Traitement automatique d'informations appliqué aux ressources humaines

Rémy Kessler

ACADÉMIE D'AIX-MARSEILLE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

THÈSE

présentée à l'Université d'Avignon et des Pays de Vaucluse
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : Informatique

École Doctorale 380 « Sciences et Agronomie »
Laboratoire Informatique d'Avignon(EA 931)

Traitement automatique d'informations appliqué aux ressources humaines

par
Rémy Kessler

Soutenue publiquement le 10 juillet 2009 devant un jury composé de :

M.	Pierre-François Marteau	Professeur, VALORIA, Vannes	Rapporteur
M.	Patrick Gallinari	Professeur, LIP6, Paris	Rapporteur
M.	Mathieu Roche	Maître de Conférence, LIRMM, Montpellier	Examinateur
M.	Djamel Abdelkader Zighed	Professeur, ERIC, Lyon	Examinateur
M.	Gerardo Sierra	Professeur, GIL/UNAM, México	Examinateur
M.	Juan Manuel Torres-Moreno	HDR, LIA, Avignon	Directeur de thèse
M.	Marc El-Beze	Professeur, LIA, Avignon	Co-Directeur de thèse

Laboratoire Informatique d'Avignon

Résumé

Depuis les années 90, Internet est au cœur du marché du travail. D'abord mobilisée sur des métiers spécifiques, son utilisation s'étend à mesure qu'augmente le nombre d'internautes dans la population. La recherche d'emploi au travers des « bourses à l'emploi électroniques » est devenu une banalité et le *e-recrutement* quelque chose de courant. Cette explosion d'informations pose cependant divers problèmes dans leur traitement en raison de la grande quantité d'information difficile à gérer rapidement et efficacement pour les entreprises. Nous présentons dans ce mémoire, les travaux que nous avons développés dans le cadre du projet E-Gen, qui a pour but la création d'outils pour automatiser les flux d'informations lors d'un processus de recrutement. Nous nous intéressons en premier lieu à la problématique posée par le routage précis de courriels. La capacité d'une entreprise à gérer efficacement et à moindre coût ces flux d'informations, devient un enjeu majeur de nos jours pour la satisfaction des clients. Nous proposons l'application des méthodes d'apprentissage afin d'effectuer la classification automatique de courriels visant leur routage, en combinant techniques probabilistes et machines à vecteurs de support. Nous présentons par la suite les travaux qui ont été menés dans le cadre de l'analyse et l'intégration d'une offre d'emploi par Internet. Le temps étant un facteur déterminant dans ce domaine, nous présentons une solution capable d'intégrer une offre d'emploi d'une manière automatique ou assistée afin de pouvoir la diffuser rapidement. Basé sur une combinaison de systèmes de classifieurs pilotés par un automate de Markov, le système obtient de très bons résultats. Nous proposons également les diverses stratégies que nous avons mises en place afin de fournir une première évaluation automatisée des candidatures permettant d'assister les recruteurs. Nous avons évalué une palette de mesures de similarité afin d'effectuer un classement pertinent des candidatures. L'utilisation d'un modèle de *relevance feedback* a permis de surpasser nos résultats sur ce problème difficile et sujet à une grande subjectivité.

Mots clés

Traitement Automatique du Langage Naturel, Apprentissage Automatique, Recherche d'Information, Ressources humaines, modèles probabilistes, mesures de similarité.

