses occurrences figure au voisinage d'une occurrence d'entité nommée.

$$Pert(TC) = \begin{cases} 1 & si \exists \text{ t,EN,e/occ(t,TC)} \wedge \text{occ(e,EN)} \wedge \text{vois(t,e)} \\ 0 & sinon \end{cases} \quad (3.9)$$

Tel que :

– **TC** : un terme candidat.

– **EN** : une entité nommée.

– **occ (x , X)** et **vois(x , y)** : indiquent respectivement que x est une occurrence de X et que et y co-occurrent dans la même phrase.

Dans la sélection des termes pertinents, les auteurs proposent de trier les termes sur la base de leurs relations de voisinage en considérant que certains voisinages sont plus marqués que d'autres.

Le poids d'un terme est définit comme suit :

$$Poids(TC) = \frac{Freq_{vois}(TC)}{FreqTotale(TC)} \quad (3.10)$$

où $Freq_{vois}$(TC) est le nombre total de relations de voisinage dans lesquelles entrent les occurrences de TC et FreqTotale(TC) sa fréquence totale (nombre d'occurrences). Le poids d'un terme est donc le nombre moyen de relations de voisinage dans lesquelles entrent ses occurrences.

3.4.3.3 Avantages

✓ Amélioration de la sélection des termes.

✓ Utilisation des entités nommées qui facilite la validation manuelle de longues listes de termes.

✓ La méthode proposée utilise le critère de voisinage des entités nommées qui permet de filtrer efficacement la liste des termes fournie par un extracteur de termes générique et d'éliminer une bonne partie de termes faiblement pertinents.

3.4.3.4 Inconvénients

– Il se peut qu'un terme soit pertinent alors qu'il n'est pas voisinage d'une entité nommée.

3.4.4 Indexation sémantique des documents multilingues

3.4.4.1 Présentation

Dans [27], l'auteur décrit une méthode d'indexation sémantique adaptée aux documents multilingues. Il propose une démarche d'extraction des concepts et des relations entre les concepts. L'idée centrale du travail réalisé est que l'utilisation des ressources sémantiques externes telle que les ontologies et les thésaurus peuvent améliorer l'efficacité des processus d'indexation.

La méthode d'indexation proposée consiste en, première étape, à identifier des concepts en repérant les termes qui les dénotent dans les documents et en projetant ces termes sur l'ontologie. La seconde étape, concerne la détection des relations qui résident entre ces concepts.

Ainsi pour l'extraction des concepts, la méthode comprend deux étapes : (i) une étape d'extraction des termes, qui permet d'associer à chaque document un ensemble de termes pertinents,
(ii) une étape de transformation de la représentation termes à la représentation concept.

3.4.4.2 Principe

a. Extraction des termes

L'approche présentée permet d'extraire dans une première étape les termes simples et dans une deuxième étape les termes composés. Elle se base sur la définition d'un corpus spécialisé et sur des mesures statistiques telles que l'information mutuelle et la loi de Zipf (ZIPF, 1949) qui vérifie manuellement que dans un corpus textuel, la fréquence (f) d'un mot est inversement proportionnelle à son rang (r). Le rang d'un mot est sa position dans la liste des fréquences triées dans l'ordre décroissant des mots du corpus. Dans cette liste le mot le plus fréquent est de rang 1. La loi portant son nom est formellement exprimée de la manière suivante :

$$\forall\ m_i \in M_c, f \quad (m_i, c) * r \quad (m_i, c) \cong constante \tag{3.11}$$

- m_i : un mot.
- M_c : l'ensemble des mots du corpus C.

La figure 3.5 présente la méthode proposée afin d'extraire automatiquement les termes simples à partir des corpus multilingues :

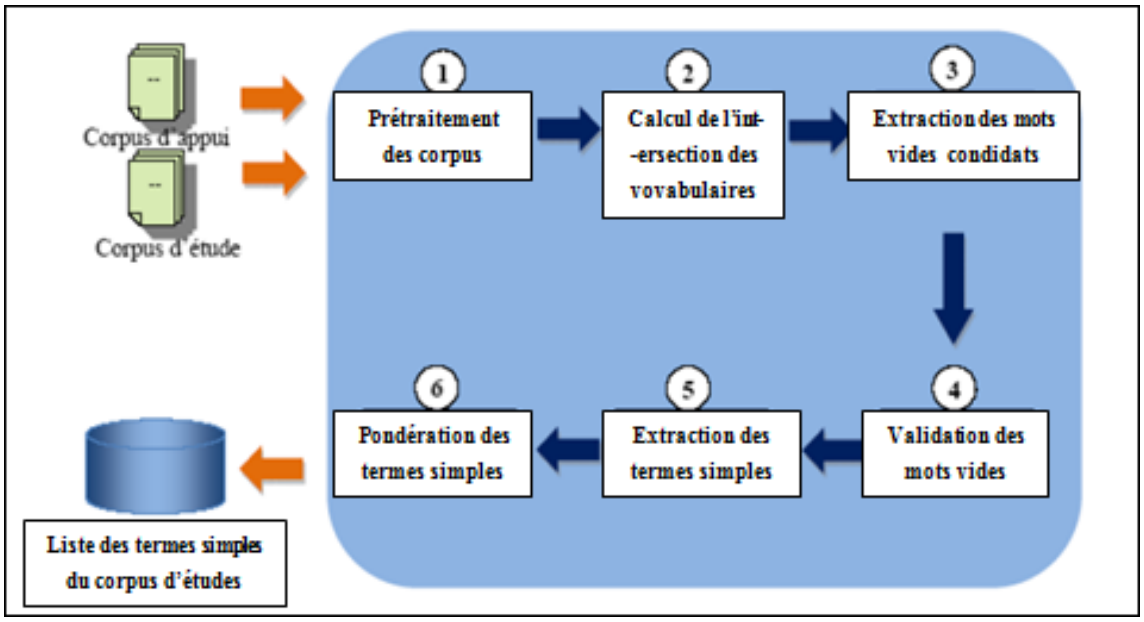


FIGURE 3.5 – Vue d'ensemble de l'approche proposée pour l'extraction automatique des termes simples à partir des corpus multilingues [27].

La pondération des termes consiste à affecter à chaque terme un poids qui représente son degré de pertinence dans le document où il apparaît. Ce poids permet de distinguer les documents entre eux. En effet, un terme ne représente d'une manière adéquate le document que si son poids dans ce document est assez significatif. Un terme qui apparaît dans tous les documents n'est pas discriminant c'est-à-dire qu'il ne permet pas de distinguer un document des autres documents. Un poids faible sera affecté à ce terme. Cette mesure n'a pas l'objectif d'éliminer des termes simples qui ont été déjà validés dans l'étape précédente. Mais, elle permet de trier ces termes par ordre d'importance.

Dans cette approche l'auteur a adapté la formule de pondération suivante :

$$TF * IDF_{i,j} = 0.4 + 0.6 * \left(\frac{tf_{i,j}}{tf_{i,j} + 0.5 + 1.5 * \frac{dl_j}{\Delta l}} \right) * \left(\frac{\log \frac{N+0.5}{n_i}}{\log(N+1)} \right) \tag{3.12}$$

- TF : Fréquence du terme t dans le document.
- IDF : Fréquence inverse du terme t dans le document.
- N : est le nombre total des documents dans le corpus.
- n_i :est le nombre de documents contenant terme i.
- $tf_{i,j}$: est la pondération locale du terme i dans le document j.
- dl_j :est la longueur du document j en nombre de mots.
- Δl : est la moyenne des longueurs des documents du corpus en nombre de mots.

L'auteur de cette approche propose une technique statistique, qui permet d'identifier les termes composés à partir d'un corpus de documents textuels multilingues. Elle se base sur une variante de l'information mutuelle. Afin de résoudre le problème de la construction des termes composés de longueur n+1 à partir des termes composés de longueur n, l'auteur propose de ne pas prendre en compte la fréquence d'un mot vide durant la construction. Ainsi, il définit une nouvelle mesure : l'information mutuelle adaptée. Pour un couple de mots (m_i, m_j), l'information mutuelle adaptée est calculée de la manière suivante :

$$IMA(\ \mathrm{m}_i, m_j) \quad = \quad \begin{cases} -log_2\left(\frac{f(m_i,m_j)}{f(m_i)*(m_j)}\right) & si \quad \mathrm{m}_j \text{ est un terme} \\ \\ -log_2\left(\frac{f(m_i,m_j)}{f(m_i)*(m_i)}\right) & si \quad \mathrm{m}_j \text{ est un mot vide} \end{cases} \tag{3.13}$$

La figure 3.6 présente les étapes d'extraction automatiquement des termes composés à partir des corpus multilingues :

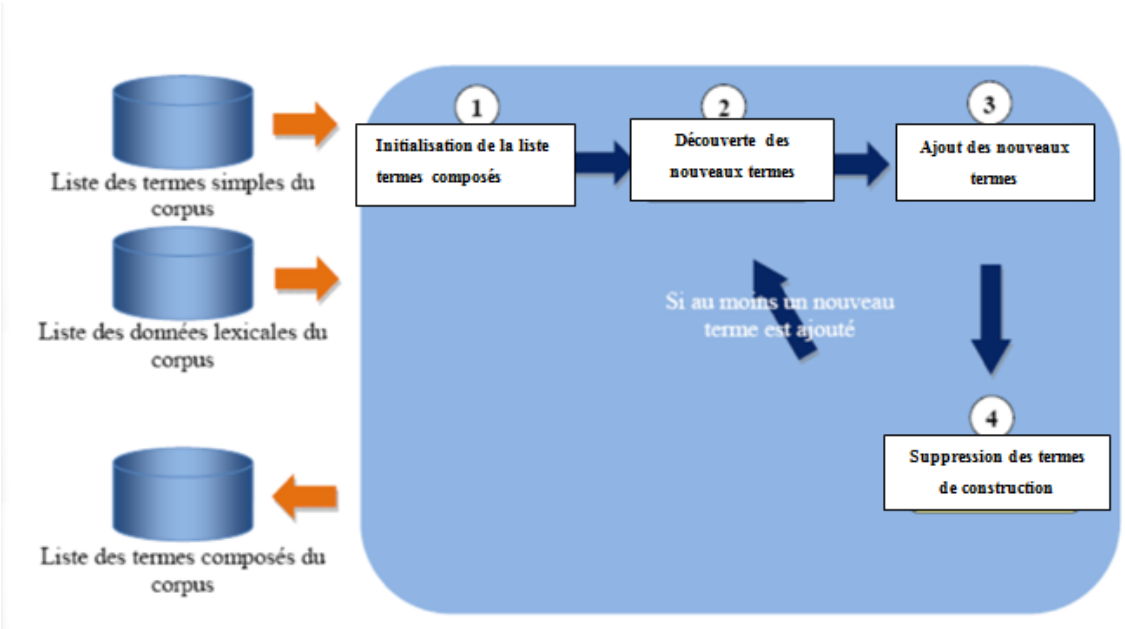


FIGURE 3.6 – Processus d'extraction des termes composés [27].

Ainsi, la pondération qui favorise les termes composés, se note : CTF * $IDF_{i,j}$.

La pondération d'un terme composé est proportionnelle à sa longueur. L'augmentation de la valeur de cette pondération se fait par : (1-1/longueur(i)). La mesure CTF * $IDF_{i,j}$ est donc exprimée en fonctions de ces facteurs de la manière suivante :

$$CTF * IDF_{i,j} = \left(1 - \frac{1}{Longueur(i)}\right) + TF * IDF_{i,j} + \left(\frac{1}{longueur(i)}\right) * \sum_{k \in i} TF * IDF_{k,j} \quad (3.14)$$

– i : un terme composé.

– j : un document.

– k : un terme simple.

– $TF * IDF_{k,j}$: la pondération du terme i dans le document j.

– TF * $IDFi, j$: la pondération du terme i dans le document j.

– *Longueur(i)* : le nombre de terme simples qui participe dans la construction du terme composé i.

b. Extraction des concepts

Le but de cette étape est d'extraire les concepts à partir des documents multilingues. La démarche suivie pour l'extraction est présentée dans la figure 3.7 :

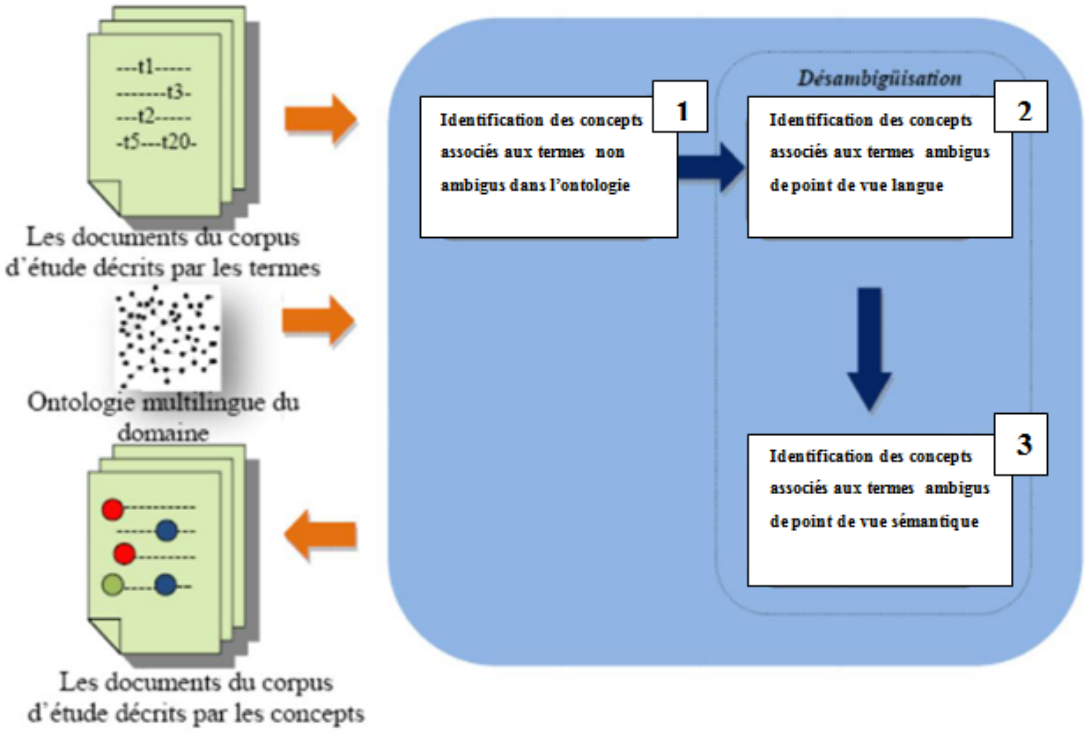


FIGURE 3.7 – Vue d'ensemble de l'approche proposée pour l'extraction des concepts [27].

3.4.4.3 Avantages

✓ L'extraction à base de termes et concepts, combinés, renforce la précision du système.

3.4.4.4 Inconvénients

– Le seuil concernant la longueur des mots vides limite la précision du système.

3.4.5 Définition d'une signature unique pour un profil

3.4.5.1 Présentaion

Le profilage ou " profiling " a trouvé son application dans le domaine de la formation des profils des personnes, en mesure d'établir un ensemble de caractéristiques spécifiques à un ou à un groupe d'individus. Le profilage informatique est reconnu comme le résultat d'une méthode informatisée faisant appel aux procédés de data Mining sur des entrepôts de données et permettant de situer, avec une marge réduite d'erreur, un individu dans une catégorie particulière et le but étant de formuler des décisions à son égard.

La thématique étudiée dans [57] s'articule autour de la notion du profilage informatique. Le but est de proposer une approche capable de détecter un profil Web d'une

façon unique. Il s'agit en effet de modéliser une empreinte Web pour chaque profil, l'idée générale étant de retrouver cette signature à partir de messages écrits dans les forums, en analysant le vocabulaire employé par chaque internaute.

3.4.5.2 Principe

L'approche consiste à extraire les textes des pages Web, ainsi que les commentaires des internautes et de construire une ontologie du discours qui comprendra le vocabulaire utilisé (tous les mots et termes spécifiques) par un individu.

Les travaux de [28] et [44] ont défini quatre familles de caractéristique : caractéristiques lexicales(la fréquence des mots, de l'alphabet, le nombre de majuscules, nombre moyen de caractères par mot...etc.), les caractéristiques syntaxiques (l'utilisation des mots tel : tant, bien, ou,.. et la ponctuation : !, ?, :,...), les caractéristiques structurelles (la manière dont une phrase ou un paragraphe est structuré), les caractéristiques liés au domaine.

Ces différentes caractéristiques seront utilisées dans le but de la détection d'auteurs (rapprocher les textes écrits par le même auteur et les rassembler au sein d'un cluster).

Soit N textes extraits du Web de M auteurs, l'outil développé aura à dégager les M clusters regroupant avec une certaine probabilité les textes rédigés par la même personne, tout en utilisant les deux algorithmes de clustering, EM (Expectation Maximisation) et K-means.

Pour calculer la performance du prototype développé et évaluer les résultats, les auteurs ont fait appel aux deux paramètres *Rappel (R)* et *Précision (P)* et la *F-mesure (F)* issus de la statistique :

$$P(i,j) = \frac{T_{ij}}{Ta_i}. \tag{3.15}$$

$$R(i,j) = \frac{T_{ij}}{Tc_i}. \tag{3.16}$$

Tel que :

- T_{ij} : Nombre de textes de l'auteur i présents dans le cluster j.
- Ta_i : Nombre total de textes de l'auteur i.
- Tc_i : Nombre total de textes du cluster i.

$$F(i,j) = (2 * P(i,j) * R(i,j))/(P(i,j) + R(i,j)). \tag{3.17}$$

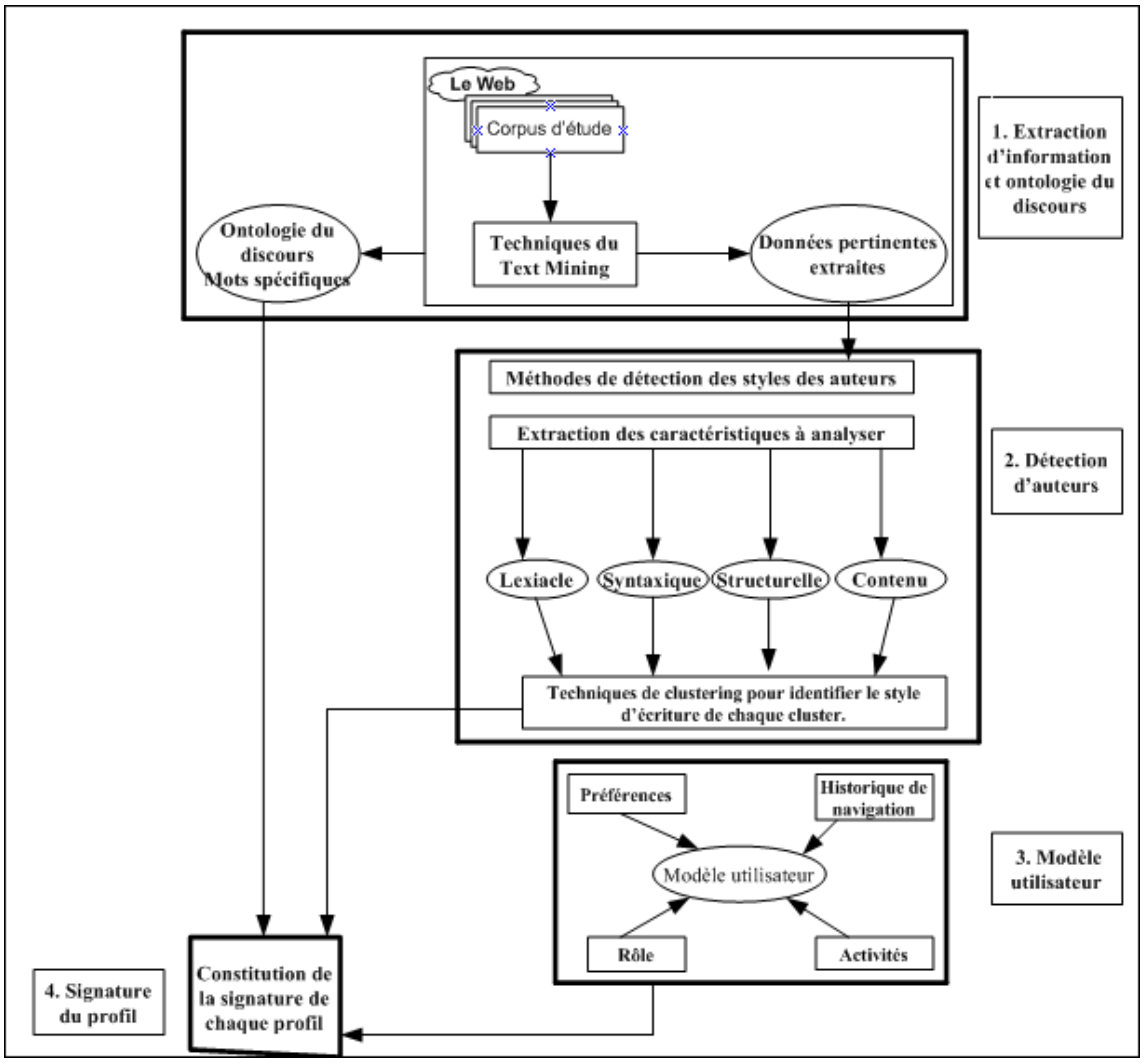


FIGURE 3.8 – Schématisation des différentes étapes de l'approche [57].

Les commentaires et messages Web sont des textes non structurés, écrit dans un langage informel et simplifié, avec des mots courts et incorrects (l'utilisation des abréviations), les auteurs utilisent les Ratio, l'approche proposée par [1], ils ont également pû détecter certains mots fréquents utilisés par les internautes qu'ils ont rajoutés aux caractéristiques.

3.4.5.3 Avantages

✓ La détection et l'intégration d'une partie du vocabulaire des internautes dans l'analyse des textes.

✓ L'utilisation de la puissance sémantique des ontologies.

✓ Réussir à avoir une signature unique du profil utilisateur et pouvoir ainsi l'identifier parmi d'autre.

3.4.5.4 Inconvénients

– Certains messages contiennent des images significatives.

– Les commentaires utilisateur peuvent être des propos rapportés, citations, poèmes article d'un journal... etc.

– Le profil utilisateur, ces centres d'intérêt et ces opinions évoluent à travers le temps.

3.4.6 Extraction de phrases pertinentes d'articles scientifiques

3.4.6.1 Présentaion

Les données de la biologie se caractérisent par leur diversité et leur hétérogénéité. En effet le langage naturel (utilisée dans les publications scientifiques) est un support de communication dont il est difficile d'extraire de l'information de façon automatique. Il est donc nécessaire de modifier les données de départ avant de pouvoir les exploiter.

3.4.6.2 Principe

Dans [5], les auteurs travaillent sur les documents textuels de PubMed, un index des articles publiés en biologie, ils s'intéressent particulièrement au contenu des résumés (Abstract), et les bases de données lexicales (LocusLink, OMIM, Gene Ontology) qui contiennent le vocabulaire nécessaire pour décrire les informations concernant les gènes (nom, rôle, les différentes pathologies, les voies métaboliques, les éventuelles interactions avec d'autres protéines, etc).