Abstract

Since the 90s, Internet is at the heart of the labor market. First mobilized on specific expertise, its use spreads as increase the number of Internet users in the population. Seeking employment through "electronic employment bursary" has become a banality and e-recruitment something current. This information explosion poses various problems in their treatment with the large amount of information difficult to manage quickly and effectively for companies. We present in this PhD thesis, the work we have developed under the E-Gen project, which aims to create tools to automate the flow of information during a recruitment process. We interested first to the problems posed by the routing of emails. The ability of a companie to manage efficiently and at lower cost this information flows becomes today a major issue for customer satisfaction. We propose the application of learning methods to perform automatic classification of emails to their routing, combining technical and probabilistic vector machines support. After, we present work that was conducted as part of the analysis and integration of a job ads via Internet. We present a solution capable of integrating a job ad from an automatic or assisted in order to broadcast it quickly. Based on a combination of classifiers systems driven by a Markov automate, the system gets very good results. Thereafter, we present several strategies based on vectorial and probabilistic models to solve the problem of profiling candidates according to a specific job offer to assist recruiters. We have evaluated a range of measures of similarity to rank candidatures by using ROC curves. Relevance feedback approach allows to surpass our previous results on this task, difficult, diverse and higly subjective.

Keywords

Natural Language Processing, Machine-Learning, Information Retrieval, Human Ressources, Statistical Approaches, similarity measures.

Remerciements

En premier lieu, je tiens à remercier les membres de mon jury. Djamel Zighed, président du jury, mes rapporteurs Pierre-François Marteau et Patrick Gallinari mais également Mathieu Roche et Gerardo Sierra, examinateurs, pour le temps qu'ils ont bien voulu consacrer à mon manuscrit. Je tiens à leur exprimer mes remerciements les plus sincères pour les remarques qu'ils m'ont adressées et les discussions que nous avons eues qui m'ont permis d'apporter, je l'espère, plus de clarté à ce document.

Je remercie par la suite mon directeur de thèse Juan-Manuel Torres Moreno pour ces années de confiance et de franchise, pour ces nombreuses discussions que nous avons eu toutes plus enrichissantes les unes que les autres. Je ne saurais exprimer en quelques mots la reconnaissance qu'il mérite ni de l'investissement dont il a fait preuve dans cette thèse.

Je remercie vivement mon autre directeur de thèse, Marc El-Bèze pour sa disponibilité et ses conseils. Je resterais sans doute encore longtemps en admiration devant la pertinence et la justesse de ses remarques.

J'adresse des remerciements particuliers à monsieur De Boutray dont le profond intérêt pour la recherche m'a permis de saisir les nuances sémantiques profondes qu'il existe entre "avoir du travail" et "être au travail". Une pensée pour Eva, Tim, Frédéric et Nicolas que *Fox River* n'a pas encore relâchés..

Je remercie les membres du Laboratoire Informatique d'Avignon et du CERI, pour leur accueil, leur gentillesse et leurs conseils pendant toutes ces années. Voici une liste, en m'excusant par avance des oublis, des personnes que je souhaite remercier tout particulièrement : Philou, ThV, Jef, Driss, Christian, Christophe, Fred, Pierrot, Patrice, Thierry, Nathalie, Franck, Jocelyne, Simone, Florian, Boris, Eric SJ, Rodrigo, Yann et Tania. Une pensée particulière pour ma voisine de bureau, la "chica linda" Sylvia, qui parmi ses nombreuses qualités, aura réussi à me faire retenir deux mots d'espagnols ainsi que mon voisin de bureau virtuel Nicolas Béchet pour l'ensemble du travail accompli ensemble.

Je souhaite exprimer ma profonde gratitude à mes grands parents, mon parrain Jean-Luc, Véronique, Jean-Claude, Sylvie et Renée pour leur soutien indéfectible. Plus particulièrement, je remercie ma mère qui m'a toujours encouragé dans mes études ainsi que mon père d'avoir eu la gentillesse de partager ce moment de fierté avec moi.

Je veux enfin remercier les amis qui a leur façon ont contribué à la réussite de cette thèse, Eric, Olivier, Jean, Lolo, Arnaud, Jean-Christophe et Pascale, Franck et Mélanie, Elodie, Jean-Loup, Bastien et plus particulièrement Lionel, pour ces nombreuses soirées en Avignon en tête à tête devant nos ordinateurs. Ta gentillesse n'a d'égal que dans ta générosité.