Ce travail est divisé en deux axes principaux, l'analyse et l'extraction d'information (avec des aspects algorithmiques et statistiques) d'une part, le stockage et la restitution de l'information (base de données et Interface Homme-Machine (IHM)) d'autre part. La première partie consiste à transformer des données bibliographiques non structurées ou semi-structurées (textes en anglais) en données structurées utilisables pour une étude statistique (liste de mots). Cela permet d'extraire les informations pertinentes. La seconde partie du projet s'articule autour de la mise en place d'une base de données permettant de stocker les informations générées et de la création d'une interface graphique permettant d'interroger la base de données ainsi constituée.

Leur but est d'extraire des informations concernant les interactions entre gènes, pour ceci, ils se basent sur le principe de ne pas rencontrer les mêmes mots dans les deux catégories de phrases. Pour ce faire, ils distinguent plusieurs taches : *Texte* (une liste de Phrases). *Phrase* (une liste de mots ou de lemmes). *Classificateur* (classe abstraite qui décrit l'interface des différents classificateurs : une phase d'apprentissage avec un corpus d'apprentissage, une méthode pour calculer la probabilité d'une phrase). *IVI et Bayes* (deux implementations de cette interface). LocusLink (un dictionnaire contenant les noms officiels et les synonymes).

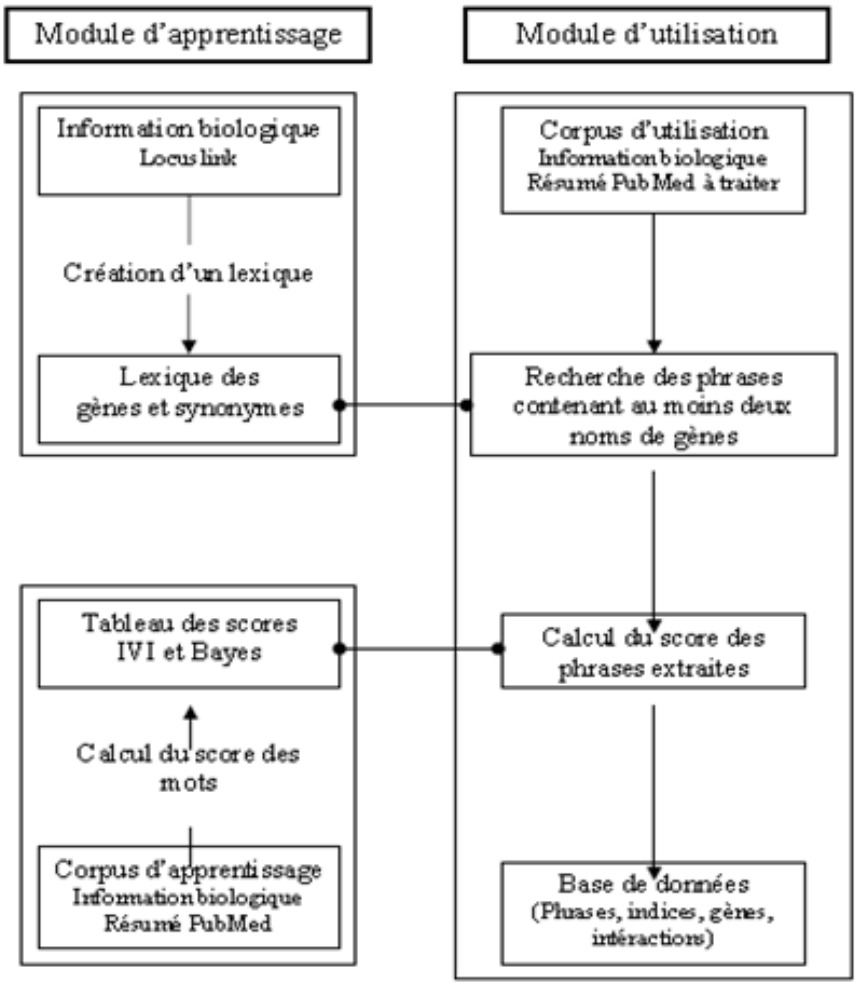


FIGURE 3.9 – Fonctionnement de l'algorithme [5].

Comme le montre la figure 3.9, le texte va être décortiqué en phrases puis en mots, et seule les phrases comportant au moins deux noms de gènes sont retenus. En utilisant les données du module d'apprentissage, une probabilité d'interaction pour chaque phrase est calculée en additionnant les indices de chaque mot de la même phrase, ainsi toute celle qui dépassera un seuil donné est sélectionnée.

3.4.6.3 Avantages

✓ Extraction rapide d'informations.

✓ La possibilité de raffiner les résultats avec les méthodes statistiques et les ontologies.

3.4.6.4 Inconvénients

– Cette méthode repose sur l'étape de l'extraction de phrase, qui est définie comme toute chaîne qui se termine avec un point, alors qu'il se peut que les gènes soient représentés par des pronoms personnels et donc ces phrases ne seront pas extraites.

– La méthode proposée peut être diluée par un faux positif (phrases extraites alors qu'elles ne décrivent pas d'interaction).

– La présence d'ambiguïté dans les noms des gènes (la présence de '/' ou d'espace).

3.5 Les domaines hybrides : Recherche et Extraction d'Information

3.5.1 Le Web

3.5.1.1 Présentation

L'extraction d'information consiste donc à extraire des connaissances à partir de différents documents en utilisant entre autres des techniques linguistiques. Ceci ajoute donc une plus-value au processus de recherche traditionnelle par mots-clés. La mise au point d'un tel système est une tâche longue et fastidieuse qui demande souvent une expertise du domaine sur lequel on travaille ainsi que des connaissances en linguistique.

3.5.1.2 Principe

L'auteur dans [61], travaille sur l'extraction des informations et la sémantique des documents afin d'en créer une ontologie en utilisant une hybridation des deux logiciels Sesei [80] qui se base sur le formalisme des graphes conceptuels afin d'avoir une représentation sémantique des documents retournés par le moteur de recherche Google, pour ensuite relier les documents fournis par Google et la requête avec une Ontologie construite à partir des mots de celle-ci, et le logiciel Text-To-Onto [8] dont les auteurs ont tenté d'automatiser le processus du maintien d'une ontologie à l'aide des techniques d'apprentissage

automatique et des méthodes statistiques, en essayant d'extraire les liens entre les différents concepts contenus dans des textes. Ceci leurs permet d'avoir une ontologie crée en fonction du domaine qu'ils essayent de traiter, pour ainsi donner naissance à *SeseiOnto*.

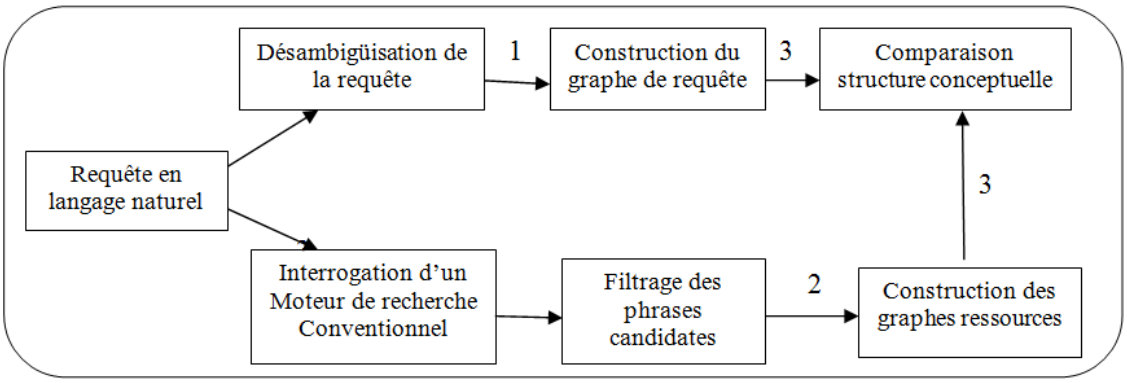


FIGURE 3.10 – Aperçue du fonctionnement de Seisi [61].

Comme c'est illustré par la *figure 3.10*, les étapes de traitement d'une requête sont : représentation par graphe grammatical, représentation par graphe conceptuel, désambiguïsation, le filtre ontologique et le filtre sémantique. Où la requête de l'utilisateur est décomposée à l'aide du logiciel *Connexior* et représentée sous forme d'un arbre, vient ensuite l'étape de la désambiguïsation où l'utilisateur aura à désambiguïser les sens des mots de sa requête fournis par WordNet (le dictionnaire utilisée) manuellement. Parmi les documents retournés par le moteur de recherche, seulement les phrases les plus proches sémantiquement de la requête seront sélectionnées, en fonction d'un score qui leurs sera attribué aux mots (synonymes, hyponyme...) et d'un poids (verbe, nom, adjectif, adverbe...).

La pertinence d'une phrase = **Score de la phrase/ Score de la requête.**

Les phrases sélectionnées seront transformées en graphes conceptuels, pour ensuite essayer de trouver une généralisation afin de contenir la requête de l'utilisateur.

Le changement survient après l'étape de création du graphe grammatical. SeseiOnto charge alors en mémoire l'ontologie spécifiée par l'usager. L'utilisateur aura encore à désambiguïser chacun des mots de sa requête. Par la suite, une ontologie qui est une combinaison de l'ontologie Text-To-Onto et de l'ontologie WordNet est crée à partir des mots de la requête. Si le mot de la requête est dans l'ontologie Text-To-Onto, cette dernière est utilisée, si le mot n'a pas été trouvé, alors le concept de l'ontologie WordNet est utilisé, puisque son sens a été spécifié par l'usager. Ceci garantit donc que chacun des mots de la requête sera retrouvé soit dans l'ontologie Text-To-Onto, soit dans l'ontologie WordNet bâtie à partir des mots de la requête. De plus, l'ontologie Text-To-Onto étant bâtie en fonction des documents qui composent le corpus, celle-ci sera beaucoup plus vaste et

permettra de diminuer le nombre de concepts inconnus dans les documents qui seront traités par Sesei.

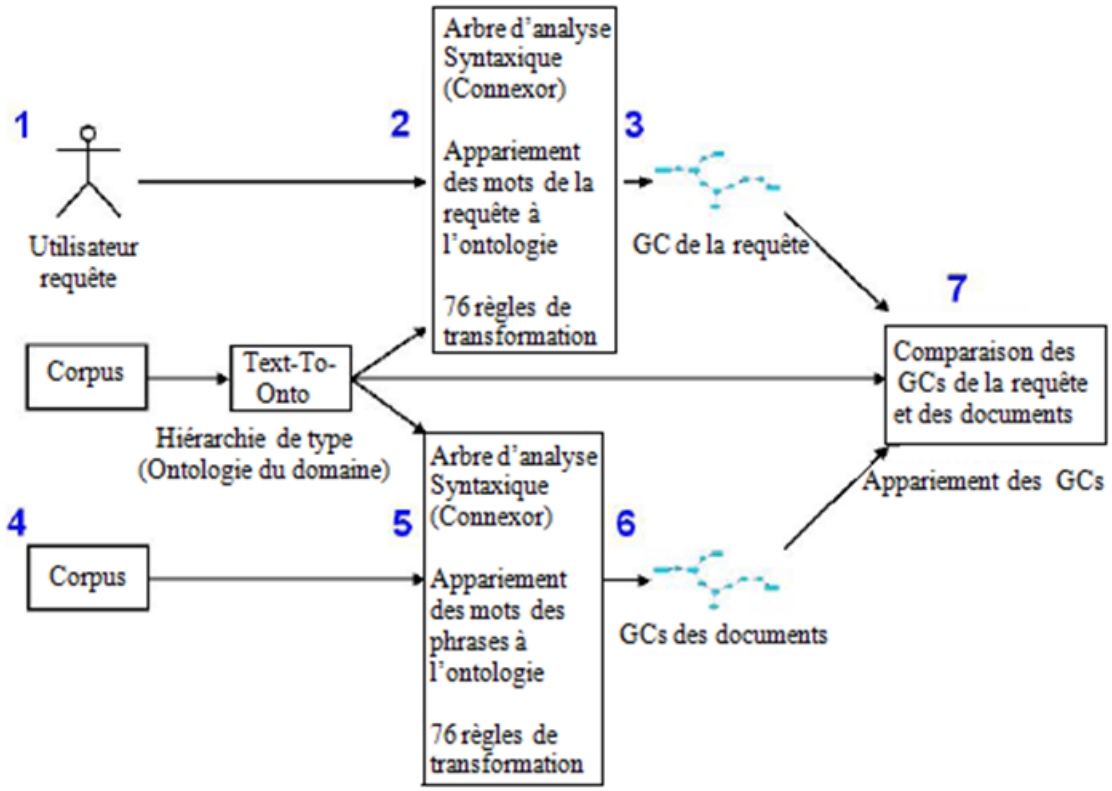


FIGURE 3.11 – Diagramme du fonctionnement globale de Seisi-Onto [61].

3.5.1.3 Avantages

✓ Deux ontologies jumelées diminuent le nombre de concepts inconnus.

✓ Avoir une ontologie globale, qui contient les concepts des documents ainsi que ceux appartenant à la requête.

3.5.1.4 Inconvénients

– Les concepts de WordNet peuvent avoir un sens différent inapproprié au domaine.

– Deux ontologies peuvent être une source d'ambigüité.

– La construction des ontologies se fait par la sélection des concepts dépassant un *seuil* de fréquence, ce qui fait que la précision du système dépend de la valeur *seuil*.

3.6 Tableau comparatif

Nous avons essayé dans le tableau 3.1 de présenter les principales différences entre les méthodes citées précédemment, en s'inspirant des critères de comparaison cités dans [15] : le type du document traité (type de page), les techniques utilisées par chaque méthode, le degré d'automatisation (qui mesure le degré d'intervention d'utilisateur), le degré de précision, la source de l'information étudiée ainsi que le domaine d'application utilisant la méthode proposée.

3.6.1 Types des données :

Les méthodes étudiées travaillent sur des informations de nature textuelle, généralement sous format électronique tel que : LATEX, HTML, XML, PDF...etc, nous en distinguons : les documents structurés, les documents non structurés et les documents semi-structurés selon les domaines de recherche.

a. Texte Structuré : Nous trouvons les informations structurées dans les bases de données et les langages informatiques. Nous reconnaissons les informations structurées au fait qu'elles sont disposées de façon à être traitées automatiquement et efficacement par un logiciel, mais non nécessairement par un humain : les documents XML.

b. Texte Semi-structuré : Tout document texte pouvant être traité semi-automatiquement : les pages HTML, les dossiers médicaux...etc.

c. Texte Non-structuré : Les textes libres sont non structurés car ils nécessitent un réel traitement du langage naturel.

3.6.2 Les techniques utilisées :

C'est l'ensemble des techniques et méthodes d'extraction utilisées par chaque méthode tel : les dictionnaires sémantiques, les modèle graphiques (conceptuels), les modèles statiques, les techniques de data-maining...etc.

Définition 3.6.1. Un dictionnaire sémantique, est un dictionnaire contenant les concepts de la connaissance humaine tel Wikipédia... etc.

Définition 3.6.2.

3.6.3 Le degré d'automatisation :

Représente le degré d'intervention de l'utilisateur dans la solution proposée et nous en avons distingué des systèmes automatiques, semi-automatiques (non automatique n'est pas intéressent).

3.6.4 La précision du système :

Représente le degré d'exactitude des résultats retournés par le système (La précision sémantique avec l'utilisation de l'ontolgie).

3.6.5 Les sources d'information :

Représente les documents utilisés ou bien les documents étudiés tel : les forums web, les pages web HTML, les articles scientifiques...etc.

3.6.6 Le domaine d'application :

Représente le domaine où la méthode est appliquée, dans notre cas les domaines représentent l'extraction, la recherche d'informations, conversations électroniques... etc.

Critères	Type des données	Techniques d'extraction	Degrés d'auto-matisation	Précision	Source d'information	Domaine
[24]	Non-Structuré	Statistiques	–	–	Articles scientifiques	Résumé
[47]	Non-Structuré	Vecteurs, Graphes et Statistiques	Automatique	–	Pages Web	Résumé
[56]	Semi-Structuré	Graphes et statistiques	Semi-automatique	Non-précis	Historique et centres d'intrêt	Recherche
[60]	Semi-Structuré	Sémantique (Ontologies)	–	Précis	Session de recherche et Ontologie	Recherche
[62]	Non-Structuré	Ontologies (Sémantique)	Automatique	–	Conversations électroniques	Extraction
[5]	Non-Structuré	Probabilités et Ontologies	–	–	Articles scientifiques	Extraction
[84]	Semi-Structuré	Arbre et XML	Automatique	–	Pages HTML	Extraction
[68]	Non-Structuré	Statistique	Automatique	–	Textes libres	Extraction
[27]	Semi-Structuré	Statistiques et Sémantiques	Automatique	Précis	Diagnostiques médicaux	Extraction
[57]	Non-Structuré	Stylométrie , Ontologie et Clustering	Automatique	–	Forum web	Extraction
[50]	Semi-structuré	Analyse lexical, Modèle probabiliste	Automatique	–	Announces d'emploi	Extraction
[61]	–	Graphes conceptuels et Ontologie	Semi-automatique	–	Document électronique	Recherche et Extraction

TABLE 3.1 – Synthèse des travaux existants.

3.6.7 Discussion

L'étude et l'analyse de quelques méthodes et techniques existantes, nous a permis d'en distinguer quelques avantages, inconvénients, et différences que nous avons présenté dans le tableau 3.1. Afin d'élaborer un système plus performant, nous avons cité les points suivants :

- L'absence de la sémantique dans la plupart des travaux.
- La fréquence du terme, présente pas toujours une information pertinente.
- L'utilisation des mots clés, nous fait perdre l'information concernant le contexte, car un concept ne peut pat être décrit par un seul mot.

3.7 Conclusion

Nous avons présenté dans ce chapitre un panel des travaux existants sur l'axe de la recherche et l'extraction d'informations. Nous avons illustré les étapes suivies lors des pré-traitements des documents textuelles et lors de détermination des termes et concepts pertinents, pour ensuite, en conclure les avantages et inconvénients de chacune d'elles.

Nous avons constaté que les problématiques et les solutions présentées par les chercheurs, diffèrent l'une de l'autre en fonction du domaine d'application, techniques utilisées pour l'extraction,... que nous avons illustré dans le tableau comparatif.

Dans le chapitre suivant, nous présentons notre système de construction du profil utilisateur.

Chapitre 4

Proposition : Extraction d'information pour la construction d'un profil utilisateur

4.1 Introduction

L'objectif des recherches actuelles dans le Web tend vers l'optimisation du rendement et de la précision des résultats retournés par le Web, ce qui représente le vecteur d'orientation de notre thématique de recherche.

Notre approche s'intéresse à la construction et la détermination d'un profil utilisateur à partir d'informations textuelles, collectées à partir des e-mails électroniques. Notre sujet de recherche est à l'intersection de plusieurs disciplines ; Intelligence Artificielle ; Traitement de la langue ; Text-mining ; Web sémantique, que nous exploiterons dans le but d'extraire les informations pertinentes concernant le profil utilisateur.

L'extraction d'informations pour la construction d'un profil utilisateur, est une notion primordiale pour différents domaines ; le commerce électronique, le domaine juridique et criminel, la recherche d'information (personnalisée), la sécurité. . .etc. Nous nous intéressons au domaine du recrutement électronique, vu l'évolution impressionnante de celui-ci via le Web, ainsi que la faiblesse des outils dédiés à la gestion des demandes d'emploies, afin de satisfaire les besoins des utilisateurs et de pouvoir assurer un rapprochement automatique entre les offres et les demandes d'emploies.

4.2 Architecture globale du système proposée

Dans le but de construire le profil utilisateur, nous proposons un système composé d'un ensemble de modules qui sont :

- ✍ Pré-traitement de l'e-mail.
- ✍ Résumé hybride.
- ✍ Détection des entités nommées.
- ✍ Matching.
- ✍ Mise à jour.

La figure 4.1 représente l'architecture globale proposé :

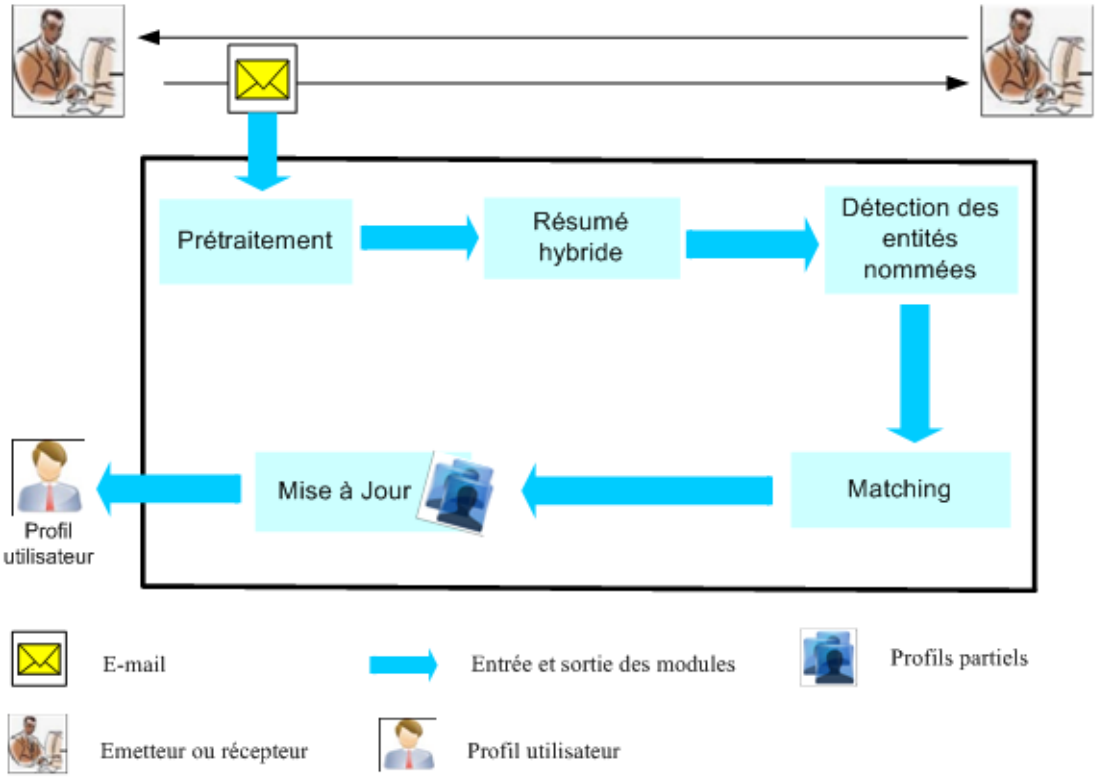


FIGURE 4.1 – L'architecture globale du système proposé.

Notre approche utilise l'information véhiculée par les e-mails électroniques au cours d'une conversation, nous supposons que :

– La conversation concerne un seul domaine fixé à l'avance, qui est le recrutement dans le domaine informatique.

– La langue utilisée dans l'e-mail est l'anglais.