Pour terminer ces remerciements, je dédie cette thèse à ma femme Laetitia et à mon fils Mickaël. Je la remercie d'avoir toujours été présente, dans les bons moments comme dans les difficiles et d'avoir toujours cru en moi. Grâce à ses encouragements et son amour, cette thèse est à présent terminée. Quant à Mickaël, garde ce sourire enjôleur et ne grandis pas trop vite...

À Mickaël...
À Laetitia...

Table des matières

Remerciements

En premier lieu, je tiens à remercier les membres de mon jury. Djamel Zighed, président du jury, mes rapporteurs Pierre-François Marteau et Patrick Gallinari mais également Mathieu Roche et Gerardo Sierra, examinateurs, pour le temps qu'ils ont bien voulu consacrer à mon manuscrit. Je tiens à leur exprimer mes remerciements les plus sincères pour les remarques qu'ils m'ont adressées et les discussions que nous avons eues qui m'ont permis d'apporter, je l'espère, plus de clarté à ce document.

Je remercie par la suite mon directeur de thèse Juan-Manuel Torres Moreno pour ces années de confiance et de franchise, pour ces nombreuses discussions que nous avons eu toutes plus enrichissantes les unes que les autres. Je ne saurais exprimer en quelques mots la reconnaissance qu'il mérite ni de l'investissement dont il a fait preuve dans cette thèse.

Je remercie vivement mon autre directeur de thèse, Marc El-Bèze pour sa disponibilité et ses conseils. Je resterais sans doute encore longtemps en admiration devant la pertinence et la justesse de ses remarques.

J'adresse des remerciements particuliers à monsieur De Boutray dont le profond intérêt pour la recherche m'a permis de saisir les nuances sémantiques profondes qu'il existe entre "avoir du travail" et "être au travail". Une pensée pour Eva, Tim, Frédéric et Nicolas que *Fox River* n'a pas encore relâchés..

Je remercie les membres du Laboratoire Informatique d'Avignon et du CERI, pour leur accueil, leur gentillesse et leurs conseils pendant toutes ces années. Voici une liste, en m'excusant par avance des oublis, des personnes que je souhaite remercier tout particulièrement : Philou, ThV, Jef, Driss, Christian, Christophe, Fred, Pierrot, Patrice, Thierry, Nathalie, Franck, Jocelyne, Simone, Florian, Boris, Eric SJ, Rodrigo, Yann et Tania. Une pensée particulière pour ma voisine de bureau, la "chica linda" Sylvia, qui parmi ses nombreuses qualités, aura réussi à me faire retenir deux mots d'espagnols ainsi que mon voisin de bureau virtuel Nicolas Béchet pour l'ensemble du travail accompli ensemble.

Je souhaite exprimer ma profonde gratitude à mes grands parents, mon parrain Jean-Luc, Véronique, Jean-Claude, Sylvie et Renée pour leur soutien indéfectible. Plus particulièrement, je remercie ma mère qui m'a toujours encouragé dans mes études ainsi que mon père d'avoir eu la gentillesse de partager ce moment de fierté avec moi.

Je veux enfin remercier les amis qui a leur façon ont contribué à la réussite de cette thèse, Eric, Olivier, Jean, Lolo, Arnaud, Jean-Christophe et Pascale, Franck et Mélanie, Elodie, Jean-Loup, Bastien et plus particulièrement Lionel, pour ces nombreuses soirées en Avignon en tête à tête devant nos ordinateurs. Ta gentillesse n'a d'égal que dans ta générosité.

Pour terminer ces remerciements, je dédie cette thèse à ma femme Laetitia et à mon fils Mickaël. Je la remercie d'avoir toujours été présente, dans les bons moments comme dans les difficiles et d'avoir toujours cru en moi. Grâce à ses encouragements et son amour, cette thèse est à présent terminée. Quant à Mickaël, garde ce sourire enjôleur et ne grandis pas trop vite...

À Mickaël...
À Laetitia...

Chapitre 1

Introduction

Nous piétinerons éternellement aux frontières de l'Inconnu, cherchant à comprendre ce qui restera toujours incompréhensible. Et c'est précisément cela qui fait de nous des hommes.
Les cavernes d'acier, Isaac Asimov, 1954

1.1 Le Traitement Automatique de la Langue

Bien que l'être humain soit capable d'analyser un discours ou un texte pour extraire et manipuler son contenu conceptuel, la quantité d'informations disponibles de nos jours avec Internet dépasse de loin ses capacités d'assimilation. Des méthodes capables de donner un sens à toute cette information deviennent indispensables. Le Traitement Automatique de la Langue Naturelle[1] (TALN) est une discipline scientifique très récente. Né aux États-Unis vers 1949[2], le TALN est dédié à la conception de méthodes et d'outils informatiques pour analyser la langue humaine. Depuis le début des travaux dans ce domaine, diverses directions de recherches ont été poursuivies. On peut notamment distinguer des approches syntaxiques liées à la théorie des langages formels et des approches numériques s'appuyant sur probabilités et statistiques. L'approche linguistique prend la phrase comme unité fondamentale d'analyse et de traitement. Ces méthodes cherchent ainsi à simuler le processus de compréhension. Les efforts ultérieurs pour traiter des textes dans leur ensemble se sont heurtés à la somme d'efforts nécessaires dans cette approche pour l'analyse des phrases puis leur intégration en un ensemble cohérent. La seconde approche est plus proche des mathématiques, et en particulier des probabilités. Plutôt que de construire des structures syntaxiques, on cherche à calculer les probabilités de co-occurrences entre mots ou expressions. Cette approche ne cherche guère à comprendre mais à reproduire une sortie adéquate. La croissance d'Internet et l'explosion en taille des banques de données ont par ailleurs favorisé la

[1] Le terme TALN provient de la traduction de NLP (Natural Language Processing).

[2] Le TALN est né pendant la guerre froide et pendant longtemps s'est concentré sur la traduction automatique avec évidemment un grand intérêt pour le passage du russe à l'anglais.

réalisation des approches numériques. Les approches que nous présentons sont majoritairement numériques avec un minimum de ressources linguistiques. Celles-ci sont robustes, permettent de traiter des masses importantes d'informations et de façon relativement rapide. Elles présentent par ailleurs l'avantage de pouvoir être adapté rapidement d'une langue à l'autre, point important pour la société partenaire, celle-ci ayant un marché international en très forte expansion.

1.2 L'Apprentissage automatique

L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens plus classiques. De façon générale, les classes sont formées des individus les plus proches selon une certaine mesure. L'apprentissage a pour but d'inculquer un comportement à un modèle en se basant sur des observations et sur un algorithme d'apprentissage. Le corpus est composés de phrases qui sont des instances du problème à résoudre et constituent les données d'apprentissage[3]. À l'issue de son entraînement, on espère que le modèle se comportera correctement face à de nouvelles situations, on parle de capacité de généralisation. Le but est de concevoir, autant que possible, des méthodes indépendantes du domaine spécifique traité ou du corpus. Les méthodes d'apprentissage automatique permettent d'identifier des relations là où la tâche serait péniblement réalisable voir infaisable à la main. L'accès à l'information textuelle (Amini, 2007) motive depuis de nombreuses années les travaux de chercheurs issus de différentes communautés comme les linguistes, les informaticiens et les statisticiens. Ces dernières années, les documents électroniques s'accumulent à grande échelle assurant une facilité de création de corpus de textes. Les collections de données textuelles font émerger des besoins fonctionnels (catégorisation de documents, constitution de ressources lexicales, système d'alerte, création de bases de connaissances, résumé automatique, etc.). De nouvelles problématiques apparaissent auxquelles les différentes communautés essaient d'apporter des réponses en adaptant leurs outils ou en en développant de nouveaux. En particulier, il est devenu important d'être capable de traiter d'énormes quantités de données textuelles, d'apporter des solutions diversifiées aux nouvelles demandes des utilisateurs, et d'automatiser les outils qui permettent d'exploiter l'information textuelle. L'apprentissage automatique propose une gamme d'outils qui permettent d'avancer dans ces directions. C'est dans ce cadre que se situe notre travail qui vise à explorer le potentiel des techniques d'apprentissage afin de résoudre les tâches particulières que sont le routage de courriels et le traitement des offres d'emplois et de leurs réponses.