Chaque échange, permet une exécution du système et ainsi la construction d'un profil partiel pour chaque e-mail, jusqu'à la satisfaction de la condition d'arrêt, qui est le changement du domaine détecté à partir du sujet de l'e-mail. Le processus du mise à jour est lancé à chaque fin de construction d'un profil partiel. Tout au début : profil global = profil initial.

4.3 Présentation du système

a. L'E-mail

Un e-mail électronique est un texte structuré, qui contient quatre champs informatifs :

- **Emetteur :** contient une information qui représente l'adresse e-mail de l'utilisateur.
- **Destinataire :** c'est le champ concernant l'information désignant l'identité du récepteur.
- **Sujet :** c'est le champ qui définit le sujet de l'e-mail.
- **Corps de l'e-mail :** contient l'information globale véhiculée par l'e-mail, en d'autres termes un développement au sujet de l'e-mail.

b. Les informations du profil utilisateur

Le modèle du profil utilisateur consiste en la modélisation de l'utilisateur à travers la description de ses caractéristiques informationnelles. Pour la construction de notre profil utilisateur, nous nous inspirons des entités du modèle proposé par Sklab [89] :

- **Données personnelles :** la dimension des données personnelles regroupe des informations personnelles sur l'utilisateur comme les données professionnelles ou démographiques. Elle peut regrouper des données comme le nom, le prénom, l'adresse, son numéro de téléphone ou de fax, son adresse e-mail etc. Ces données sont généralement stables et peu changeantes.
- **Education :** il s'agit de détailler et de préciser toutes les études menées, les diplômes préparés ou obtenus, les formations suivies. Chaque formation est décrite séparément par : l'année de la formation (début et fin), l'intitulé de la formation, l'établissement fréquenté et le diplôme obtenu.
- **Expériences professionnelles :** chaque expérience professionnelle est décrite séparément avec les informations suivantes : l'intitulé du stage ou de la fonction occupée, la durée de la fonction (début et fin), le nom de l'entreprise et sa localisation géographique.

- **Préférences de l'utilisateur :** concerne les exigences ou préférences d'un utilisateur, tel le salaire, lieu de travaille... etc.

Toutes ces représentations sont annotés par des ontologies qui donnent une description sémantique sur les attributs.

La figure 4.2 représente le modèle du profil utilisateur que nous utiliserons pour le stockage des informations du profil.

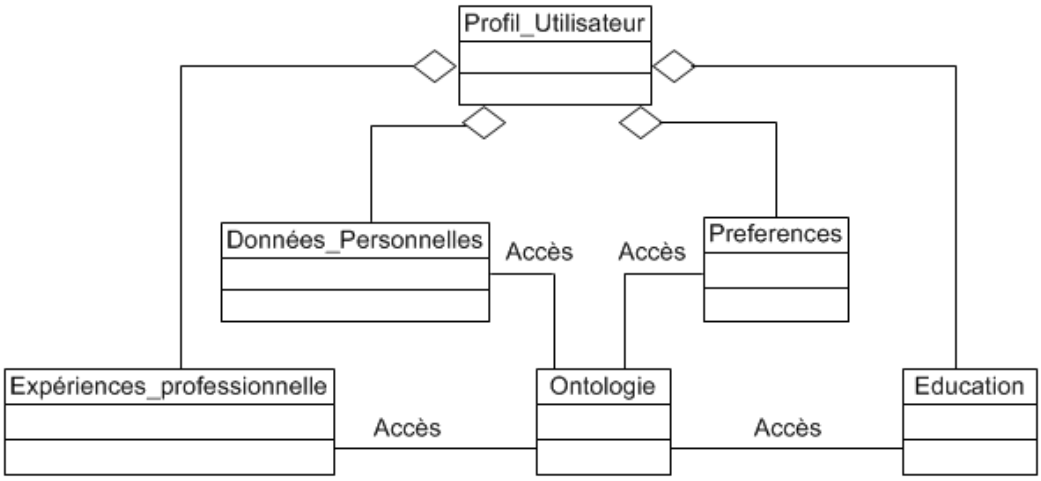


FIGURE 4.2 – Le modèle du profil utilisateur inspiré de [89].

4.3.1 Module de pré-traitement de l'e-mail

Le module du pré-traitement de l'e-mail assure une succession de traitements élémentaires, que nous présentons comme suit :

4.3.1.1 Séparation de l'entête du corps de l'e-mail

Consiste à séparer l'entête du corps de l'e-mail, et récupérer les informations concernant l'émetteur, le récepteur ainsi que le sujet de l'e-mail.

4.3.1.2 Représentation du corps sous format XML

Il y a eu un intérêt croissant concernant XML (*eXtensible Markup Language*) depuis qu'il a été choisit comme la représentation standard des données et échanges sur le Web. Vu les différents avantages offerts par cette représentation [45] : *Un modèle plus structuré et plus organisé, facile à implementer, un format d'échange de données par excellence et l'éventualité d'une représentation relationnelle des données.*

Cette manipulation aisée des documents semi-structurés XML, nous a poussé à son utilisation. La figure 4.3, illustre un extrait du fichier XML que nous avons proposé pour structurer les informations contenues dans le corps de l'e-mail, et qui sera utilisé tout au long de notre approche, et dont les différents champs le composant seront remplis au fur et à mesure.

```
<? xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Email Id="" Sender="" Receiver="" Subject="">>
  <Phrase id="">                          <!-- Numéro de la phrase (La position de la phrase dans le texte global)-->
     <!-- Cette partie sera remplis lors du module du prétraitement -->
 <OriginalPhrase>    </OriginalPhrase>          <!-- Ici, le texte de la phrase originale -->
  <WordNumber> </WordNumber>                    <!-- Le nombre de mots dans la phrase -->
  <TitleSimilarity> </TitleSimilarity>          <!-- Valeur de la métrique sur la similarité de la phrase avec le titre-->
 <PhrasePosition> </PhrasePosition>             <!-- Valeur de la métrique sur la position de la phrase -->
 <SumWordFrequencies> </SumWordFrequencies>     <!-- Valeur de la métrique sur la fréquence des mots dans la phrase-->
 <SumWordWeight>  </SumWordWeight>              <!-- Valeur de la somme des poids des mots dans la phrase  -->
 <SumStatistiqueMetric>  </SumStatistiqueMetric>  <!-- Somme des valeurs des métriques statistiques -->
 <SenteceWeight>   </SenteceWeight>             <!-- Poids de pertinence d'une phrase -->
    <ResumeSentene> </ResumeSentence>           <!— Yes / No Appartient au résumé hybride ou pas -->
      <Word  id=" ">
           <NameW>    </NameW>                  <!-- Le mot dans sa forme originale dans la phrase -->
           <Lemma> </Lemma>                     <!-- lemme, Après lemmatisation -->
           <POS>     </POS>                     <!-- L'annotation utilisée par Part Of Speach-->
           <Sign>     </Sign>                  <!-- L'annotation utilisée  dans les lises d'indicateurs -->
           <EmptyWord></EmptyWord>              <!—Yes / No mot vide ou pas -->
         </Word>
     <WordNormalized  id="">
           <Word  id="">  </Word>
           <NameWN>  </NameWN>
     </WordNormalized>
<!-- Cette partie sera remplis lors du module de Détermination  des Entités Nommées -->
        <NamedEntity  id="">
           <Word id="">  </Word>                <!-- Contient les identificateurs des mots constituants l'entité nommée -->
           <Type>  </Type>                      <!-- Contient le nom  de l'entité nommée -->
        </NamedEntity>
  <!-- Cette partie sera remplis lors du module du Matching et Détermination des Unités d'Informations -->
        <UI id="">
          <Idphrase> </Idphrase>                <!—Identificateur de la phrase-->
          <Subject>  </Subject>                 <!—Le sujet de la phrase -->
```

FIGURE 4.3 – Extrait de notre structure XML.

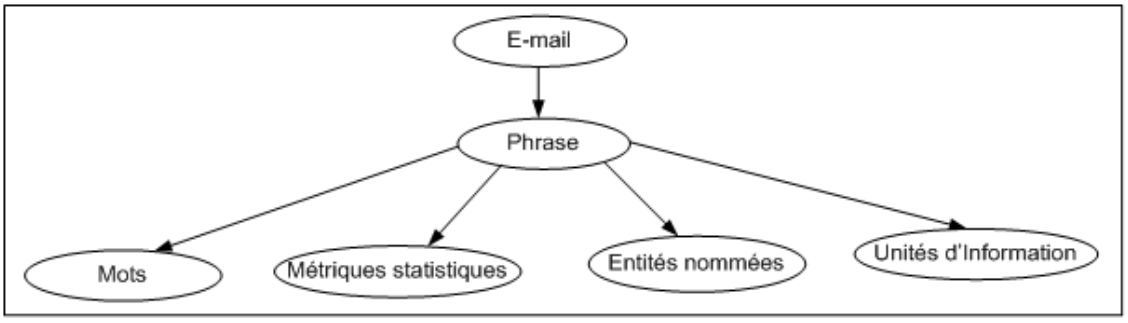


FIGURE 4.4 – Arborescence de la structure XML représentant l'e-mail.

4.3.1.3 Segmentation en phrases

La segmentation de texte est une phase nécessaire pour la majorité des applications spécialisées dans le traitement automatique du langage (TAL), elle consiste à diviser un texte en plusieurs unités textuelles de plus petite taille [7]. La phrase constitue l'unité fondamentale contenant des informations sur le profil utilisateur. En supposant que ces informations sont véhiculées dans des phrases uniques (non dispatchées sur plusieurs phrases à la fois), nous choisissons les phrases comme unités textuelles afin de garder la cohérence et la sémantique du texte.

Dans la segmentation en phrases, nous attribuons à chaque phrase du corps de l'e-mail un identifiant, qui représente son rang [1] d'apparition dans le document, qui sera utilisé dans les étapes ultérieures.

4.3.1.4 Normalisation des dates et des numéros

La normalisation [2] a pour but premier de s'affranchir des variations orthographiques des mots en regroupant les mots porteurs du même sens sous un seul format (Format date, Format numéro). La conséquence directe de la normalisation est la diminution de la complexité des traitements numériques (i.e. moins de termes à considérer lors des calculs). Dans notre cas, nous nous intéressons à la normalisation des dates et des numéros. Effectivement, se sont des informations qui sont représentées avec plus d'un token [3], pour cela, nous devons les identifier comme étant une seule entité.

4.3.1.5 Identification des mots vides

Les mots vides sont des mots non significatifs figurants dans un texte, qui sont tellement communs [4] qu'il est inutile de les indexer, nous les opposons aux mots pleins. Les mots vides sont principalement des mots caractéristiques à chaque langue comme

1. Le rang d'une phrase représente son numéro d'apparition dans le texte.
2. Une norme est un document établi par consensus et approuvé par un organisme de normalisation reconnu (ISO, CEI, UIT-T, ETSI, W3C, ...), ici, la normalisation consiste à associer à chaque type d'informations un format spécifique.
3. Un token est défini comme une suite de lettres comprise entre deux délimiteurs, transcrivant un son ou groupe de sons d'une langue auquel est associé un sens, et que les usagers de cette langue considèrent comme formant une unité autonome. Le délimiteur est le plus souvent le caractère espace, mais il peut s'agir d'autres éléments comme une suite d'espaces, une tabulation ou un signe de ponctuation (voire même parfois aucun espace comme en chinois ou en japonais).
4. Qui n'est pas propre à un concept précis tel : de, le, la...etc.

les prépositions, les articles et les pronoms, appelé aussi anti-dictionnaire. L'opération consiste à supprimer tous les mots vides de la langue anglaise, tirés de textfixer [5].

4.3.1.6 Lemmatisation des mots

C'est un processus morphologique permettant de regrouper les variantes d'un mot. En effet, nous pouvons trouver dans un texte différentes formes d'un mot désignant le même sens. Ils seront représentés par un seul mot désignant le concept véhiculé, appelé lemme en TAL, (ex : écologie, écologiste, écologique).

Nous distinguons parmi les principaux types de lemmatisation, l'analyse grammaticale en utilisant un dictionnaire [70], l'algorithme de Porter [6], que nous exploiterons lors de la lemmatisation des mots.

Algorithm 1 Algorithme du pré-traitement de l'e-mail.

Entrée: E : Email

Sortie: *CoprsXML* : Corps de l'email converti en XML

Début

1: Recuperer-info-entête() *//Fonction qui récupère et enregistre les informations de l'entête de l'email dans le modèle du profil utilisateur «adresse de l'émetteur, de récepteur et le sujet de l'email»*
La construction du fichier XML se fait avec les résultats des étapes 2,3,4 et 5.
2: Segmentation-phrase(). *//Fonction qui segmente le corps de l'e-mail en phrase*
3: Normalisation() *//Fonction qui retourne la forme normalisée des dates et des numéros.*
4: Elimination-mot-vide() *//Fonction qui élimine les mots vides du texte*
5: Lemmatisation() *//Fonction qui remplace chaque mot par son lemme.*

Fin

4.3.2 Module du résumé hybride

Résumer un texte consiste à le réduire en un nombre limité de mots afin de produire une représentation condensée, le texte ainsi réduit doit rester fidèle aux informations ainsi qu'à la sémantique du texte original. Contrairement à la plupart des approches de résumé se basant sur les techniques statistiques, nous proposons une approche hybride, en utilisant deux méthodes exploitant l'aspect formel offert par les statistiques ainsi que la sémantique des ontologies.

5. http ://www.textfixer.com/resources/common-english-words.txt Consulté le 26 Mars 2013.
6. http ://ir.dcs.gla.ac.uk/resources/linguistic-utils/porter.java Consulté le 15 Avril 2013.

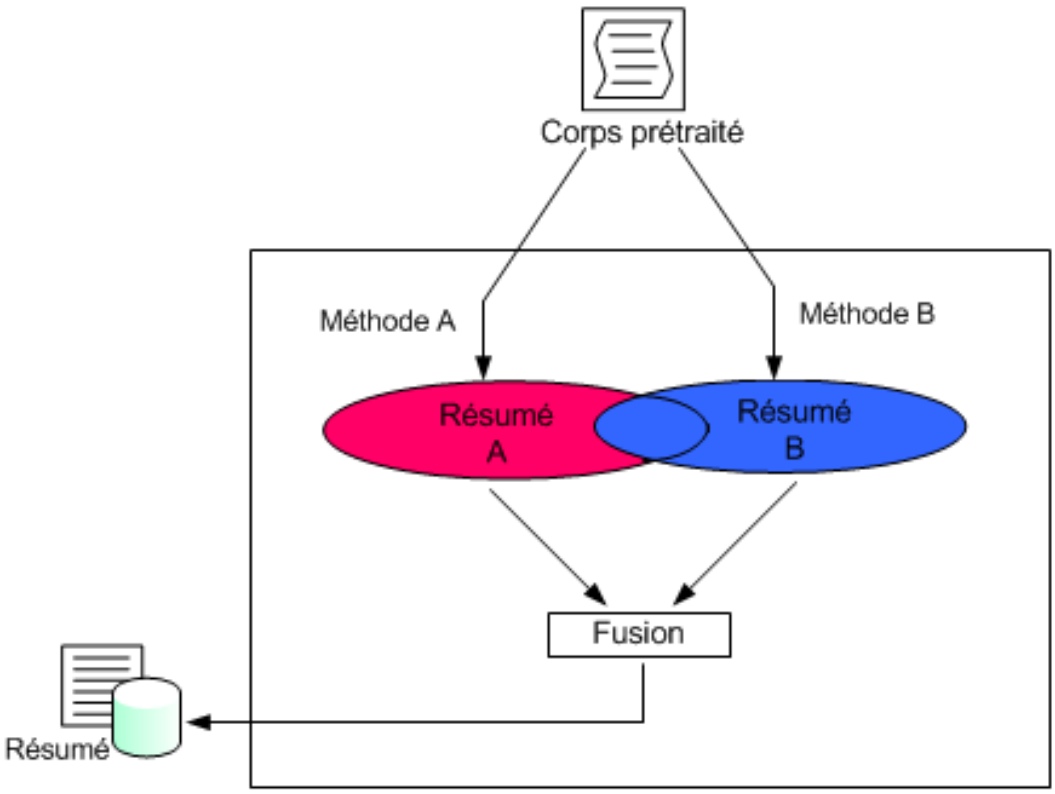


FIGURE 4.5 – Schématisation du module de résumé hybride.

Le module du résumé hybride, présenté par la figure 4.5, retourne le résultat d'une fusion de deux résumés obtenus de deux méthodes de résumé automatique distinctes, dans le but d'augmenter la quantité d'informations qui peut être considérée comme pertinente et minimiser la perte de celle-ci.

Le processus concerne l'application de deux méthodes de résumés :

– **Méthode A :** inspirée de celle de [24] que nous avons amélioré avec l'intégration des ontologies de domaines (Voir section 3.2.1).

– **Méthode B :** représente celle de [47] qui utilise un algorithme glouton(Voir section 3.2.2).

Dans ce qui suit, nous allons détailler les différents modules de l'architecture présentée par la figure 4.5.

4.3.2.1 Résumé produit par la méthode A

Les méthodes numériques sont les plus étudiées dans le domaine du traitement de la langue, tel le domaine du résumé automatique, [22, 44, 71] , il est cependant clair que ces techniques vont atteindre les limites de leur puissance. Les combiner avec d'autres

méthodes pourrait permettre de franchir un palier et de s'approcher un peu de ce que peut faire un humain.

Les expériences de [24], ont démontré qu'une combinaison de plusieurs métriques de pertinence est statiquement toujours plus performante que la meilleure des métriques, concluant ainsi, l'importance de toutes les métriques dans la pondération finale des phrases.

Pour notre système, nous allons exploiter les métriques statistiques suivantes, jugées intéressantes, apportées par le travail de [24] au système de résumé automatique YACHS [7] (*Yet Another Chemistry Summarizer*), utilisé dans le domaine de la chimie, ainsi que quelques unes connues par leur utilisation dans les systèmes de résumé automatiques. Pour cette méthode, en plus des procédés statistiques, nous allons exploiter la puissance des ontologies de domaines. Le texte à traiter "le corps de l'e-mail" sera représenté sous forme matricielle, en se basant sur l'idée de YACHS. C'est à partir de cette matrice que vont être calculées les métriques suivantes :

- **Similarité au titre :**

Vu son importance, selon l'hypothèse de [22], ainsi que les résultats de [24], nous utiliserons pour le calcul de cette métrique, la mesure bien connue de l'angle cosine [11], entre les représentations vectorielles du sujet de l'e-mail t= $(a_1, a_2, a_i, ...a_n)$ et de la phrase du corps S= $(s_1, s_2, s_i, ...s_n)$. La mesure de similarité sera exprimée en fonction de l'angle formé par les deux vecteurs représentant de ces derniers :

$$cosine(t, s_j) = \cos(\theta) = \frac{t * s_j}{||t|| * ||s_j||} \tag{4.1}$$

Tel que :

- ***** : Produit scalaire de deux vecteurs.
- $||\mathbf{t}||$: La norme du vecteur t.
- $||s_j||$: La norme du vecteur s_j (représentant de la phrase j).

En notant que les composantes vectorielles s_i et a_i représentent W_i(x), le poids tf*idf du terme i dans le document x [35], [71].

$$W_i(x) = tf_i(x) * \log \frac{N - n}{n} \tag{4.2}$$

Tel que :

- **N** : Nombre de phrases du texte étudié.

7. Version de démonstration disponible sur http ://daniel.iut.univ-metz.fr/yachs/

– **n** : Nombre de phrases dans les quelles apparaît le terme i.

– tf_i : Fréquence du terme i dans la phrase x, tel que :

$$tf_i(x) = \frac{t_i(x)}{f_i(x)} \tag{4.3}$$

Tel que :

– t_i**(x)** : Nombre d'occurrence du terme i dans la phrase x

– f_i**(x)** : Nombre de termes distincts de la phrase x.

• **Position de la phrase P :**

Notre approche consiste à traiter le contenu d'un e-mail électronique dans le domaine du recrutement, une demande d'emploi. Selon le format universel de cette dernière, nous constatons que les phrases appartenant au début et fin du texte ne sont que des formules de politesse, alors que l'essentiel de l'information envoyée se retrouve au milieu du document, donc devraient être favorisées afin de faire partie du résumé. Nous avons adapté la formule de [24] à notre type de document, d'où la fonction P(x) suivante :

$$P(x) = 1 - |\frac{x - \frac{m}{2}}{\frac{m}{2}}| \tag{4.4}$$

Tel que :

– **m** : Nombre de phrases du document.

• **Somme des fréquences des mots d'une phrase :**

L'une des métriques utilisées par les deux systèmes de résumé YACHS et Cortex (Cortex es OtroResumidor de TEXtos) [48], [49] est la somme des fréquences des mots d'une phrase x, F(x) . Elle représente le nombre de termes informatifs, en d'autres termes, c'est la somme des fréquences des termes restants après le pré-traitement pour chaque phrase :

$$F(x) = \sum_{y=1}^{n} A_{xy} \tag{4.5}$$

Tel que :

– A_{xy} : Représente la fréquence du mot y dans la phrase x.

• **Somme des poids des mots d'une phrase :**

T(x) est une métrique qui représente la somme des poids tf*idf de tous les termes de la phrase x, une métrique qui a fait ses preuves dans Cortex, formulée comme suit :

$$T(x) = \sum_{y=1}^{n} A_{xy} * idf(A_{xy}) \tag{4.6}$$

Tel que :

– A_{xy} *** idf (**A_{xy}**)** : correspond à la multiplication de la fréquence du terme x dans la phrase y, par son poids idf.

✍ ***Algorithme de décision :***

Pour décider de la pertinence d'une phrase j, nous appliquons l'algorithme de décision suivant, qui combine les valeurs normalisées de toutes les métriques statistiques calculées précédemment avec l'utilisation d'une fonction pré-définie, appelée PROTEGE qui nous permettra une validation sémantique.

Algorithm 2 Score-Décision

Entrée: *Lm* : Liste des N métriques statistiques pour chaque phrase p.
C1, C2 : constantes.
Z : Tableau vide de taille nombre de phrases du texte.

Sortie: *Z* : Tableau de scores des phrases.

Début

// Pour chaque phrase j

```
Z[j] :=0
pour i :=1 to N faire
    Z[j] :=Z[j] + Lm(i)
fin pour
pour k :=1 to longueur(phrase j) faire
    Valeur := PROTEGE(terme(k))
    si Valeur == Exact  alors
        Z[j] := Z[j]+ C1
    sinon si Valeur==Fail alors
        Z[j] := Z[j]+ C2
    fin si
fin pour
```

Fin

Les phrases à intégrer dans le résumé seront ainsi les phrases dépassant un score-seuil qui sera fixé à partir des exemples d'applications.

La méthode du résumé "A" nous permet de déterminer les informations pertinentes à un domaine donné grâce à l'utilisation des ontologies, ce qui permet ainsi, une extraction d'informations orientée vers un domaine spécifique en fonction de la sémantique des mots.

Dans le bute de minimiser la perte d'information, et assurer la cohérence entre phrases, nous proposons une hybridation de notre méthode avec une méthode des résumés déjà existante.

4.3.2.2 Résumé produit par la méthode B

Le deuxième résumé comporte un pré-traitement additionnel, c'est le filtrage des chiffres, pour ensuite, exécuter la méthode proposée dans [47], qui a abordé le problème du résumé comme étant un problème d'optimisation en utilisant les algorithmes gloutons, en se basant sur le système Cortex [25] , [49], l'un des systèmes de référence dans le domaine du résumé automatique, qui effectue une extraction non supervisée des phrases pertinentes en utilisant plusieurs métriques pilotées par un algorithme de décision.

Le texte prétraité sera représenté sous forme d'un graphe d'unités textuelles (ici, les phrases), ainsi l'importance d'un nœud n'est pas en fonction de son contenu mais de son emplacement en utilisant le principe de centralité.

Tel que :

– Il existe une arrête entre une phrase i et une phrase j, si et seulement s'il existe au moins un mot en commun entre les deux phrases.

– Chaque sommet a un poids, qui est représenté par le nombre d'arrêtes entrantes au sommet.

– Un degré est calculé pour chaque sommet, c'est le nombre de mots partagés avec les autres phrases.

La représentation des mots est produit par une matrice $S_{[P*N]}$ de fréquences/absences compsée de $\mu = 1, ..., P$ phrases (lignes) ; $\sigma_\mu = S_{\mu,1}$,..., $S_{\mu,i}$,..., $S_{\mu,N}$ et un vocabulaire de i= 1,..., N termes (colonnes).

$$\begin{pmatrix} \mathrm{S}_{1,1} & \mathrm{S}_{1,2} & ... & \mathrm{S}_{1,N} \\ \mathrm{S}_{2,1} & \mathrm{S}_{2,2} & ... & \mathrm{S}_{2,N} \\ . & . & ... & . \\ . & . & ... & . \\ \mathrm{S}_{P,1} & \mathrm{S}_{P,2} & ... & \mathrm{S}_{P,N} \end{pmatrix}$$

$$S_{i,j} = \begin{cases} TFi & si\ le\ terme\ j\ appartient\ \ la\ phrase\ i \\ 0 & sinon \end{cases} \tag{4.7}$$

Tel que :

– TF_i : La fréquence du terme i.

La présence du mot i est représenté par sa fréquence TFi (Son absence par 0 respectivement), et une phrase σ_μ est donc un vecteur de N occurences. S est une matrice entière car ses éléments prennent des valeurs fréquentielles absolues.

A partir de la matrice définit dans la relation (4.7), les auteurs ont utilisé une matrice A[P*P], tel que le calcul sera comme suit : parcourir la ligne i =1...P, et pour chaque élément $a_{i,j}$ égal à 1, descendre par la colonne j pour identifier d'autres phrases qui partagent ce mot. $a_{i,j} = 1$ si un mot présent dans la phrase i l'est aussi dans la phrase j ; 0 autrement.

Ainsi pour trouver les phrases les plus lourdes, ceci revient à chercher une variante du problème de l'arbre de poids maximum. Pour cela, l'algorithme glouton est comme suit :

Algorithm 3 Resumeur-Glouton

Entrée: L : Le nombre de phrases à prendre dans le résumé.
U=V : Qui correspond à l'ensemble des nœuds du graphe (les phrases).
Degré de chaque sommet, T = ∞
//Trier les arêtes de G par ordre croissant du poids w.

Sortie: *Rendre la séquence des sommets calculée.*

Début
// Ajouter à T les éléments de la liste ordonnée comme suit :
1: T=V(i) ; i=1
2: **si** l'arête (V(i),V(i+1)) existe **alors**
3: ajouter V(i+1) à T
4: V(i+1) est dans U
5: T=T ∪ V(i+1)
6: U= U - V(i+1)
7: **sinon**
8: aller à 6
9: **fin si**
10: Faire i := i + 1
11: **si** | T | = L **alors**
12: arrêter.
13: **sinon**
14: aller à 5.
15: **fin si**//Retourner la séquence des sommets traités.
Fin

4.3.2.3 Fusion des deux résumés

Le résumé du corps de l'e-mail sera le résultat de la fusion des deux résumés, ç-à-d, l'ensemble des phrases avec élimination des phrases redondantes, en appliquant l'algorithme de fusion suivant :

Algorithm 4 Fusion

Entrée: *Résumé-A :* le résumé obtenu par la méthode A.
Résumé-B : le résumé obtenu par la méthode B.

Sortie: *Résumé-hybride.*

Début

1: Résumé-hybride = Résumé-A
2: **pour** chaque phrase i appartenant à Résumé-B **faire**
3: **si** phrase i n'appartient pas à Résumé-A **alors**
4: Ajouter phrase i à Résumé-hybride
5: **fin si**
6: **fin pour**

Fin

4.3.3 Module de détection des entités nommées

Après avoir structurer les phrases les plus pertinentes, nous exploitons les travaux de [3].

Cette phase permet la détermination des entités nommées, qui représentent les différents types d'attributs du profil utilisateur, que nous avons classées en quatre catégories :

- **Nom (particulier/collectif)** : Conserne les attributs des dimensions du modèle de profil suivantes :
  - *Données personnelles* : Name, Nationality, Family-Situation.
  - *Education* : Institution, Degree, Domain.
  - *Expérience-Professionnelle* : Company, JobTitle.
- **Information spatiale** : Conserne les attributs des dimensions suivantes :
  - *Données personnelles* : BirthPlace, Residence.
  - *Expérience-Professionnelle* : Location.
  - *Préférence-Utilisateur* : Area.
- **Information temporelle** : Correspond aux attributs des dimensions suivantes :
  - *Données personnelles* : BirthDate.
  - *Education* : StartDate, EndDate.
  - *Expérience-Professionnelle* : StartDate, EndDate, Duree.
- **Information numérique** : Correspond aux attributs des dimensions suivantes :
  - *Données personnelles* : Age.
  - *Préférence-Utilisateur* : Salary.

Nous proposons dans la figure 4.6 l'architecture de ce module :

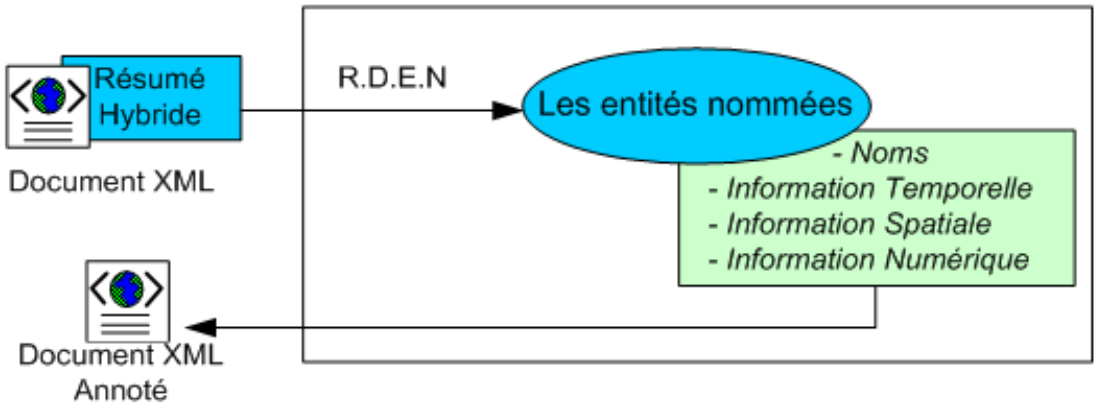


FIGURE 4.6 – Représentation du module de détection des entités nommées.

La détection des entités nommées, a pour but de localiser et de classer les éléments du texte dans des catégories prédéfinies correspondantes aux attributs du profil utilisateur. Pour cela, nous utilisons un ensemble de Règles de Détection d'Entités Nommées (R.D.E.N), appelées aussi, règles d'exploration contextuelle, tout en s'inspirant des travaux de [3].

4.3.3.1 Règles d'exploration contextuelles

Les règles d'exploration contextuelle s'appliquent à des portions textuelles, pour cela la définition de la notion d'espace de recherche, une règle est divisée en trois parties : une partie Déclaration d'un Espace de Recherche E, une partie Condition et une Action qui n'est déclenchée que si la partie Condition est vérifiée.

- **Déclaration d'un espace de Recherche : E**

Elle permet de construire un segment textuel, l'espace de recherche, en appliquant différentes opérations sur la structure du texte. Dans notre cas, nous avons choisi de définir comme espace de recherche la phrase pour toutes les règles.

- **Partie Condition : C**

Elle explicite les conditions que doivent vérifier les indicateurs, comme l'existence, la position et l'agencement de ceux ci. D'autres conditions permettent d'exprimer des contraintes sur les attributs des unités lexicales. Nous supposons que tous les indicateurs sont balisés au départ.

- **Partie Action : A**

Elle indique le type d'actions réalisées par la règle. Dans notre cas, nous nous intéressons à l'attribution d'une annoation à un segment textuel, en d'autres termes, l'attribution d'une balise à une information (mot ou ensemble de mots).

Le résultat de cette étape est la structure présenté dans la figure 4.7 :

```
<?xml version ="1.0"encoding="UTF-8"standalone="no"?>
<Email Id =" " Sender =" " Receiver=" ">
<Phrase id="">                          <!-- Numéro de la phrase (La position de la phrase dans le texte global) -->
<!-- Cette partie sera remplis lors du module de Détermination des Entités Nommées -->

        <NamedEntity id="">
        <Word id=" "> </Word>      <!-- Contient les indentificateurs constituants l'entité nommée -->
          <Type> </Type>                    <!-- Contient le nom de l'entité nommée -->
      lamedEntity>

<!-- Cette partie sera remplie lors du module du Matching et Détermination des Unités d'Informations -->
          <Ulid="">
          <IdPhrase></IdPhrase>             <!-- Identificateur ou la position de la phrase -->
          <Subject></Subject>               <!-- Le sujet -->
```

FIGURE 4.7 – Extrait du fichier XML concernant le module de détection des EN.

En s'inspirant du modèle de [3], nous déclarons des listes de mots, appelés indicateurs qui seront annotés préalablement (ex : déterminants, pays, fonctions, diplômes...etc). Pour ensuite appliquer un ensemble de règles, afin de déterminer les différentes catégories d'entités nommées.

La figure 4.8 illustre un extrait du diagramme général des entités nommées considérées :

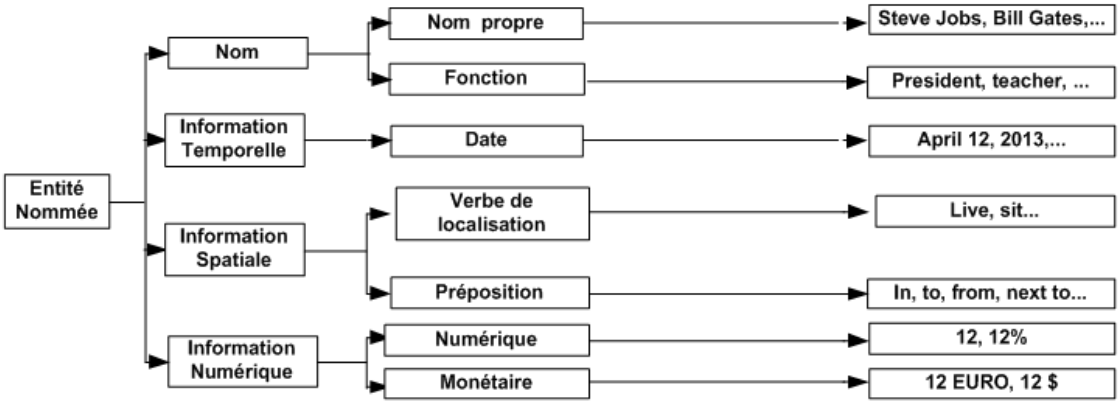


FIGURE 4.8 – Extrait du diagramme général des EN.

Nous utilisons cette représentation afin de repérer toute information exprimant une entité nommée (une information du profil utilisateur) dans le document textuel. Dans la figure 4.9 un exemple de règles définissant quelques une de ces informations :

Exemples de règles d'exploration contextuelle

Condition

// Nom Propre
<Det>? & (<Title>? | <Fonc>? | <Natn>?) & <MotMaj>+
// règle de détermination d'une information temporelle
<Prep>? & <JourSemaine>? & <Mois>? & <NbreJour>? & <NbreAnné>?
// règle de détermination d'une information spatiale
<PrepLieu> & <Contr>

Action

Annoter <Nom> | <Date> | <Lieu>

FIGURE 4.9 – Exemple d'application des règles d'annotation des EN.

Prenons comme exemple la phrase " I met the professor Lee Izuki in China... ". Le résultat de la détermination des entités nommées de cette phrase est illustré par la figure 4.10 :

```
<?xml version="1.0"encoding="UTF-8"standalone="no"?>
<Email Id = " " Sender = "" Receiver = " ">
<Phraseid="5 ">                    <!-- Numéro de la phrase (La position de la phrase dans le texte global)-->

<!-- Cette partie sera remplis lors du module du détermination des entités nommées  -->
<OriginalPhrase> ...I  met the professor Lee Izuki  in  China  ... </OriginalPhrase>

        <NamedEntity  id=" 1">              <!—L'identificateur de l'entité nommée -- >
        <Word id="5"> Lee </Word>           <!—L'identificateur des termes composants les entités nommées -- >
        <Word id= '' 6'> Izuki <Word>
        <Type>Acteur </Type>                <! – Le type de l'entité nommée -- >
        <NormalizedEntity>Lee Izuki</NormalizedEntity>
</NamedEntity>

<NamedEntity  id=" 2">
        <Word id="8"> China </Word>         <!—L'identificateur de l'entité nommée -- >
        <Type>Lieu</Type>                   <!—L'identificateur des termes composants les entités nommées -- >
<NormalizedEntity>China</NormalizedEntity><! – Le type de l'entité nommée -- >
        </NamedEntity>
```

FIGURE 4.10 – Extrait du fichier XML représentant le texte annoté.

4.3.4 Module de Matching avec les informations du profil utilisateur

Dans le but d'extraire le profil utilisateur à partir d'une source textuelle, nous considérons que les informations concernant les attributs du profil sont exprimées par des segments guidées par des éléments déclencheurs et des indicateurs sémantiques.

Une phrase peut véhiculer plusieurs idées et informations à la fois, exprimées de différentes manières. Il est donc difficile d'extraire les informations concernant le profil

utilisateur, cette difficulté revient à la complexité des relations sémantiques entre les différents composants d'une phrase. Afin de remédier à ce problème, nous nous inspirons des travaux des auteurs [30] dans le domaine de résumé automatique qui se basent sur une technique de génération d'un résumé à partir des éléments d'information sous la forme Sujet-Verbe-Objet. Dans notre approche, nous proposons de structurer la plus petite information véhiculée dans une phrase sur le profil utilisateur sous la forme " *Sujet - Verbe - Objet (Valeur, AttributConcurrents, Contexte)* " . En ayant une structure cohérente et une représentation unifiée des informations que comporte une phrase, nous pouvons concevoir des règles génériques [8] pour l'extraction des attributs du profil utilisateur, où chaque attribut est lié à un sujet spécifique avec l'intermédiaire d'un élément déclencheur.

✍ **Définition d'Unité d'Information :** Une unité d'information est l'élément le plus petit comportant une information cohérente sur le profil utilisateur dans une phrase. Une unité d'information **UI** = (Sujet, Verbe, Objet) Avec Objet = (Valeur, AttributConcurrents, Contexte).

• **Sujet** : est représenté par tout terme ou entité désignant celui qui est concerné par l'attribut. L'identification du sujet et de ses attributs va réduire considérablement le temps de recherche, puisque nous nous intéressons uniquement aux informations le concernant.

• **Verbe** : on appelle aussi *élément déclencheur*. Il est représenté par tout terme exprimant l'événement accomplit par le sujet. Le verbe est d'une importance primordiale dans une phrase ou dans une unité d'information, car il représente un lien sémantique entre le sujet et l'attribut de celui-ci.

• **Objet** : Comporte l'information du profil à extraire, l'ensemble des attributs concurrents, et les indicateurs sémantiques liés à chaque attribut (Contexte), qui vont nous permettre de mieux identifier celui-ci.

La détermination des valeurs des attributs du profil utilisateur revient à :

- *Détermination des unités d'informations.*
- *Sélection des attributs candidats.*
- *Validation.*

La figure 4.11 illustre les différentes étapes de ce module :

8. Règles génériques : permettent d'écrire en une seule règle un ensemble de règles explicites très semblables.

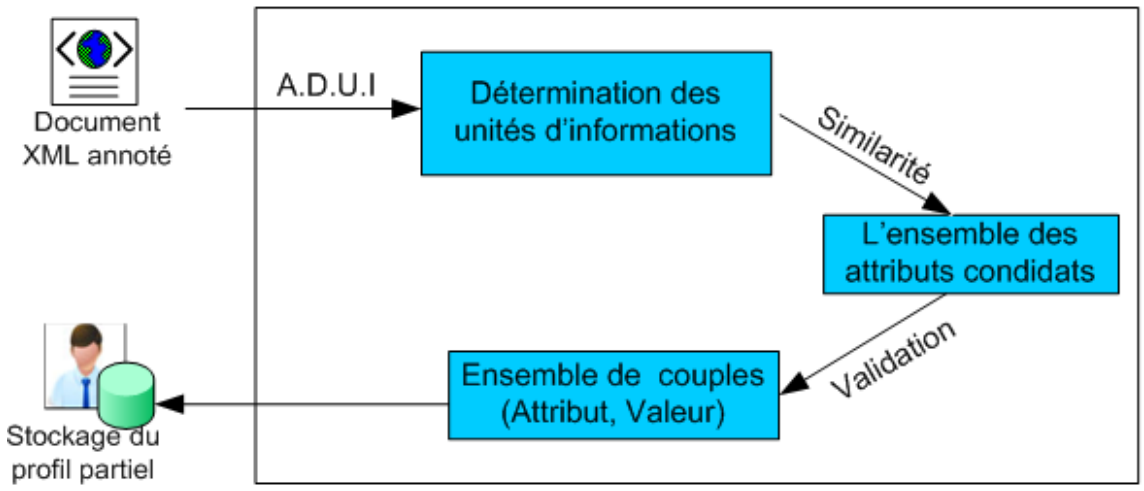


FIGURE 4.11 – Schématisation du Module de Matching.

a. Détermination des unités d'informations : pour cela, nous proposons un algorithme de traitement de dépendances entre termes (relations grammaticales entre termes), qui sont représentées par des prédicats sous la forme PREDICAT (Mot1, Mot2) ayant comme valeur Vrai si la relation entre les deux termes existe, Faux si elle n'existe pas. Dans ce cadre, nous exploiterons deux classes de dépendances ; les dépendances récupérant le sujet ainsi que celles récupérant les objets de ce derniers. Cet algorithme retourne les différents composants de l'unité d'information : Sujet, Verbe déclencheur et la valeur de l'objet, selon la nature d'information de celui-ci, nous déterminerons l'ensemble des attributs concurrents.

L'algorithme 5, représente l'algorithme proposé pour la récupération des unités d'informations :

Algorithm 5 Acquisition-UI

Entrée: *Ph :* Phrase tel que Ph= W_i

// Ph est une phrase composée de mots W_i

PreSujet : L'ensemble des prédicats déterminant les dépendances entre un verbe déclencheur et son sujet.

PreObjet : L'ensemble des prédicats déterminant les dépendances entre verbe déclencheur et ses objets.

Decl : L'ensemble des verbes déclencheurs.

LA : Liste des attributs du profil utilisateur.

Sortie: *UI : L'ensemble des unités d'informations.*

Début

1: **pour** chaque mot W_i de la phrase Ph **faire**
2: **si** Wi ∈ Decl **alors**
3: **si** (∃ Pri ∈ PreSujet) ET (∃ un mot W_s ∈ Ph)tel que Pri (Wi, Ws)== Vrai **alors**
4: **si** (∃ Prj ∈ PreObjet) ET (∃ un mot W_k ∈ ph)tel que Prj (Wi, Wk)== Vrai **alors**
5: C1 = ContextGauche (W_i) ∪ ContextDroit (W_i)
// ContextGauche(X) retourne le contexte gauche du terme X
//ContextDroit(X) retourne le contexte droit du terme X
6: C2 = ContextGauche(W_k) ∪ ContextDroit(W_k)
7: C = C1 ∪ C2 ∪ W_s ∪ W_k
8: ListAttribut = getListAttributType(Wk) ∩ getListAttributDecl(Wi)
// getListAttributType(x) retourne l'ensemble des attributs ayant le même type que x.
// getListAttributDecl(x) retourne l'ensemble des attributs ayant x comme verbe déclencheur.
9: UI.Sujet= W_s ; UI.Verbe= W_i ; UI.Objet=(W_k, ListAttribut, C)
10: **fin si**
11: **fin si**
12: **fin si**
13: **fin pour**
14: **return** (UI)

Fin

b. Elimination des attributs concurrents et sélection des attributs candidats : Consiste à retourner un attribut candidat, en fonction de la similarité des verbes

déclencheurs et des indicateurs sémantiques de chaque attribut avec les termes constituant les unités d'informations, pour les quelles nous avons déterminé les contextes gauches et droits des objets et des verbes déclencheurs. Tel que l'attribut candidat maximisera la fonction de la similarité. La démarche est détaillée par l'algorithme 6 :

Algorithm 6 Sélection

Entrée: *Ph :* Phrase

UI : L'unité d'information //UI=(Sujet, Verbe, (Valeur, ListAttribut, C)).

DECL(x) : Liste des déclencheurs de l'attribut x.

IND(x) : Liste des indicateurs de l'attribut x.

Sortie: *AttrCand :* L'ensemble des attributs Candidats.

Début

1: **pour** UI_t de la phrase ph **faire**
2: **pour** chaque $A_i \in$ Objet.ListAttribut **faire**
3: Li= $\{Mz/Mz \in (Decl(A_i) \cup IND(A_i)\}$
4: Similarité $(A_i, C) = \frac{1}{|Li|}\sum_{z=1}^{|Li|}[\frac{1}{|C|}\sum_{j=1}^{|C|} Simil(M_z, C_j)]$
// Simil(x, y) : une fonction qui calcule le degré de similitude entre x et y
5: **fin pour**
6: AttrCand = Maximum (Similarité (A_i, C))
// Maximum (S_i) : une fonction qui retourne le maximum des S_i
7: **return** AttrCand
8: **fin pour**

Fin

c. Validation : Nous proposons un ensemble de règles d'EC [9], dans le but de la validation de l'attribut candidat retourné lors de l'étape précédente, en appliquant la règle adéquate pour confirmer l'information (Utilisation des indicateurs dans les règles). Pour ce faire, nous nous inspirons de l'approche de [3] et nous adoptons la nomination d'UPennTreeBank II [10], l'une des annotations standard dans le domaine du traitement de langue naturelle.