[3]Une partie de ces "observations" servent aussi a valider le modèle

1.3 Contexte

Nos travaux se sont déroulés au sein de la thématique Traitement Automatique de la Langue Naturelle Écrite (TALNE) du Laboratoire Informatique d'Avignon[4] et de la société Aktor Interactive[5], agence de communication française spécialisée dans le l'*e-recrutement* dans le cadre d'une convention CIFRE de l'ANRT[6] ayant pour but le développement du système E-Gen. Les fonctions de ce système sont l'analyse et la catégorisation assistées d'offres d'emploi et un profilage automatique des réponses des candidats. En quelques années, Internet est devenu l'un des principaux sinon le principal média de recrutement. C'est un marché en très forte progression. Pourtant, ces grandes masses de données sont souvent mal exploitées car les techniques disponibles de gestion de CV's sont limitées face à l'afflux des information à traiter. Les applications de ce travail représentent un enjeu stratégique évident sur le plan économique et correspondent aussi aux centres d'intérêt de la communauté scientifique de Traitement de la Langue Naturelle, active dans le domaine de la recherche d'information.

1.4 Les problématiques abordées

Le traitement automatique des ressources humaines est une tâche faisant intervenir un grand nombre d'acteurs, comme l'illustre la figure 1.1, et génère de ce fait, une masse importante de documents afin de communiquer entre ces différents intervenants. Nous nous plaçons dans le cas où l'ensemble des communications se fait par courriers électroniques, ceci représentant la majorité des échanges dans ce domaine. L'ensemble de ces documents se présentant généralement sous format libre et généralement propriétaire, il n'existe pas de bases de données structurées dans le domaine. L'explosion du

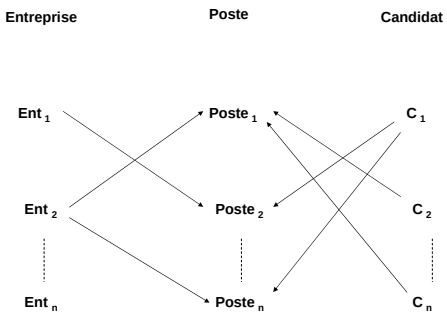


FIG. *1.1* – *Illustration des différents acteurs/objets et de leur relations.*

[4] http://lia.univ-avignon.fr/
[5] http://www.aktor.fr/
[6] Association Nationale de la Recherche et de la Technologie, http://www.anrt.asso.fr

recrutement en ligne a permis un recrutement beaucoup plus large lors de la diffusion d'une annonce mais entraîne de ce fait des temps de traitement manuel beaucoup plus importants. Cependant généralement l'entreprise qui recrute a besoin de son futur collaborateur le plus rapidement possible. Il est donc nécessaire de réduire les délais au mieux, que ce soit lors de la mise en ligne d'une annonce ou dans le tri des réponses.

La masse d'informations à traiter et les temps réduits entraînent des évaluations rapides des candidatures, pouvant occasionner des erreurs. Les Cv, comme nous le verrons par la suite, sont des documents textuels particuliers avec une structure visuelle, voire parfois esthétique, entraînant une subjectivité évidente de la part du recruteur lors d'une première évaluation rapide.