Nous proposons des règles d'exploration contextuelles génériques regroupées selon le type d'information :

- **Information de type " Nom " :** *Name, Nationality, FamilySituation, JobTitle, Degree, Domain, NameInstitution, Company.*

9. EC : règles d'Exploration Contextuelles
10. Réalisé par M.P.Marcus, B.Santorini et M.A.Marcinkiewicz pour le traitement du langage naturel (langue anglaise)

✍ (PRP|PRP$) & (INDparticulier) ? & (DECparticulier) & (INDparticulier) ? & (Title | IN) ? & (Name) & (INDpariculier) ?

✍ (PRP) & (DECcollectif) & (*PRP*) ? & (IN|JJ) ? & (INDcollectif) & (IN) & (Name).

✍ (PRP) & (DECnumérique) & (Numérique) & (INDage) ?

Exemple 4.3.1. – *I have my own office in IBM.*

– *I have got a training in Network maintenance.*

– *I have two children.*

- **Information de type " Temporelle" :** *BirthDate, StartDate, EndDate, Duree, Age.*

✍ (PRP) & (NN|JJ) ? & (DECtemporel) & (IN) & (DATE).

✍ (PRP) & (DECtemporel) & (PRP$) & (INDeducation | INDexprérience) ? & (IN) & (Date| Periode).

Exemple 4.3.2. – *I worked for January 1, 2011 to June 19, 2013.*

– *I have start my job in June.*

- **Information de type " Spatiale " :** *BirthPlace, Résidence, Location, AreaPref.*

✍ (PRP) & (DECspatiale) & (IN) ? & (INDspatiale) ? & (IN) & (LIEU) & (INDspatiale) ?

✍ (PRP) & (DECspatiale) & (IN) ? & (INDspatiale) ? & (IN) ? & (LIEU).

Exemple 4.3.3. – *I live in London city.*

– *I love London.*

- **Information de type " Numérique " :** *Age, PhoneNumber, Salary.*

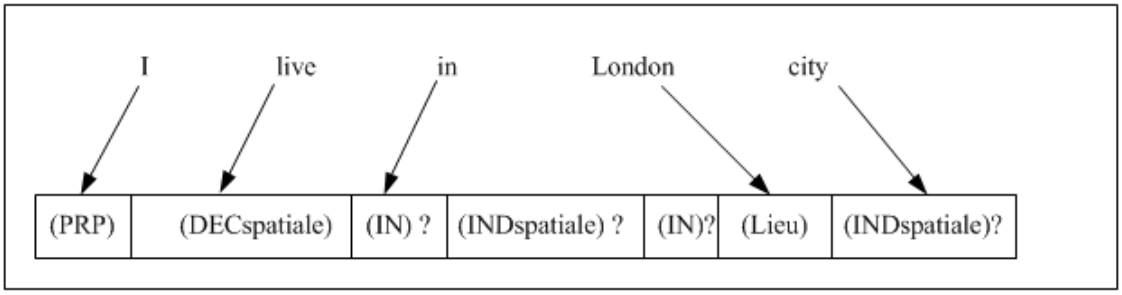
✍ (PRP) & (DECnumérique) & (Numérique) & (INDage) ?

✍ (PRP|PRP$) & (INDphoneNumber) ? & (DECnumérique) & (INDphoneNumber) ? & (Numérique).

Exemple 4.3.4. – *I have 20.*

– *My phone number is 001 432 989 439*

- **Exemple d'application des règles :**



Durant l'étape de la validation, nous constatons deux scénarios :

• **Cas1 : Un seul attribut candidat** : L'étape de la validation consiste à faire correspondre la règle adéquate à l'unité d'information.

• **Cas2 : Plusieurs attributs candidats** : Ici, nous exploitons les indicateurs sémantiques qui vont nous servir comme des éléments de décision sur l'attribut à choisir.

L'algorithme 7 illustre les détails du traitement :

Algorithm 7 Validation

Entrée: *UI :* L'unité d'information, UI=(Sujet Verbe, Objet(Valeur, ListAttribut, C))
AttrCand : L'ensemble des attributs candidats
RegEC : L'ensemble des règles d'exploration contextuelle.

Sortie: *AttrGagn :* L'attribut gagnant.

Begin

1: RègApp = $\{R_j / R_j \in$ *catgorie des rgles correspondante au type de l'attribut candidat*$\}$
2: **Si** Card(AttrCand)==1 **Alors** //Card(x) retourne la cardinalité de l'ensemble x
3: **Si** ($\exists$ $R_j \in$ RègApp) ET (Application (R_j , UI)== Vrai) **Alors** // Application(R,U) : prédicat de vérification de la correspondance entre la règle R et le segment textuelle U.
4: AttrGagn=AttrCand
5: **return** (AttrGagn)
6: **Finsi**
7: **Sinon** ($\exists$ Rj $\in$ RègApp) ET (Application (Rj , UI)== Vrai)
8: AttrGagn= AttrNbrInd(ListAttribut, C)
9: **return** (AttrGagn)
//AttrNbrInd(ListAttribut, C) : Foction qui retourne l'attribut ayant le maximum d'indicateurs appartenant à C.
10: **Finsi**

Fin

4.3.5 Module de mise à jour

Suite à la caractéristique d'évolution en fonction du temps, qui caractérise le profil utilisateur. Nous proposons le module de mise à jour, qui a pour but : l'actualisation des informations du profil utilisateur au fur et à mesure de la réception d'e-mails.

Après le traitement du premier e-mail, nous obtenons le premier profil, qui sera stocké dans la base de connaissance comme un profil initial.

A l'arrivée du suivant e-mail, nous procédons aux étapes suivantes :

1. Vérifier l'émetteur et le récepteur de l'e-mail, s'ils sont les mêmes, ainsi que le sujet de l'e-mail.
2. Si oui, alors suivre la même procédure du traitement de l'email pour un nouveau profil, et lancer la procédure de mise à jour.
3. Si non, retour à 1.

La figure 4.12 représente le module de mise à jour :

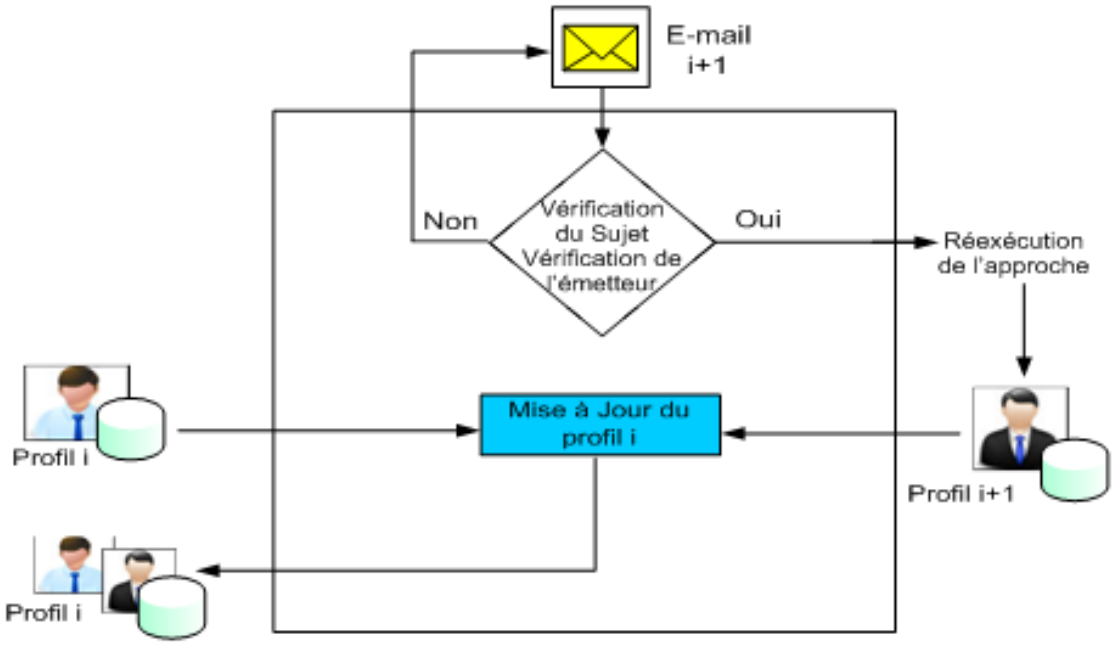


FIGURE 4.12 – Module de mise à jour.

La phase de la mise à jour se fait de trois manières, en fonction de la nature de l'attribut :

• **Les informations statiques :** Cette catégorie d'attributs n'est pas concernée par le processus de mise à jour. Ces informations ont un rapport avec les attributs des dimensions du profil suivantes :

– *Données personnelles* : Name, BirthDate, BirthPlace.

– *Education* : StartDate, EndDate.

La mise à jour de ce type d'information se fait comme suit :

$$\forall\ j, \forall\ A_i, A_i \in InfoStatique, Profil_{j+1}(A_i) == Profil_j(A_i).$$

• **Les informations dynamiques :** Dans ce cas, pour chaque attribut nous associons la valeur la plus récente, en d'autres termes, nous associons la valeur obtenue du dernier profil déterminé. Les attributs concernant par ce type d'informations sont :

- *Données personnelles* : Age, Family-Situation, Residence.
- *Experience-Professionnelle* : Company, JobTitle, Location, StartDate, EndDate.
- *Preference-Utilisateur* : Salary.

La mise à jour se fait comme suit :

$$\forall\ j, \forall\ A_i, A_i \in InfoDynamique, Profil_j(A_i) \leftarrow Profil_{j+1}(A_i).$$

• **Les informations évolutives :** Les valeurs de ce type d'attributs sont représentées par l'union des valeurs obtenues de chaque profil partiel (intermédiaire). Telles les attributs des dimensions suivantes :

- *Données personnelles* : Nationality.
- *Education* : Institution, Degree, Domain.
- *Preference-Utilisateur* : Area.

La mise à jour de ce type d'information est comme suit :

$$\forall\ j, \forall\ A_i, A_i \in InfoEvolutive, Profil_{j+1}(A_i) \leftarrow Profil_j(A_i) \bigcup Profil_{j+1}(A_i).$$

4.4 Mise en œuvre du système proposé

L'objectif de notre travaille est d'élaborer un système capable de construire le profil utilisateur à partir d'un document textuel. Et ceci en combinant un ensemble de métriques statistiques avec la puissance des ontologies de domaine ainsi que les relations grammaticale entre termes. Ayant comme but, la validation de notre approche, nous sommes dans l'obligation d'étudier quelques cas d'exécution (scénarios) en présentant les résultats de chaque module de notre système.

4.4.1 Environnement de travail

Dans le but d'atteindre notre objectif et de valider notre système proposé, nous présentons un prototype, en respectant un ensemble de contraintes :

Les hypothèses sont :

- Les e-mails à traiter sont écrits en anglais.
- Le corps de l'e-mail n'est pas un fichier joint.
- L'information est exprimé grâce à une seul phrase, ç-à-d, l'information n'est pas dispatchée (sur plusieurs phrases).
- Les phrases traitées ne contiennent pas de la négation.
- L'utilisation d'une ontologie existante dans le domaine du recrutement informatique, des connaissances linguistiques regroupées dans des classes sémantiques, représentant l'ensemble des indicateurs et verbes déclencheurs de chaque attribut.

Les objectifs visés sont :

- Le système doit être capable de reconnaitre l'ensemble des entités nommées (nom propre, information temporelle, information spatiale, . . .etc.), qui représentent l'ensemble des éventuelles valeurs du profil utilisateur.
- Le système doit retourner un fichier XML où tous les mots du texte seront balisés (annotés).
- Le système doit retourner l'ensemble des informations concernant l'utilisateur citées dans l'e-mail traité, le profil utilisateur.
- Le système retourne un fichier XML qui représente, une structuration du profil utilisateur.

4.4.2 Les outils de mise en œuvre

Dans cette partie, nous présentons les différents outils nécessaires pour la mise en œuvre et l'implémentation de notre approche. Pour cela, nous avons opté pour l'utilisation d'un ensemble d'outil de développement suivant :

• **Eclipse 3.3 :** qui est un environnement de développement permettant la construction des applications JAVA.

• **L'API JDOM** (Java Document Object Model) : qui se base sur le traitement hiérarchique (en arbre) des documents XML. DOM est une recommandation de W3C qui décrit une interface indépendante de tout langage de programmation et de toute plateforme, permettant à des programmes informatiques et à des scripts d'accéder ou de mettre à jour le contenu, la structure ou le style de documents XML. Le document peut ensuite être traité et les résultats de ces traitements peuvent être réincorporés dans le document tel qu'il sera présenté.

• L'approche consiste à utiliser une ontologie de domaine qui sera crée en utilisant l'éditeur PROTEGE, ainsi que Appache Jena, pour l'interrogation de cette ontologie, qui est un Framework Java pour le développement des applications Web sémantique. Pour

faciliter notre implémentation nous avons opté pour l'utilisation d'un ensemble de liste de termes organisée concernant quelques attributs du profil utilisateur (Degree, Domain, Nationality,..etc).

• Le manuel **Stanford** [11] : qui représente les dépendances et les relations textuelles. Il permet de fournir une description simple des rapports grammaticaux dans une phrase .

• L'annotation *Penn tree bank II* ou bien appelée aussi Part Of Speach (POS).

• L'utilisation de **WordNet : : Similarity** 2.05, un outil de calcul de similarité entre deux termes en utilisant la base de connaissance *WordNet*.

La figure 4.13 représente les différents outils utilisés pour l'implémentation de notre approche :

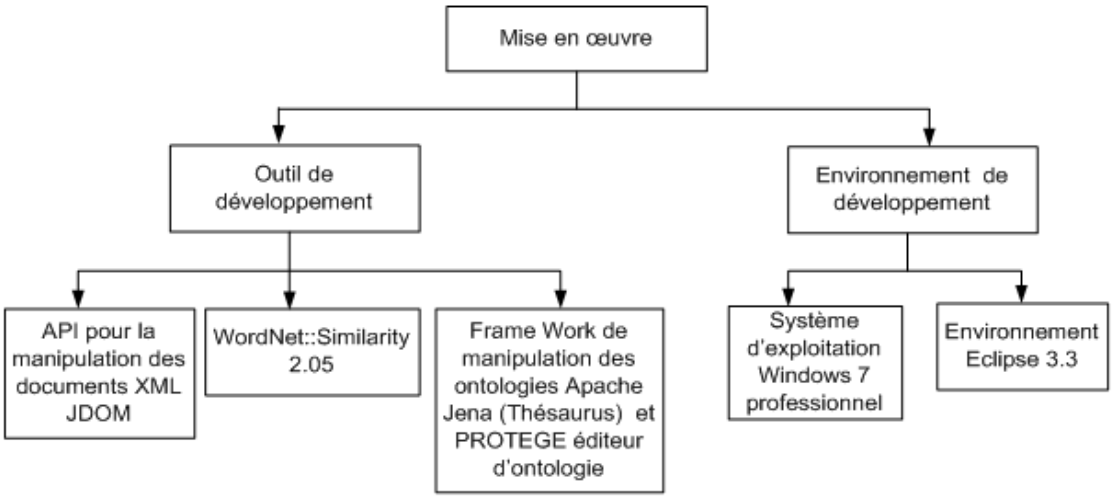


FIGURE 4.13 – Les outils utilisés pour la mise en œuvre du système propose.

4.4.3 Modèle du profil utilisateur

La figure 4.14 représente le modèle UML du profil utilisateur, qui se compose de quatre dimensions, chacune comporte un ensemble d'attributs et liée à une ontologie de domaine.

11. http ://nlp.stanford.edu/software/stanford-dependencies.shtml

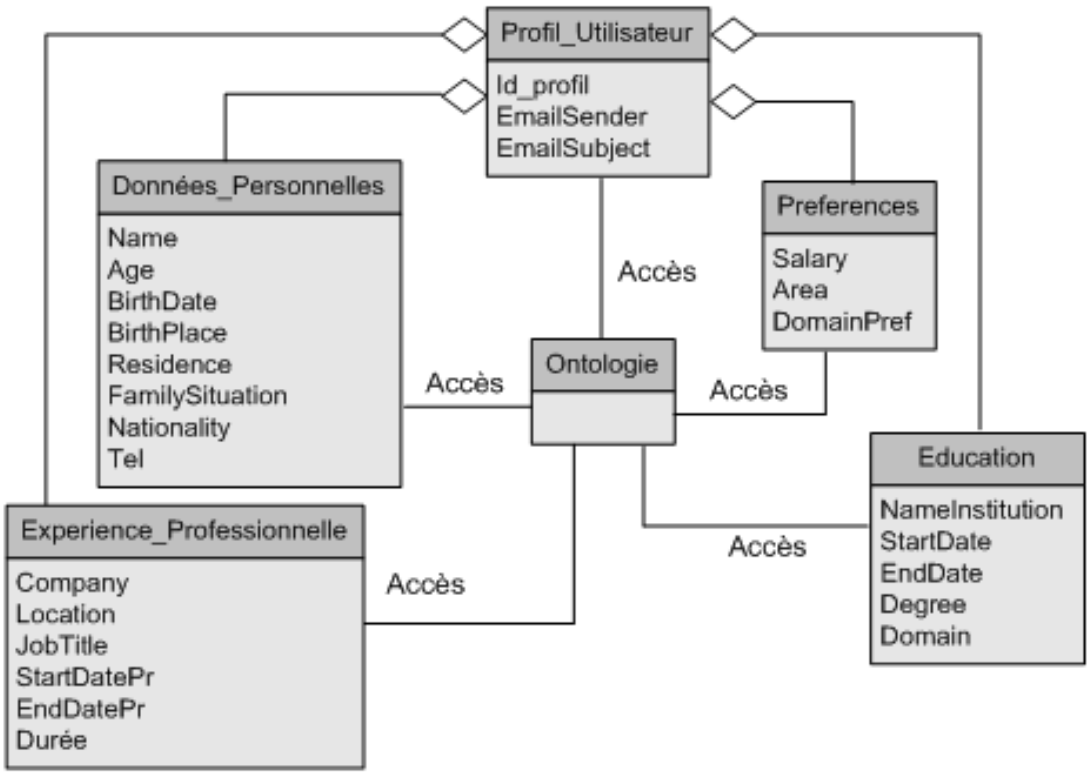


FIGURE 4.14 – Diagramme de classe du profil utilisateur.

4.4.4 Processus d'exécution

La figure 4.15, résume le fonctionnement de l'approche proposée dans le processus suivant :

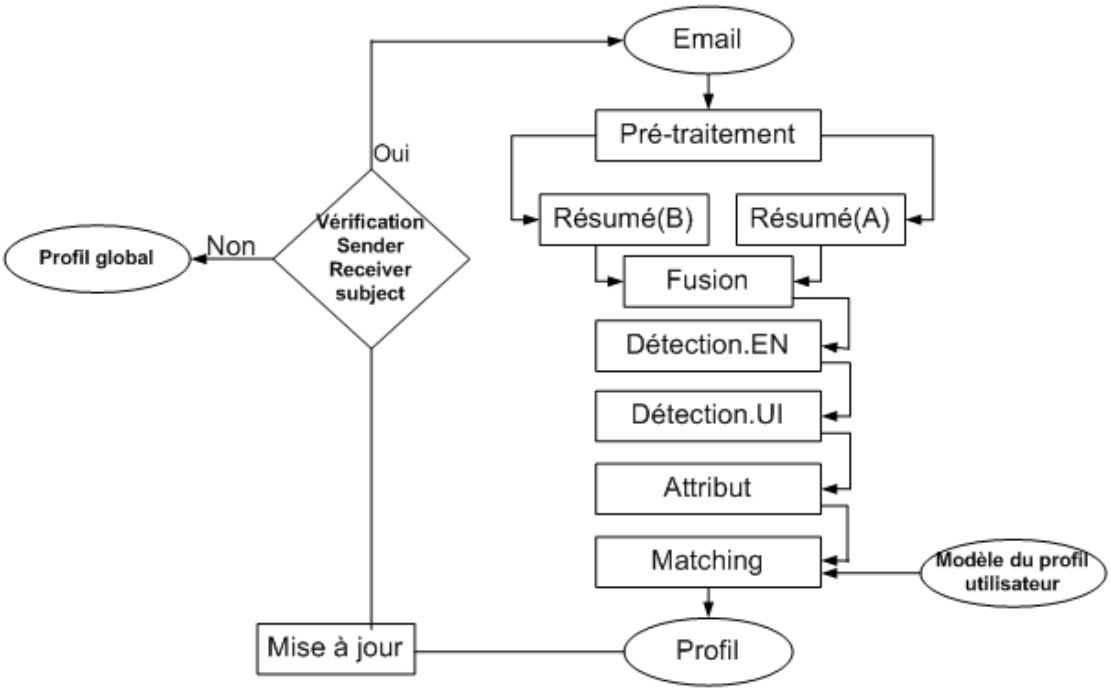


FIGURE 4.15 – Le processus d'exécution de l'approche proposée.

4.4.5 Les scénarios d'exécution

Dans le but d'illustrer le fonctionnement de notre système, nous proposons deux scénarios d'exécution en utilisant l'exemple d'e-mail suivant :

Sender: johnsmith@gmail.com
Seceiver: computing_company@yahoo.fr
Subject: Science computing domain

Dear ,
I am writing this email after reading your announce in The Sun newspaper, in order to join your great company.
I am Mr.John Smith, I come from Algeria. I was born on April 30,1990. I was graduated from MIT in June, 2013, I have a Magister degree. Actually, I live in London, i have worked with IBM company for three years, as a developer. I am interested on the Web, Intelligence Artificial and programming. I can be contacted at 001 024 234 342.

Thank you for your time and consideration, I look forward to speaking with you about this employment opportunity,
Sincerely.

FIGURE 4.16 – Exemple d'Email.

4.4.5.1 Module de prétraitement

Ce module consiste à récupérer l'adresse de l'émetteur, l'adresse du récepteur, ainsi que le sujet de l'email. Pour ensuite pré-traiter le corps de l'e-mail par la :

- Segmentation en phrase.
- Normalisation des dates et des numéros.
- Elimination des mots vides.
- Lemmatisation des termes.

La figure 4.17 représente un fichier XML qui est le résultat de la phase du prétraitement :

```
<? xml version ="1.0" encoding ="UTF-8" standalone ="no" ?>
<Email Id =" 1" Sender =" Johnsmith@gmail.com" Receiver="computing_company@yahoo.fr " Subject=" Science computing domain">

<Phrase id=" 2 ">                                   <!-- Numéro de la phrase (La position de la phrase dans le texte global) -->
        <!-- Cette partie sera remplis lors du module du prétraitement -->
  <OriginalPhrase> I am Mr. John Smith, I come from Algeria </OriginalPhrase>
    <Word id=" 1 "> <Name> I </Name>                <!-- Le mot dans sa forme originale dans la phrase -->
          <Lemma> I </Lemma>                        <!-- lemme, Après lemmatisation -->
          <POS> PRP </POS>                          <!-- L'annotation utilisée par Part Of Speach -->
          <Sign> -- </Sign>                         <!-- L'annotation utilisée pour les termes déclarés comme indicateurs -->
          <EmptyWord> No </EmptyWord>               <!—Identification du mot vide -->
    </Word>
    <Word id=" 2 "> <Name> am </Name>               <!-- Le mot dans sa forme originale dans la phrase -->
          <Lemma> be</Lemma>                        <!-- lemme, Après lemmatisation -->
          <POS> VBP</POS>                           <!-- L'annotation utilisée par Part Of Speach -->
          <Sign> -- </Sign>
          <EmptyWord> Yes </EmptyWord>
    </Word>
    <Word id=" 3"> <Name> Mr </Name>                <!-- Le mot dans sa forme originale dans la phrase -->
          <Lemma> Mr</Lemma>                        <!-- lemme, Après lemmatisation -->
          <POS> NNP </POS>                          <!-- L'annotation utilisée par Part Of Speach -->
          <Sign> Title </Sign>
          <EmptyWord> No </EmptyWord>
    </Word>
```

```
<Word id=" 4 "> <Name> John </Name>                <!-- Le mot dans sa forme originale dans la phrase -->
        <Lemma>John </Lemma>                       <!-- lemme, Après lemmatisation -->
        <POS> NNP </POS>                           <!-- L'annotation utilisée par Part Of Speach -->
        <Sign> -- </Sign>
        <EmptyWord> No </EmptyWord>
</Word>
<Word id="5 "> <Name> Smith </Name>                <!-- Le mot dans sa forme originale dans la phrase -->
        <Lemma> Smith </Lemma>                     <!-- lemme, Après lemmatisation -->
        <POS> NNP </POS>                           <!-- L'annotation utilisée par Part Of Speach -->
        <Sign> -- </Sign>
        <EmptyWord> No </EmptyWord>
</Word>
<Word id="6 ">  <Name> I </Name>                   <!-- Le mot dans sa forme originale dans la phrase -->
        <Lemma> I </Lemma>                         <!-- lemme, Après lemmatisation -->
        <POS> PRP </POS>                           <!-- L'annotation utilisée par Part Of Speach -->
        <Sign> -- </Sign>
        <EmptyWord> No </EmptyWord>
</Word>
<Word id="7">  <Name> come </Name>                 <!-- Le mot dans sa forme originale dans la phrase -->
        <Lemma> come</Lemma>                       <!-- lemme, Après lemmatisation -->
        <POS> VBN </POS>                           <!-- L'annotation utilisée par Part Of Speach -->
        <Sign> -- </Sign>
        <EmptyWord> No </EmptyWord>
</Word>
<Word id="8 ">  <Name> from </Name>                 <!-- Le mot dans sa forme originale dans la phrase -->
        <Lemma> from </Lemma>                      <!-- lemme, Après lemmatisation -->
        <POS> IN</POS>                             <!-- L'annotation utilisée par Part Of Speach -->
        <Sign>-- </Sign>                           <!-- L'annotation utilisée pour les termes déclarés comme indicateurs -->
        <EmptyWord> Yes </EmptyWord>               <!—Identification du mot vide -->
</Word>
<Word id="9 ">  <Name> Algeria </Name>             <!-- Le mot dans sa forme originale dans la phrase -->
        <Lemma> Algeria </Lemma>                   <!-- lemme, Après lemmatisation -->
        <POS> NNP </POS>                           <!-- L'annotation utilisée par Part Of Speach -->
        <Sign> Contr </Sign>
        <EmptyWord> No </EmptyWord>
</Word>
</phrase>
```

FIGURE 4.17 – Résultat du prétraitement de la phrase N°2 de l'e-mail.

Les phrases obtenues après la phase de prétraitement de l'e-mail sont les suivantes :

1. Be write e-mail read announce sun newspaper join great company.
2. Be Mr John Smith come Algeria.
3. Be born April 30, 1990.
4. Be graduate MIT June, 2013 have magister degree.
5. Live London have work IBM company three years developer.
6. Be interest Web Intelligence Artificial program.
7. Can be contact 00124587.
8. Thank time consider look speak employ.

4.4.5.2 Scénario N° 01 : Exécution de l'approche sans le module du " Résumé hybride "

Le premier scénario représente le cas où nous traitons toutes les phrases de l'e-mail, et l'information pertinente ; phrase pertinente ; sera déterminée suivant l'existence ou pas

d'unité d'information dans celle-ci. Dans ce cas, nous procédons directement au module de détermination des entités nommées. En appliquant les règles d'exploration contextuelle, nous obtenons des phrases annotées présentées dans un fichier XML, comme illustré dans la figure 4.18 :

FIGURE 4.18 – Extrait du fichier XML résultant de la phase de détermination des entités nommées.

Cette étape nous permet de déterminer et de localiser toute information pouvant être une éventuelle valeur d'un attribut du profil utilisateur. La figure 4.19 présente le corps de l'e-mail après l'étape de la détection des entités nommées :

FIGURE 4.19 – Résultat du module de détection des entités nommées.

Après la détection des EN, le système, grâce au module de Matching, détermine les différentes unités d'informations d'une phrase afin d'attribuer à chaque attribut du profil utilisateur la valeur adéquate.

Dans le tableau 4.1, nous illustrons les dépendances de Stanford entre termes, que nous exploitons dans le but de déterminer les éventuelles valeurs d'attributs du profil utilisateur. Pour cela, nous appliquons l'algorithme d'acquisition des unités d'informations que nous avons proposé, dont le principe consiste en la détermination des différentes dépendances grammaticales (directes ou indirectes) liées aux différents verbes déclencheurs appartenant à chaque phrase.

Phrase	Verbes Déclencheurs	Dépendances au sujet	Dépendance à l'objet	Contexte
1	Join	nsubj (hope, I)	ccomp (announce, hope) prep$_i n(hope, newspaper)$ dobj (join, company)	Aucune EN
2	Am	Advmod (am, I)	nsubj(am, John)	Am, Mr, John Smith
	Come	partmod (I, come)	prep$_f rom(come, Algeria)$	Come, from, Algeria
3	Born	Advmod(born, I)	prep$_o n(born, April)$	Born, on, April
4	Graduated	nsubj (graduated, I)	prep$_f rom(graduated, MIT)$	Graduated, from, MIT, in
			prep$_i n(graduated, June)$	Graduates, from, in, June
	Have	nsubj (have, I)	nn (degree, Magister)	
			dob (have, degree)	Have, Maister, degree
5	Live	nsubj (live, I)	prep$_i n(live, London)$	I, live, in, London
	Work	partmod (I, worked)	prep$_w ith(worked, company)$ nn (company, IBM)	Worked, with, IBM, company
			prep$_f or(worked, years)$ num(years, three)	Worked, with, for, three, years
			prep$_f or(worked, years)$	
			prep$_a s(years, developer)$	Worked, with, as, developer
6	Interested	nsubj (interested, I)	prep$_o n(interested, Web)$	Am, interested, on the, Web, Inteligence
			prep$_o n(interested, Artificial)$ nn (Artificial, Intelligence)	Am, interested, on, intelligence, artificial
			prep$_o n(interested,$ programming)	Am, interested, on, programming
7	Contacted	nsubjpass (contacted, I)	prep$_a t(contacted, 001)$	Be, contacted, at
8	Pas de verbes déclencheurs	Aucun verbe déclencheur	–	–

L'acquisition des UI, nous permet d'avoir un ensemble d'attributs concurrents qui représentent l'ensemble des attributs du profil utilisateur ayant le même type d'information que l'objet de l'UI, et partageant son verbe déclencheur. L'ensemble des verbes déclencheurs et indicateurs sont représentés dans un thésaurus que résume le schéma de la figure 4.20 :

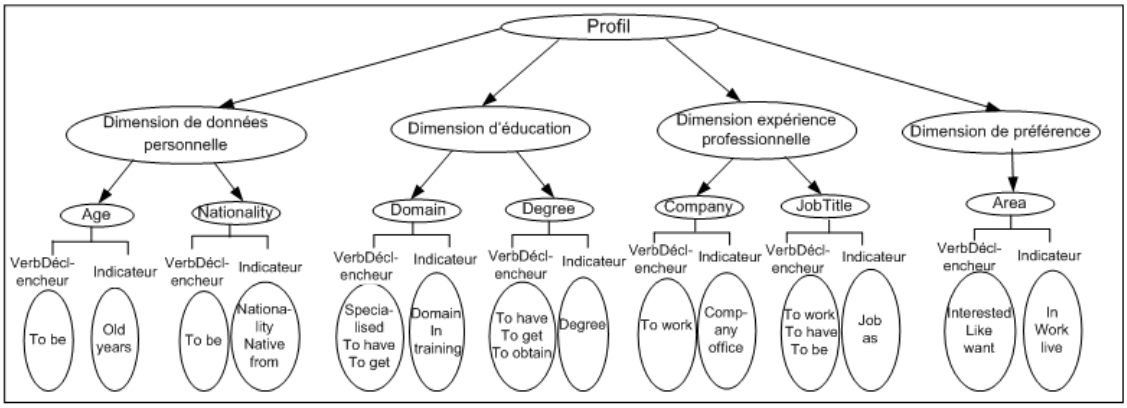


FIGURE 4.20 – Extrait du thésaurus utilisé.

La sélection d'un attribut candidat est assurée par la mesure de similarité entre termes. Pour cela, nous exploitons ***WordNet : : Similarity***, une base de donnée lexicale qui met en relation le contenu lexical et sémantique de la langue anglaise, nous utilisons la mesure de Lin, offerte par cet outil, une mesure qui se base sur le contenu informationnel de chaque terme, dans le tableau 4.2 nous présentons l'ensemble des informations détectées, véhiculées par les phrases de l'e-mail :

ID Phrase	ID Unité d'Information	Attributs concurrents	Similarité à l'UI	Attribut candidat	Attribut gagnant	Bon / Mauvais attribut
1	–	–	–	–	–	–
2	Am, Mr, John Smith	Name	0.16	Yes	Name	Bon
		Nationality	0.11	–		
		Family-Situation	0.14	–		
		JobTitle	0.16	Yes		
		Degree	0.13	–		
		Domain	0.11	–		
		Name-Institution	0.12	–		
	Come, from, Algeria	Residence	–	Yes	Residence	Bon
3	Born, on, April	Birthdate	–	Yes	Birthdate	Bon
4	I, Graduated, from, MIT, in	Name-Institution	0.15	–	Name-Institution	Bon
		Degree	0.12	–		
		Domain	0.10	–		
	I, Graduates, from, in, June	StartDate	0.07	–	EndDate	Bon
		EndDate	0.15	Yes		
	Have, a, Master degree	Degree	0.24	Yes	Degree	Bon
		Family-Situation	0.12	–		

ID Phrase	ID Unité d'Information	Attributs concurrents	Similarité à l'UI	Attribut candidat	Attribut gagnant	Bon Mauvais attribut
5	I, live, in, London	Residence	–	Yes	Residence	Bon
	Worked, with, IBM, company	JobTitle	0.22	–	Company	Bon
		Domain	0.16	–		
		Company	0.23	Yes		
	Worked, with, for, three, years	Duree	–	Yes	Duree	Bon
	Worked, with, as, developer	JobTitle	0.23	Yes	JobTitle	Bon
		Domain	0.12	–		
		Company	0.21	–		
6	Am, interested, on, Web, Intelligence	Domain-Pref	0.21	Yes	Domain-Pref	Bon
		JobTitle	0.18	–		
	Am, interested, on, intelligence artificial	Domain-Pref	0.17	Yes	Domain-Pref	Bon
		JobTitle	0.15	–		
	Am, interested, on, programming	Domain-Pref	0.20	Yes	Domain-Pref	Bon
		JobTitle	0.18	–		
7	I, contacted , at	phone-Number	–	Yes	phone-Number	Bon
8	–	–	–	–	–	–

TABLE 4.2 – Sélection et validation des attributs gagnants.

– : l'attribut n'est pas pris en considération.

Dans le tableau 4.3, nous illustrons les résultats de notre approche sur quelques phrases tirées à partir des e-mails électroniques concernant le recrutement dans le domaine informatique :

1. I am Bill.
2. I live in London city.
3. I have got a diploma from MIT.
4. I have done a training in MIT

5. I have studied from January 1, 2011 to June 19, 2013

ID Phrase	ID Unité d'Information	Attributs concurrents	Mesure Similarité	Attribut gagnant	Bon Mauvais attribut
1	I, am, Bill	Name	0.2491	JobTitle	Mauvais
		Nationality	0.1687		
		Family-Situation	0.1655		
		JobTitle	0.2545		
2	I ,live, in, London , city	Residence	–	Residence	Bon
3	Have, get, a, from, MIT	Name-Institution	0.1111	Degree	Mauvais
		Domain	0.1012		
		Degree	0.1385		
4	Have, done, a, in, MIT	Institution	0.2045	Domain	Mauvais
		Domain	0.2328		
		Company	0.1489		
5	Have, studied, from, January,to	StartDate	0.1142	StartDate	Bon
		EndDate	0.1091		
		Durée	0.1012		

TABLE 4.3 – Autre exemple de phrase d'e-mail.

Les résultats obtenus dans le tableau 4.3, nous permet de discuter et de répondre à quelques cas particulier, rencontrés par notre système, que nous détaillons comme suit :

• ***Cas N°01 : Est-ce possible d'avoir un attribut qui appartient à deux types d'informations distincts ?***

Prenons l'exemple : " *I am 25* ", dans ce cas, *25* est considéré comme étant une information `numérique`. Alors que dans : I am 25 years, nous détectons une information de type `période`, *25 years*. Les deux phrases doivent contribuer à fournit la valeur du même attribut `âge`. Afin de remédier à ce problème, nous confirmons la possibilité qu'un attribut puisse appartenir à deux types d'information. Ici, l'attribut âge appartient aux types numérique et temporelle.

- ***Cas N°02 :***

Nous constatons à partir des résultats obtenus dans le tableau précédant, que l'exactitude de ces résultats dépend de la mesure de similarité, entre le contexte de l'unité d'information détectée et l'ensemble des termes associés à chaque attribut dans notre ontologie de domaine.

Prenons l'exemple de StartDate et EndDate, deux attributs dans l'unité n°2 de la phrase 4, et les attributs concurrents de la phrase 4 du tableau 4.3, les résultats sont dus à l'ensemble des verbes déclencheurs et indicateurs partagés entre les deux attributs.

✍ ***Solution proposée :***

Nous proposons comme solution de comparer les deux dates et d'attribuer respectivement la date minimale et maximale aux attributs StartDate et EndDate.

- ***Cas N°03 :***

Les cas échéants de notre approche peuvent être dû à l'utilisation d'une partie de l'ontologie de domaine, et au fait de ne pas prendre en considération tous les termes pouvant être associés à un attribut. Prenons l'exemple de l'unité 3 du tableau 4.3.

Nous avons testé les résultats de la mesure de similarité dans les deux cas où la liste des verbes déclencheurs et indicateurs de chaque attribut est enrichît ou pas :

◦ *Institution :* graduate, get, study, diploma. Dans ce cas la mesure de similarité est de 0.1111.

◦ *Domain :* have, get, prepare, training, software. Dans ce cas la mesure de similarité est de 0.1012.

◦ *Degree :* have, degree, prepare, Licence , Master. Dans ce cas la mesure de similarité est de 0.1385.

Ces résultats sont non signifiants pour témoigner de la non validité de notre approche, ce que nous avons prouvé avec l'ajout des indicateurs : university et institution à l'attribut ***Institution*** : Sim (Institution) = 0.15, Sim (Domain) = 0.10, Sim (Degree) = 0.13.

D'où l'enrichissement de l'ontologie. Ou bien avoir une ontologie de domaine complète, nous permet d'avoir des résultats encore plus exacts et plus précis.

- ***Cas N°04 :***

Une information peut être exprimée de différentes manières, nous y trouvons par exemple la forme : " *I was born in Algeria, I had my diploma from the university of Abdrahmane Mira. Now, i am working there as a teacher* ". Les informations dispatchées sur plusieurs phrases peuvent être une source de lacune.

✍ ***Solution proposée :***

Nous envisageons d'améliorer l'algorithme d'acquisition des unités d'informations, dans nos perspectives à venir.

- ***Cas N°05 :***

"*I am Bill . . . I am Algeria . . . I am a Master degree . . .*". Nous constatons dans ces exemples, les ambiguïtés sémantiques entre les informations d'un même type, tel illustré dans les exemples précédant où l'attribut Name subit une erreur d'attribution de valeur.

✍ ***Solution proposée :***

Nous proposons l'utilisation d'un Thésaurus, qui sera sous forme de liste de valeurs d'attributs (degree, domain, residance, company, nationality, jobTitle).

4.4.5.3 Scénario N° 02 : Exécution de l'approche avec le module du " Résumé hybride "

Le deuxième scénario représente le cas où nous appliquons le résumé sur le corps de l'e-mail :

- **Cas N°01 : Application du résumé " A " :**
  - **Calcule des métriques statistiques :**

– Similarité au titre (Title-Similarity).
– Position de la phrase dans le texte (Phrase-Position).
– Somme des fréquences des mots d'une phrase avec la mesure TFi (SumWord-Freq).
– Somme des poids des mots d'une phrase avec la mesure TFi*IDFi (SumWord-Weight).
– Somme des valeurs statistiques (SumStatic-Metric).
– Poids de pertinence d'une phrase avec la validation sémantique (Sentence-Weight-semantic).

En appliquant les différentes métriques statistiques sur les phrases obtenues après la phase de pré-traitement, nous obtenons le poids de chaque phrase comme présenté dans l'extrait du fichier XML illustré dans la figure 4.21.

```
<!-- Cette partie sera remplis lors du module du prétraitement du résumé A-->

<OriginalPhrase>  I am Mr. John Smith, I come from Algeria  </OriginalPhrase>

  <WordNumber> 9 </WordNumber>                          <!-- Le nombre de mots dans la phrase -->
  <TitleSimilarity> 0 </TitleSimilarity>                <!-- Valeur de la métrique sur la similarité de la phrase avec le titre-->
  <PhrasePosition> 1/2 </PhrasePosition>                <!-- Valeur de la métrique sur la position de la phrase -->
  <SumWordFrequencies> 1 </SumWordFrequencies>          <!-- Valeur de la métrique sur la fréquence des mots dans la phrase-->
  <SumWordWeight> 0.62</SumWordWeight>                  <!-- Valeur de la métrique sur la somme des poids des mots dans la phrase -->
  <SumStatistiqueMetric>2.12</SumStatistiqueMetric>     <!-- Somme des valeurs des métriques statistiques -->
  <SenteceWeight> 4.12 </SenteceWeight>                 <!-- Poids de pertinence d'une phrase -->
```

FIGURE 4.21 – Représentation des métriques statistiques liées à la phrase N°2

Après avoir effectué le même traitement sur toutes les phrases de l'e-mail, nous aurons les résultats représentés dans le tableau 4.4 :

ID-phrase	Title-Similarity	Phrase-Position	SumWord-Freq	SumWord-Weight	SumStatic-Metric	Sentence-Weight-semantic	Order
1	0	0.25	1	0.64	1.89	2.89	7
2	0	0.5	1	0.62	2.12	4.12	4
3	0	0.75	1	0.40	2.15	3.15	6
4	0	1	1	0.60	2.60	5.60	2
5	0	0.75	1	0.69	2.44	5.94	1
6	0	0.5	1	0.62	2.12	5.12	3
7	0	0.25	1	0.51	1.76	3.26	5
8	0	0	1	0.84	7.84	2.34	8

TABLE 4.4 – Résultats des mesures calculés pour résumé "A".

En combinant les valeurs des métriques statistiques, nous obtenons le poids de chaque phrase. Par suite, Nous validons les calculs obtenus pour chaque phrase par l'application de l'algorithme de décision, qui intègre la technologie sémantique en utilisant l'ontologie de notre domaine.

Pour chaque mot i de la phrase j, nous vérifions l'appartenance du mot à l'ontologie et cela en se basant sur les résultats de la fonction PROTEGE. Deux cas sont présentés :

- Si le terme i fait parti d'une des classes de l'ontologie (PROTEGE (termei) = Exact), alors ajouter 1/2 pour le poids statistique de la phrase.
- Sinon ;(PROTEGE (termei) = Fail) ; garder la valeur obtenue avec les mesures statistiques.

A partir des poids obtenus par l'intégration de la sémantique de l'ontologie, définie dans le tableau précédent, nous déterminons l'ordre du classement des phrases : phrase 5, phrase 4, phrase 6, phrase 2, phrase 7, phrase 3, phrase 1, phrase 8.

Ainsi les phrases à intégrer dans le résumé (basé sur la méthode A) sont celles qui appartiennent à 75% de la taille de l'e-mail traité. Donc pour les 8 phrases, nous assimilons que les 6 premières phrase (classées par ordre décroissant par rapport au poids), d'où le résumé " A " sera composé par les phrases : 2, 3, 4, 5, 6, 7.

Le résultat de cette méthode répond parfaitement à notre besoin, il élimine que la première et la dernière phrase qui ne sont que des phrases de politesse. L'élimination de ces deux phrases nous permet la réduction du temps et du traitement pour les modules prochain.

- **Cas N°02 : Application du résumé " B " :**

Dans ce cas, nous présentons le résumé de la méthode " B " qui se base sur les graphes. Cette méthode est basée sur l'algorithme qui modélise un document texte comme un graphe où l'on déduit la pondération des phrases.

Le corps de l'e-mail contient 8 phrases. Après les processus de pré-traitement et de vectorisation, nous obtenons une matrice S de $\boldsymbol{P = 8}$ phrases et de $\boldsymbol{N = 43}$ termes, tel que les lignes représentent les phrases, et les colonnes représentent les mots.

$$\begin{pmatrix} be & work & have & company & ... \\ 1 & 0 & 0 & 1 & \\ 1 & 0 & 0 & 0 & \\ 1 & 0 & 0 & 0 & \\ 1 & 0 & 1 & 0 & \\ 0 & 1 & 1 & 1 & \\ 1 & 0 & 0 & 0 & \\ 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & \end{pmatrix}$$

La matrice d'adjacence A de P*P phrases est représentée comme suit :

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Tel que :

- Chaque ligne représente la phrase d'arrivée i.
- Chaque colonne représente la phrase de départ j.
- Il existe une arête entre phrase i et une phrase j, si elles ont au moins un mot en commun.

Nous monterons le fonctionnement de l'algorithme sur le graphe correspondant, comme montré dans la figure 4.22 :

1. Choisir le sommet 1 dont le poids est 10, T= 1.
2. Parmi les sommets voisins, qui ne sont pas dans T, on choisit le sommet 5 avec un poids de 9.
3. Le plus lourd voisin de 5 qui n'est pas dans T, est le sommet 4 dont le poids est 7.

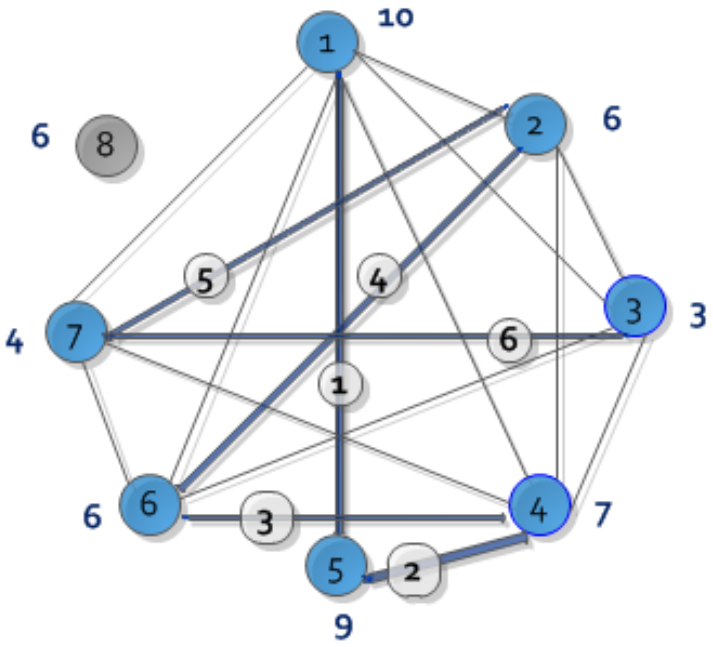


FIGURE 4.22 – Graphe pour le corps de l'email.

Les arcs les plus gros, montrent le chemin suivi par l'algorithme pour visiter les phrases qui formeront le résumé. La suite des sommets obtenus par l'algorithme glouton utilisé dans REG sont : 1, 5, 4, 6, 2, 7, 3 qui seront représenter dans le résumé basé sur la méthode " B ".

Comme résultat de cette méthode, est qu'elle représente dans le résumé la phrase 1, alors qu'elle n'est pas pertinente (Porte pas d'entité nommée).

- **Cas N°03 : Application du résumé hybride :**
  - Résumé A = 2, 3, 4, 5, 6, 7.
  - Résumé B = 1, 2, 3, 4, 5, 6, 7.
  - Résumé_hybride = 1, 2, 3,4, 5, 6, 7.

4.4.5.4 Comparaison des deux scénarios d'exécution

La différence entre les deux scénarios est l'intégration du module du résumé hybride. Nous constatons que la méthode du résumé que nous avons proposé, nous permet effectivement un résumé orienté domaine grâce à l'utilisation des ontologies de domaine. Alors que le résumé B en utilisant des métriques purement statistiques a inclus la phrase 1, ayant un poids élevé, alors qu'elle ne véhicule aucune information pertinente pour le profil utilisateur.

Le principe du résumé hybride est d'éliminer toute phrase jugée non pertinente par les deux résumés, ce qui nous permettra d'ignorer ces phrases lors de l'exécution des différents modules de l'approche sans avoir à perdre de l'information.

4.4.5.5 Métriques d'évaluation du système proposé

L'exécution de notre système sur un ensemble d'email, nous a permis de définir un ensemble de métriques d'évaluation :

✓ La relation entre le nombre d'unités d'un email et la précision du système.

✓ **Précision** = Nombre d'attributs bien détecté (dans un email) / Nombre d'attribut détectés (dans un email)

✓ **Taux d'erreur** = 1- Précision.

Dans le tableau 4.5, nous présentons les résultats retournés par notre système pour un échantillon de cinq e-mail :

	Nombre Phrase	Nombre Unité	cas1	cas2	cas3	cas4	cas5	Précision	Taux erreur
Email 1	8	14	–	–	–	–	–	100%	0%
Email 2	4	5	–	Yes	Yes	–	Yes	40%	60%
Email 3	4	6	–	–	–	–	Yes	50%	50%
Email 4	4	6	–	Yes	–	–	Yes	66%	34%
Email 5	3	7	–	Yes	–	–	Yes	71%	29%
					Précision	du	Système	64.4%	

TABLE 4.5 – Evaluation du Système proposé.

La relation entre le nombre d'unités d'un email et la précision du système. La figure 4.23, représente une relation proportionnelle entre la précision du système et le nombre d'unité détecté dans l'email.

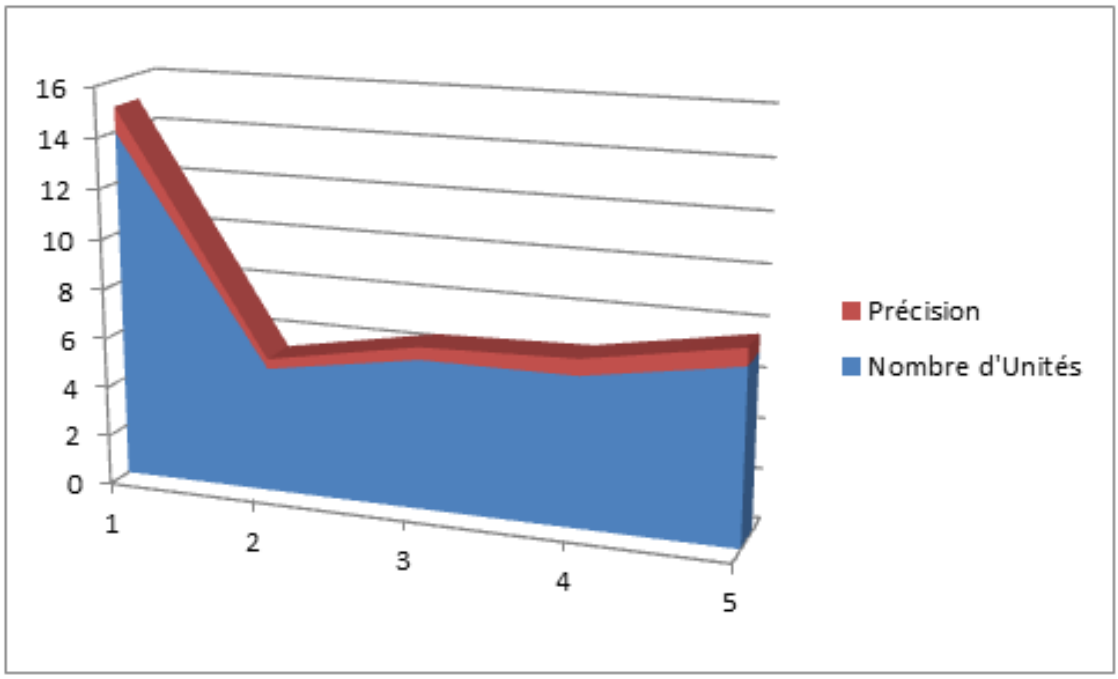


FIGURE 4.23 – La relation entre la précision du système et le nombre d'unité de l'email.

4.5 Conclusion

Durant ce chapitre nous avons présenté une nouvelle approche pour la construction d'un profil utilisateur à partir d'un document textuel, dans notre cas, nous avons pris l'exemple des demandes d'emploi dans le domaine informatique.

Nous avons commencé par présenter l'architecture globale de notre système, pour ensuite détailler le processus de ces différents modules. Pour l'évaluation de l'approche proposée, nous avons défini quelques mesures statistiques dans le but de déterminer la performance et la précision de celle-ci.

Conclusion générale et perspectives

DURANT ces dernières années, nous avons constaté la nécessité de l'intégration de l'aspect sémantique dans le traitement automatique de l'information via le Web. Le recrutement électronique est l'un des domaines les plus touchés par ce phénomène.

Dans le cadre du traitement et de la gestion des demandes d'emplois, nous avons proposé un système de construction de profil utilisateur, en utilisant les techniques d'extraction d'informations à partir des e-mails électroniques en se basant sur l'ontologie du domaine de recrutement en informatique.

L'architecture globale de l'approche proposée comporte plusieurs modules : Un module de pré-traitement permettant d'extraire l'émetteur et le récepteur à partir du l'entête de l'e-mail avec le sujet de la demande d'emploi. Nous avons également effectué des traitements sur le corps de l'e-mail pour avoir finalement les différentes phrases qui le compose. Un module du résumé hybride où nous avons proposé une méthode de résumé en exploitant l'aspect formel offert par les statistiques ainsi que la sémantique de l'ontologie du domaine. Afin de minimiser la perte d'information, nous avons hybridé notre proposition avec une autre méthode du résumé déjà existante. Un module de détermination des entités nommées qui a pour but d'identifier et classer les éléments du texte représentant les différents attributs du modèle du profil utilisateur en exploitant des règles d'exploration contextuelle. Un module de correspondance avec les informations du profil. L'approche se termine par une mise à jour entre le profil courant et celui obtenu précédemment.

Pour une meilleure manipulation et représentation de l'information textuelle ainsi que l'organisation des différentes étapes de traitement, nous avons utilisé tout au long de notre approche la structure XML.

Suite aux différents scénarios d'exécution et testes appliqués à un ensemble d'e-mail électroniques, nous envisageons plusieurs perspectives :

- Le traitement les documents envoyés comme pièce-jointe.

• La construction d'un profil utilisateur à partir d'un e-mail multi-domaine.

• Le traitement des expressions temporelles, et la détection de la chronologie des événements, préférences et centres d'intérêt, exemple : *Last years, i worked as a designer with IBM company, now i am a developer.*

• L'amélioration de l'algorithme d'acquisition des Unités d'Informations, afin de prendre en charge les phrases complexes (un verbe déclencheur avec plusieurs objets associés), et l'utilisation d'une segmentation sémantique autre que la segmentation en phrase.

• L'intégration d'un analyseur lexicale basé sur la sémantique.

• L'étude du cas de détermination des dépendances grammaticales par transitivité (indirectement).

• Comme perspective, nous envisageons également le traitement automatique et la gestion des différentes catégories de profils utilisateurs obtenues par notre système, dans le but d'attribuer à chaque tache le profil utilisateur le plus adéquat.

Bibliographie

[1] A.Anderson, M.Corney, and G.Mohay. Mining e-mail content for author identification forensics. *ACM SIGMOD Record*, 30(4) :55–64, 2001.

[2] A.Bossard. Contribution au résumé automatique multi-documents. *Thèse de Doctorat en informatique présentée à l'Université Paris 13 ,Paris-Nord*, 2010.

[3] A.Bouhafs. Utilisation de la méthode d'exploration contextuelle pour une extraction d'informations sur le web dédiée à la veille,réalisation du système informatique java veille. *Thèse de Doctorat en Mathématiques, Informatique Appliquées aux Sciences Humaines présentée à l'Université Paris IV Sorbonne*, 2005.

[4] A.Budanitsky and G.Hirdt. Evaluating wordnet-based measures of lexical semantic relatedness. *Journal of Computational Linguistics*, 32(1) :13–47, 2006.

[5] A.Claude, F.Chantelot, and V.Zoonekynd. Interaction entre gènes : Extraction d'information d'un corpus de résumés d'articles de recherche. Mars 2004.

[6] A.Csomai and R.Mihalcea. Linking documents to encyclopedic knowledge. *IEEE Intelligent Systems*, 23 :34–41, 2008.

[7] A.Labadie. Segmentation thématique de texte linéaire et non-supervisée : Détection active et passive des frontières thématiques en français. *Thèse de Doctorat présentée à l'université des sciences et Techniques du Languedoc, Montpelier*, 2009.

[8] A.Maedche and S.Staab. Mining ontologies from text. *In Proceedings of the* 12^{th} *European Workshop on Knowledge Acquisition, Germany*, pages 189–202, 2000.

[9] A.Mokrane. Représentation de collections de documents textuels : application à la caractérisation thématique données du web et leurs sources. *Thèse de Doctorat en Informatique présentée à l'Université Montpillier II*, Novembre 2006.

[10] A.Stefani and C.Strapparava. Personalizing access to web sites : The siteif project. *In Proceedings of the* 2^{nd} *Workshop on Adaptive Hypertext and Hypermedia, USA*, pages 69–75, June 1998.

[11] A.Wong, C.Yang, and G.Salton. A vector space model for automatic indexing. *Communications of the ACM*, 18 :613–620, 1975.

[12] A.Woods. Conceptual indexing : A better way to organize knowledge. *Technical Report SMLI TR-97-61, Sun Microsystems Laboratories, Mountain View, CA*, April 1997.

[13] B.Tan, X.Shen, and C.Zhai. Mining long-term search history to improve search accuracy. *In Proceedings of the* 12^{th} *international conference on Knowledge Discovery and Data Mining, University of Illinois at Urbana-Champaign*, pages 718–723, August 2006.

[14] C.Brewster and Y.Wilks. Dialogue-based user models. *In Proceedings of IEEE*, 74(7) :948–960, 1986.

[15] C.Chang, M.Kayed, M.R.Girgis, and K.Shaalan. A survey of web information extraction systems. *In proceedings of IEEE Transactions on knowledge and data engineering, TKDE*, August 2006.

[16] C.Rijsbergen. Information retrieval, in information retrieval experiments. *Butterworths, London,*2^{nd} *edition*, 1979.

[17] D.Billsus and M.Pazzani. A hybrid user model for news stories classification. *In Proceedings of the* 7^{th} *International Conference on User Modelins*, pages 99–108, 1999.

[18] D.Kostadinov. Personnalisation de l'information : une approche de gestion de profils et de reformulation de requêtes. *Thèse de Doctorat en informatique présentée à l'Université de Versailles Saint-Quentinen- Yvelines, France*, Septembre 2008.

[19] D.Kostadinov and M.Bouzeghoub. Personnalisation de l'information : Aperçu de l'état de l'art et définition d'un modèle flexible de définition de profils. *In Actes de la seconde édition de la Conférence en Recherche d'Information et Applications, Coria*, pages 201–218, 2005.

[20] D.Shen and Z.Chen. Web-page classification through summarization. *In Proceedings of the* 27^{th} *Annual International ACMSIGIR Conference on Research and Development in Information Retrieval, Sheffield*, pages 242–249, 2004.

[21] D.Widyantoro and J.Yin. A swift messenger in cyberspace. *In Proceedings of Spring Symposium Workshop on Intelligent Agents in Cyberspace, Stanford*, pages 62–67, Mars 1999.

[22] H. P. Edmundson. New methods in automatic extracting. *In proceedings of the ACM*, 16(2) :264–285, 1969.

[23] F.Boubekeur. Contribution à la définition de modèles de recherche d'information flexibles basés sur les cp-nets. *Thèse de Doctorat en informatique présentée à l'Université Paul Sabatier*, 2008.

[24] F.Boudin. Exploration d'approches statistiques pour le résumé automatique de texte. *Thèse de Doctorat en informatique présentée à l'Université d'Avignon et des Pays de Vaucluse*, 2008.

[25] F.Boudin and J.M.Torres. Neo-crtex : a performant user-oriented multi-document summarization system. *In Proceedings of Spring LNCS, Mexico DF*, pages 551–562, 2007.

[26] F.Boudin and J.M.Torres. Résumé automatique multi-document et indépendance de la langue : une première évaluation en français. *In proceedings of TALN'09 Senlis, France*, 2009.

[27] F.Harrathi. Extraction de concepts et de relations entre concepts à partir des documents multilingues : Approche statistique et ontologique. *Thèse de Doctorat en InfoMaths : Distribution et Recherche d'Information Multimedia, présentée à L'Institut Nationale des Sciences Appliquées de Lyon*, Septembre 2009.

[28] F.Iqbal and H.Binsalleeh. Mining write prints from anonymous e-mails for forensic investigation. *Faculty of Engineering and Computer Science, Concordia University, Montreal*, 2010.

[29] G.Amato and U.Straccia. User profile modelling and applications to digital librairies. *In Proceedings of the* 3^{rd} *European Conference on Research and advanced technology for digital libraries*, pages 184–187, 1999.

[30] P.E. Genest and G.Lapalme. Framework for abstractive summarization using text-to-text generation. *Proceedings of the* 49^{th} *Annual Meeting of the Association for Computational Linguistics, Portland, Oregon*, 32(1) :64–73, 24 June 2011.

[31] G.Erkan and D.RRadev. Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*, page 39, 2004.

[32] G.Gentili, A.Micarelli, and F.Sciarrone. Infoweb : An adaptive information filtering system for the cultural heritage domain. *Applied Artificial Intelligence (AAI)*, 17(8-9) :715–744, 2003.

[33] G.Mourad. Présentation de connaissances linguistiques pour le repérage et l'extraction de citations. *Thèse de Doctorat présentée à l'université de Paris-Sorbonne, Paris IV*, 2001.

[34] G.Salton and C.Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 29 :351–372, 1973.

[35] G.Zipf. Human behaviour and the principal of least effort. *Addison Wesley*, 1949.

[36] G.Van Heijst and A.Schreiber ANS B.Wielinga. Using explicit ontologies in kbs development. *Int.J. of Human-Computer Studies*, 46(2/3) :183–292, 1997.

[37] H.Lieberman. Autonomous interface agents. *In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, New York USA*, pages 67–74, April 1997.

[38] H.Sakagami and T.Kamba. Learning personal preferences on online newspaper articles from user behaviors. *In Proceedings of the* 6^{th} *International WWW Conference, Santa Clara, California*, pages 7–11, April 1997.

[39] I.Fidelia. Fouille de texte, méthodes, outils et applications. *Lavoisier et Hermes science à Paris*, pages 200–201, Mars 2007.

[40] I.Mani, J.Pustejovsky, and R.Gaizauskas. The language of time. *A Reader*, 2005.

[41] I.Mani and M.T.Maybury. Advances in automatic text summarization. *The MIT Press*, 1999.

[42] J.Budzik and K.Hammond. Users interactions with everyday applications as context for just in time information access. *In Proceedings of the* 5^{th} *international conference on intelligent user interfaces, ACM New York, NY, USA*, 2000.

[43] J.Gowan. A multiple model approach to personalised information access. *Mémoire de Master en informatique présenté à l'Université de College Dublin*, February 2003.

[44] J.Li, R.Zheng, and H.Chen. From fingerprint to writeprint. *Communications of the ACM -Supporting exploratory search*, 49 :76–82, 2006.

[45] J.Min, J.Y.Ahn, and C.W.Chung. Efficient extraction of schemas for xml documents. *Division of Computer Science, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology*, 2002.

[46] J.Minel. Filtrage sémantique : du résumé automatique à la fouille de textes. *Hermès, Université Paris-Sorbonne*, 2002.

[47] J.M.Torres-Moreno and J.R.Rodrigez. Reg : Un algorithme gloutton appliqué pour le résumé automatique. *In Prceedings of* 10^{th} *International Conference on Statical Analysis of Textual Data, Montréal Canada*, 2010.

[48] J.M.Torres-Moreno, P.Velázquez-Morales, and J.G.Meunier. Cortex : un algorithme pour la condensation automatique de textes. *Dans les actes de Colloque Interdisciplinaire en Sciences Cognitives (ARCo)*, 2 :65–75, 2001.

[49] J.M.Torres-Moreno, P.Velázquez-Morales, and J.G.Meunier. Condensés de textes par des méthodes numériques. *Dans les actes de Journées internationales d'Analyse statistique des Données Textuelles (JADT)*, 2 :723–734, 2002.

[50] J.Séguéla and G.Saporta. e-recrutement : recherche de mots-clés pertinents dans le titre des annonces d'emploi. 10^{eme} *Journées internationales d'Analyse statistique des Données Textuelles,Paris : France*, pages 4–7, 2010.

[51] J.Torres-Moreno. Résumé automatique de documents : une approche statistique. *Hermes science*, 2011.

[52] K.Abbas. Système d'accès personnalisés à l'information : application au domaine médical. *Thèse de Doctorat en informatique présenté à l'Institut National des sciences Appliquées de Lyon, France*, Decembre 2008.

[53] K.Ottens. Un système multi-agent adaptatif pour la construction d'ontologies à partir de textes. *Thèse de Doctorat en Informatique et Télécommunication présentée à l'Université Paul Sabatier de Toulouse III*, Octobre 2007.

[54] L.Chen and K.Sycara. Webmate : A personal agent for browsing and searching. *In Proceedings of the* 2^{nd} *international conference on autonomous agents and multi agent systems, USA*, pages 10–13, 1998.

[55] L.Tamine and M.Boughanem. Accès personnalisés à l'information : Approches et techniques. *Thèse de Doctorat en informatique présentée à l'Institut de recherche en informatique de Toulouse*, Janvier 2005.

[56] L.Tamine-Lechani, N.Zemirli, and W.Bahsoun. Approche statistique pour la définition du profil d'un utilisateur de système de recherche d'information. *Information - Interaction - Intelligence, Cépaduès Editions*, 7 :5–25, 2007.

[57] S.El manar el bouanani and I.Kassou. Vers une méthodologie de modélisation d'une signature unique des profils web : Module de détection des auteurs des forums web. *Thèse de Doctorat présenté à l'Université Mohammed V Souissi - Rabat*, 2012.

[58] M.Baziz. Indexation conceptuelle/sémantique guidée par ontologie pour la recherche d'information. *Thèse de Doctorat en informatique présentée à l'Institut de Recherche en Informatique de Toulouse (IRIT)*, 2005.

[59] M.Daoud. Accès personnalisé à l'information : approche basée sur l'utilisation d'un profil utilisateur sémantique dérivé d'une ontologie de domaines à travers l'historique des sessions de recherche. *Thèse de Doctorat en informatique présentée à l'Université Paul Sabatier, France*, Decembre 2009.

[60] M.Daoud and L.Tamine-Lechani. Construction des profils utilisateurs à base d'ontologie pour une recherche d'information personnalisée. *Journal of Laboratoire IRIT Université Paul Sabatier*, 2008.

[61] M.Morneau. Recherche d'information sémantique et extraction automatique d'ontologie du domaine. *Mémoire de Maîtrise en informatique de maître des sciences présenté à la Faculté des études supérieures de l'Université Laval*, 2006.

[62] M.Nicoletti, S.Schiaffino, and D.Godoy. Mining interests for user profiling in electronic conversations. *Journal of Isistan Research Institute, Conicet-Unicien, Campus Universitario, Paraje Arroyo Seco*, 2012.

[63] M.Pazienza. Information extraction : a multidisciplinary approch to an emerging information technology. *In proceedings of International Summer School, SCIE-97 Frascati, Italy*, July 1997.

[64] N.Aussenac-Gilles, B.Biébow, and N.Szulman. Revisiting ontology design : a method based on corpus analysis. *Lecture Notes in Artificial Intelligence*, pages 172–188, 2000.

[65] N.Belkin and W.Croft. nformation filtering and information retrieval : Two sides of the same coin. *Communication of the ACM*, 35(12) :29–38, 1992.

[66] N.Guarino. Understanding, building, and using ontologies : A commentary to using explicit ontologies in kbs development. *International Journal of Human and Computer Studies*, 46 :293–310, 1997.

[67] N.Guarino and P.Giaretti. Ontologies and knowledge bases : Towards a terminological clarification. *In proceedings Towards Very Large Knowledge Bases, NJI*, pages 25–32, Mars 1995.

[68] N.Omrane, A.Nazarenko, and S.Szulman. Le poids des entités nommées dans le filtrage des termes d'un domaine. *In proceedings of* 9^{eme} *internationale de Terminologie et Intelligence Artificielle, Paris : France*, 2011.

[69] N.Zemirli. Modèle d'accès personnalisé à l'information basé sur les diagrammes d'influence intégrant un profil utilisateur évolutif. *Thèse de Doctorat en informatique présentée à l'Université Paul Sabatier de Toulouse III*, Juin 2008.

[70] O.Ressad-Bouidghaghen. User profiling with case-based reasoning and bayesian networks. *Thèse de Doctorat présenté à l'université de Toulouse*, 2011.

[71] P.Luhn. The automaic creation of litterature abstracts. *In proceding of Research and Development*, 2(2) :159–165, April 1958.

[72] P.Sheridan and A.Smeaton. The application of morpho-syntactic language processing to effective phrase matching. *Information Processing and Managment*, 28(3) :349–370, 1992.

[73] R.Brandow, K.Mitze, and L.F.Rau. Automatic condensation of electronic publications by sentence selection. *In Processing Information and Management : an International Journal*, 31 :675–685, 1995.

[74] R.Gaizauskas and Y.Wilks. Information extraction : Beyond document retrieval. *Computational Linguistics and Chinese Language Processing*, 3 :17–60, August 1998.

[75] R.Loth. Les offres d'emploi comme texte ? annoter et étudier un corpus pour un projet d'extraction d'information. *Proceedings " Les Cahiers de l'ED ", Université Paris Ouest Nanterre La Défense Nanterre*, septembre 2010.

[76] R.Mihalcea and D.Moldovan. Semantic indexing using wordnet senses. *In Proceedings of ACL Workshop on IR AND NLP, Hong Kong*, Octobre 2000.

[77] R.Milhacea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. *In ACL'04 on Interactive poster and demonstration sessions, Association for Computational Linguistiscs, Morristown, USA*, pages 181–184, 2004.

[78] R.Neches, R.E.Fikes, and T.Finin. Enabling technology for knowledge sharing. *AI Magazine*, pages 16–36, 1991.

[79] S.Brin and L.Page. The anatomy of a large-scale hypertextual web search engine. *Network and ISDN System, Elsevier Science Pub B.V*, pages 107–117, 1998.

[80] S.Nicolas. un filtre sémantique pour les moteurs recherche conventionnels par comparaison de structures de connaissance extraites depuis des textes en langage naturel. *Thèse de Master en informatique et génie logiciel présentée à l'Université Laval*, pages 1–72, 2003.

[81] S.Robertson and K.Sparck. Relevance weighting for search terms. *Journal of The American Society for Information Science*, 27(3) :129–146, 1976.

[82] S.Schiaffino and A.Amandi. User profiling with case-based reasoning and bayesian networks. *In Proceedings Open Discussion Track – International Joint Conference, Brazil*, pages 12–21, 2000.

[83] T.Gruber. A translation approach to portable ontology specifications. *Journal of Knowledge Systems Laboratory, USA*, 14 :199–220, April 1993.

[84] T.Kaczmarek and D.Zyskowski. Information extraction from web pages for the needs of expert finding. *Studies in logic, grammar and rhetoric*, 22 :141–157, 2010.

[85] T.Poibeau. Extraction automatique d'information : du texte brut au web sémantique. *Publisher, Hermes Science Publications*, 2003.

[86] V.Pillet. Méthodologie d'extraction automatique d'information à partir de la littérature scientifique en vue d'alimenter un nouveau système d'information. *Thèse de Doctorat présentée à l'Université de droit, d'économie et des sciences ,Marseille*, 2000.

[87] V.Raghavan and H.Sever. On the reuse of past optimal queries. *In Proceedings of the* 18^{th} *annual international ACM SIGIR conference on Research and development in information retrieval, NY, USA*, pages 344–350, 1995.

[88] Wahlster and A.Kobsa. Natural language processing as a foundation of the semantic web. *Delft, Netherlands : now Publishers Inc*, 2009.

[89] Y.Sklab. Spécification d'un modèle formel pour le contexte utilisateur. *Mémoire de Magister en informatique : Réseaux et systèmes distribués présenté à l'université Abd-Rahmane-Mira de Béjaia*, 2013.

[90] Y.Wilks and M.Stevenson. Sense tagging : Semantic tagging with a lexicon. *In proceedings of the SEGLEX Workshop, University of Sheffield*, pages 47–51, 1997.

[91] Z.Boucetta. Appariemeny sémantique des cvs/offres d'emploi dans le cadre du e-recrutement. *Mémoire de Magister en informatique présenté à l'université Mentouri de Constantine*, 2008.

Annexe A

Annexe

A.1 Les différents modules de l'approche proposée

A.1.1 Module du prétraitement

✍ **Segmentation du texte en phrases :**

Algorithm 8 Segmentation-phrase

Entrée: *C :* Corps de l'e-mail.

Sortie: *LP :* Liste des phrases.

Début

1: Remplacer plusieurs sauts de ligne " n̊ " par un seul.

2: Remplacer plusieurs espaces ou plusieurs tabulations par un seul espace.

3: Parcourir le texte afin de déterminer les différents termes, et ce par rapport à la rencontre d'un blanc ou d'un caractère spéciale.

4: Supprimer ce qui est entre parenthèses, qui représente généralement une information redondante. //* Nous supposons que nos segments textuels unitaires (phrase) sont délimités par les signes de ponctuation suivants : " ? ", " ! ", " . ", " ... " que nous avons appelé les délimiteurs pivot.

5: L'application des règles de segmentation.

Fin

Les règles de segmentation sont formulées comme suit :

Algorithm 9 Regle-segmentation

Entrée: *DP* : liste des délimiteurs de phrases : !. ?

Sortie: *LP* : Liste des phrases.

Début

1: **Soit** un Délimiteur Pivot X
2: **si** le contexte gauche de X est G ET le contexte droit de X est D **alors**
3: prendre la décision Y (fin ou non-fin de phrase)
4: **fin si**

Fin

La prise de décision Y revient à évaluer le contexte gauche, G, et le contexte droit, D, par rapport au délimiteur pivot, en prenant en compte les différents cas particuliers et ainsi l'élimination de l'ambiguïté, exemple du point, qui ne signifie pas toujours une fin de phrase, dans les adresses électroniques " adres@live.fr ", les sigles " I.B.M ", les entités numériques, le cas de " Mr.Jobs " ainsi que le cas des trois points de suspension " ... " qui est une succession de points.

✍ **Normalisation des dates et des numéros :**

Nous supposons que les dates qui peuvent être rencontrées dans le texte seront sous les formes suivantes : " 12/6/2013 ", "12.6.2013 ", " 12th JUNE " , "12 JUNE 2013 ", "12 JUNE ", " Monday, JUNE 12th 2013 ", " JUNE 12th 2013 ", " JUNE 12th 2013 ", " JUNE 12th " , "Monday, 21 JUNE, 2013" , " 21 March 2013 " , " JUNE 21, 2013 ".

Algorithm 10 Normalisation

Entrée: *L* : liste des phrases obtenues de la segmentation.
L1 : liste des jours de la semaine [saturday..Friday], [1..31]
L2 : liste des mois [January..December], [1..12]
L4 : liste des signes de ponctuation utilisés dans les dates : . / ,

Sortie: *Dates-normalisées (JJ-MM-AAAA).*
Numero-normalise.

Begin
// Parcourir les phrases appartenant à L.
si on rencontre un chiffre C **alors**
  Etudier son contexte gauche G, et son contexte droit D.
  si G ou D appartiennent à L1 ou L2 **alors**
    Normaliser_Date(C)
  sinon si C est suivi d'un signe de ponctuation appartenant à L4 **alors**
    Normaliser_numero(C).
  fin si
fin si
Fin

Nous supposons que les numéros qui peuvent figurer dans notre type de texte s'écriront de la manière suivante : " 2.5 ", "2,5 ", " 2/5 ", " 90

L'algorithme suivant permet de prendre en considération ce type d'information et de la normaliser selon une norme bien déterminée.

Entrée : *LS* : Liste des séparateurs

Sortie : *NumNormalise* // Numero Normalisé

Début

Si (C est suivi d'un séparateur X) **Alors**
  Si (X est suivi d'un num Y) **Alors**
    Si (Y n'est pas suivi d'un séparateur appartenant à Ls) **Alors**
      NumNormalise ← CXY
    Sinon
      Si (Y est suivi d'un '%') **Alors**
        NumNormalisé ← CXY%
      Sinon
        Normaliser_Date (C)
      Fin Si
    Fin Si
  Sinon
    NumNormalise ← CX
  Fin Si
Sinon
  NumNormalise ← C
Fin Si

Fin.

✍ **Identification des mots vides :**

Algorithm 11 Identification_motsvides

Entrée: *ListPh* : Liste des phrases extraites.

ListVid : Liste des mots vides de la langue anglaise.

Sortie: *Lp* : Liste des mots pleins du texte par phrase.

Début

1: Parcourir les tokens Tv des phrases appartenant à ListPh.
2: **si** Tv $\neq$ mots de ListVid **alors**
3: Lp := Tv
4: **fin si**

Fin

A.1.2 Module de résumé hybride

a. Méthode de résumé A

✍ **Similarité entre le titre et les phrases du texte :**

Un titre est construit de manière similaire à un résumé, il est donc considéré comme un représentatif de la thématique du document. Les phrases partageant des mots, ou contenant des mots, similaires/apparentés avec le titre ont plus de chances d'être importantes pour un résumé selon [22].

✍ **Position de la phrase P :**

Des travaux précédents ont montré que la position d'une phrase à l'intérieur d'un document est un paramètre très caractéristique de son importance relative [73], [41]. L'information n'est pas répartie de façon homogène tout au long d'un document mais éparpillée soigneusement par l'auteur en respectant des règles d'écriture acceptées universellement. Les débuts et les fins des documents contiennent généralement une grande partie des phrases importantes du fait de leurs buts originaux qui sont de présenter et de résumer le document. La position relative de la phrase dans le document est par conséquent utilisée comme métrique. Dénotée P(x), pour la x-ième phrase, elle est calculée à partir d'une fonction paramétrées par le nombre total de phrases m dans le document, tirée de [24] :

$$P(x) = |\frac{x - \frac{m}{2}}{\frac{m}{2}}| \tag{A.1}$$

✍ **Algorithme de décision :**

La fonction PROTEGE() nous permet une validation sémantique avec l'utilisation de l'ontologie. Ce module de traitement avec l'ontologie du domaine nécessite la tâche suivante :

– **La tâche de comparaison avec l'ontologie :** permet de comparer le terme en entrée avec les termes l'ontologie (définie avec l'outil PROTEGE) et ce conformément aux deux principaux modes de comparaison utilisés dans l'algorithme de décision (Module du résumé A) : le mode *Exact* et le mode *Fail*.

1. **Le mode Exact :** retourne vrai, si il existe une correspondance exact entre l'entrée et l'un des composants de l'ontologie ; nous aurons *matching exact*.
2. **Le mode Fail :** retourne faux, si aucune correspondance entre l'entrée et l'ontologie ; (échec de matching).

Nous appliquons par suite un test de subsumption sur les sorties (outputs) de la fonction PROTEGE. Ensuite, nous attribuons un score pour chaque mode de matching : Exact (score = 1/2), Fail (score = 0).

Algorithm 12 PROTEGE

Entrée: *D :* Entrée de la demande.

O : Sortie.

Sortie: *Résultat-comparaison*

Début

1: **si** (O = D) **alors**
2: **return** Exact
3: **sinon**
4: **return** Fail
5: **fin si**

Fin

//La fonction Protege retourne résultat de comparaison avec l'ontologie.

A.1.3 Module de détermination des entités nommées

Dans le but de déclarer nos règles d'exploration de contexte, et pour une meilleure compréhension de celles-ci, nous utiliserons les différents symboles présentés dans le tableau A.1 :

Le symbole	***Signification***	***Exemple***
?	0 ou 1 fois	
+	Répétition 1 ou N fois	
*	Répétition 0 ou N fois	
&	Concaténation	A & B : A suivi de B
|	Ou/ Choix	A | B : A ou B
[a- z]	Tous les caractères minuscules	a , b , c . . .
[A- Z]	Tous les caractères majuscules	A, B, C. . .
MotMaj	Tout mot commençant par une majuscule	Bill, Gates, Steve, Jobs, . . .
Title	Classe des titres	Sir, Dr, Pr, Mr, Mme,..
Fonc	Classe des fonctions	Teacher, designer,. . .
Natn	Classe des nationalités	French, Americain,. . .
Domain	Classe des noms des domaines	Web, IA, Programming. . .
Degree	Classe des noms des degrés	Licence, Master. . .
Prep	Classe des prépositions	During, since, ago, for, in, from..
NotionT	Classe des notions temporelles	Day, week, year, . . .
Mois	Classe des mois	January, Febrery, March. . .

NbreJour	Classe des jours	[1-31], one, two, three...
NbreAnne	Contient les entiers comme année	2013, 2012...
JourSemaine	Classe des jours de la semaine	Saturday, Sunday,...
Contr	Classe des pays	UK, USA, China, Canada,...
IndicatM	Classe d'indicateurs de monnaie	£, dollars, euro, ...
SitFam	Classe contient les situations familiale	Bachelor, Single

TABLE A.1 – Exemple des symboles utilisés.

✍ **Les règles de repérage des entités nommées de type Nom :**

Nous avons adapté la représentation d'un acteur tel qu'elle est proposée par [3] à notre domaine, afin de pouvoir étiqueter les différents acteurs en utilisant les différentes classes d'indicateurs illustrées dans la figure A.1 :

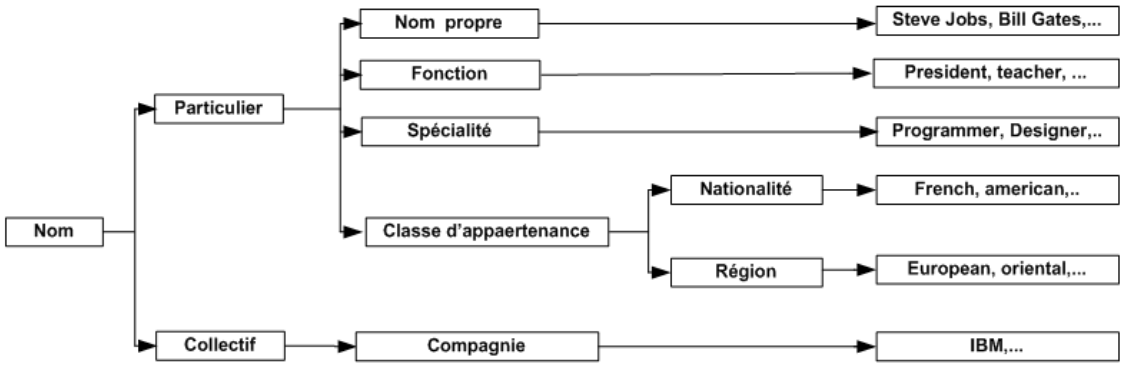


FIGURE A.1 – Diagramme général de l'information de type Nom.

Nous utilisons cette représentation afin de repérer toute information exprimant un Nom (particulier ou collectif) dans le document textuel. La figure A.2 illustre un exemple de règles définissant l'information d'un Nom particulier :

Règles d'exploration contextuelle pour l'annotation d'un Nom:

Exemple

... Professor Lee Kazuki....
... I worked with an americain organization.

Condition

<Det>? & (<Title>? | <Fonc>? | <Natn>?) & <MotMaj>+
| <[A-Z]> & <[.]> & <MotMaj>+
| <MotMaj>+ & <MotMaj>+
| <MotMoj>? & ([A-Z]+[.])+ & <MotMaj>?

Action

Annoter <Nom>

FIGURE A.2 – Exemples de règles détermination des entités nommées de type Nom.

✍ **Les règles de repérage des entités nommées de type information temporelle :**

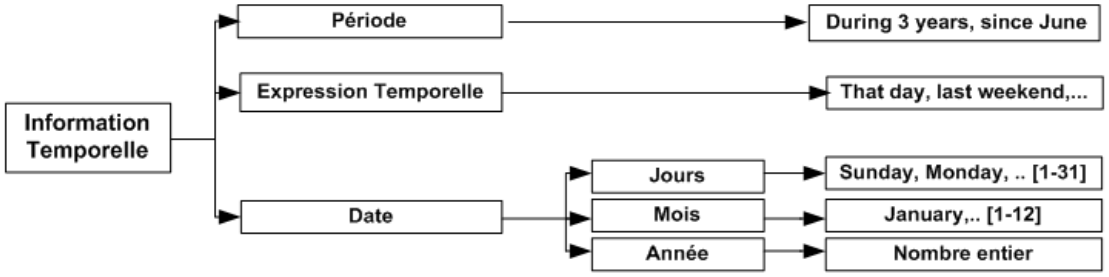


FIGURE A.3 – Diagramme général de l'information temporelle.

L'information temporelle peut être présentée par différentes manières :

– **Les Dates :** numériques (16/07/2002, 16-07-2002, 16-07-02, 16/7/02, etc.), 16 June 2002, June 1974, des dates non absolues ("25 March", "in June"), des dates absolues ("the 1st April", etc.)

– **Les durées :** des durées quelconques ("for 3 years"), des intervalles temporels (from February 11th to October 27th,, etc.), des durées absolues ("since September"), des durées relatives au moment d'élocution ("One year ago"), des durées relatives au focus temporel ("since"), atomes de temps (last year, 4 years, etc.).

– **Les expressions Temporelles :** des dates relatives au focus temporel ("3 days later"), des dates absolues, de forme particulière ("With the beginning of the year 1980"),

des dates relatives, de forme particulière (" that day, last week, in the beginning of... etc. ").

– **Les connecteurs Temporels :** then, before, later, finely...etc.

La figure A.4, présente quelques règles d'annotation d'information temporelle :

Règles d'exploration contextuelle pour l'annotation des Informations Temporelles - Date- :

Exemple

I was born on April 12, 2000 ….

Condition

<Mois> & <NbreJour> & <[,]> & <NbreAnné>
| <Prep>? & <Mois> & <[,]>? <NbreAnné>?
| <Det>? & <JourSemaine> & <[,]>? & <Mois> & <NbreJour> & <[,]>? & <NbreAnné>?
| <Prep>? & <JourSemaine>? & <Mois>? & <NbreJour>? & <NbreAnné>?

Action

Annoter <Date>

FIGURE A.4 – Exemple de règles d'annotation d'information temporelle -Date-.

```
<? xml version ="1.0" encoding ="UTF-8" standalone ="no" ?>
<Email Id =" " Sender =" " Receiver=" " Subject=" " >
  <Phrase id=" 5 ">                    <!-- Numéro de la phrase (La position de la phrase dans le texte global) -->

    <!-- Cette partie sera remplis lors du module de détermination des entités nommées -->
  <OriginalPhrase> I was born on April 12, 2000... </OriginalPhrase>

     <NamedEntity id=" 1">                          <!—L'identificateur de l'entité nommée -- >
          <Word id="5"> April </Word>               <!—L'identificateur des termes composants les entités nommées -- >
          <Word id="6">12 <Word>
          <Word id="7"> 2000<Word>
          <Type> Date </Type>                        <! – Le type de l'entité nommée -- >
          <NormalizedEntity> April 12, 2000 </NormalizedEntity>   <! – Résultat de détection de l'entité nommée -- >
     </NamedEntity>
```

FIGURE A.5 – Extrait du fichier XML représentant le résultat d'application des règles d'annotation -Date-.

Règles d'exploration contextuelle pour l'annotation des Informations Temporelles - Periode- :

Exemple

… for the last two years ...

Condition

<Det> <ExpT> <Nbr>? <NotionT>
<PrepT> <ExpT> <Nbr>? <NotionT>

Action

Annoter <Periode>

FIGURE A.6 – Exemple de règles d'annotation d'information temporelle -Periode-.

✍ **Règles de repérage des entités nommées de type information spatiale :**

Dans le but de déterminer une information spatiale, nous utilisons comme indicateurs : les prépositions spatiales, les verbes, les noms de villes... etc, qui seront exploités dans les règles de repérage des expressions spatiales. La figure A.7 présente le diagramme général de l'information spatiale :

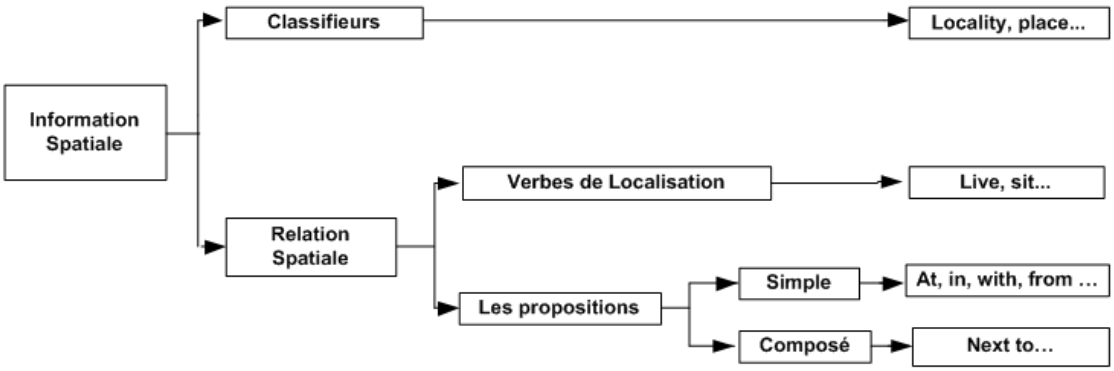


FIGURE A.7 – Diagramme général de l'information spatiale.

La figure A.8, représente un exemple de règles d'annotation concernant les informations spatiales :

Règles d'exploration contextuelle pour l'annotation des Informations Spatiales :

Exemple
I worked in China

Condition
<PrepLieu> & <Contr>
Action
Annoter <Lieu>

FIGURE A.8 – Exemple de règles d'annotation d'information spatiale.

✍ **Règles de repérage des entités nommées de type Numéro :**

Les entités numériques sont soit des entités de mesure soit des entités monétaires soit des pourcentages. Tel qu'un nombre est soit :

- *Numérique simple :* 200 £, 100 euros.
- *Numérique avec virgule :* 2,5 % .
- *Numérique composé :* 100 thousand dollars.

En voici, Figure A.9, un exemple de règle pour la détermination des informations numériques :

Règles d'exploration contextuelle pour l'annotation des Informations Numériques :

Exemple

I earn more than 3000 Euro per month

Condition

[-|+] ? & <Numero> & <IndicatM>

Action

Annoter <Numero>

FIGURE A.9 – Exemple de Règles permettant la détection des entités nommées Numériques.

RÉSUMÉ

Notre recherche se résume à la tentative constante de simplification automatique de la vie de l'utilisateur et de la réduction de l'effort cognitif fourni par celui-ci devant une activité de recherche ou d'exploration d'un domaine. Notre approche préconise un système d'extraction d'information pour la construction d'un profil utilisateur et cela à partir des e-mails électroniques dont le sujet est le recrutement dans le domaine informatique, en combinant entre plusieurs métriques statistiques et les technologies du Web sémantiques. Alors notre système doit tenir en compte toutes les informations pertinentes pouvant représenter les différents centres d'intérêts, préférences et activités d'un utilisateur.

Mots clés : Extraction d'information, Profil utilisateur, Traitement de la langue naturelle, Web sémantique.

ABSTRACT

Our research has resulted in the constant attempt to automatically simplify the user's life and reducing the cognitive effort made by this one in front of a research activity or exploration of a domain. Our approach advocates a system for extracting information to build a user profile from electronic e-mails with the subject recruitment in the computer field, combining between several statistical metrics and Web technologies semantics. So, our system should take into account all relevant information that may represent different interests, preferences and user activities.

Key words : Information Extraction, User Profile, Natural language processing, Semantic Web .