1.5 L'approche proposée

Pour aborder les problématiques évoquées, nous avons utilisé les méthodes numériques avec un minimum de ressources linguistiques. Celles-ci sont robustes, et globalement indépendante de la langue. Elles permettent la génération de modèle d'apprentissage efficace et relativement rapide en généralisation. L'ensemble des documents sources étant rédigés en langue naturelle, le TALN paraissait un choix évident. Concernant le profilage automatique de candidatures, d'autres méthodes ont été adoptées, par similarité, afin de faire face aux peu de données disponible ne permettant pas un apprentissage.

1.6 Corpus d'expérimentation et protocole d'évaluation

Les recherches en TALN demandent toujours une étape d'évaluation pour mesurer les performances des logiciels construits et valider les hypothèses. En général, les mesures d'évaluation peuvent être classées en deux catégories : les méthodes extrinsèques et les méthodes intrinsèques. Dans les premières, les sorties du système à évaluer sont jugées en se basant sur leur aptitude à accélérer la complétion d'autres tâches. À l'opposé, les mesures intrinsèques réalisent un jugement direct des résultats selon au moins l'une deux méthodes suivantes :

- manuellement en évaluant la qualité du texte produit comme la complexité de la langue ou la présence des concepts majeurs du document source ;
- automatiquement en calculant des mesures de similarité vis à vis de références produites par des humains.

Les évaluations intrinsèques automatiques sont devenues un standard de la communauté. Nous utilisons donc un vaste éventail de mesures intrinsèques pour évaluer nos travaux. Ces mesures seront détaillées dans les chapitres à venir.

À l'heure actuelle, la constitution de corpus suffisamment bien construits, en différentes langues, différents domaines et souvent visant des applications TALN spécifiques, est un champ de recherche en soit. Il existe depuis quelques années des cor-

pus d'expérimentation incontournables sur certaines problématiques qui permettent la comparaison directe des résultats entre groupes de recherche (DUC[7], DEFT[8], TREC[9], CHOI[10], etc.), cependant la plupart ne s'applique pas aux travaux qui ont fait l'objet de cette thèse. Les courriels étant généralement des données personnelles, rares sont les corpus existants. Nous n'avons pu trouver à l'époque de corpus disponible en langue française afin d'effectuer nos travaux. Nous avons donc constitué un corpus, celui-ci sera détaillé par la suite. L'ensemble des corpora d'offres d'emploi et de candidatures ont été fournis par la société Aktor Intéractive. Ils s'agit de différentes extractions du système d'information de la société. Ces corpora seront détaillés dans les chapitres où ils sont abordés.

1.7 Organisation de la thèse

Ce travail est organisé en cinq chapitres de la manière suivante. Dans le chapitre 2, nous présentons un tour d'horizon des techniques utilisées, ainsi qu'un aperçu des différents travaux liés au traitement automatique des ressources humaines et du routage de courriels. Nous allons effectuer par la suite un découpage des chapitres en fonction des modules du système E-Gen développés au cours de nos travaux (la figure 1.2 illustre ce découpage). Le chapitre 3 est consacré au module 0 et traite des différentes approches développées afin d'effectuer un routage précis de courriels à l'aide d'apprentissages supervisés, semi-supervisés et non supervisés. Le chapitre 4 présente le module 1 du système E-gen et les méthodes utilisées afin d'analyser les offres d'emploi. Ce module effectue une analyse et une catégorisation du contenu de chaque offre d'emploi afin d'en extraire différentes informations et donc automatiser sa mise en ligne. Les modules 2 et 3 sont développés dans le chapitre 5 qui présente les différentes approches que nous avons suivies afin d'effectuer un classement pertinent des candidatures en réponse à une offre d'emploi. Nous terminerons par une conclusion sur l'ensemble du travail réalisé ainsi que des possibilités d'améliorations, d'ouverture ou les perspectives que nous proposons.

[7] *Document Understanding Conferences* (DUC), sponsorisées par l'Advanced Research and Development Activity (ARDA) et organisées par le National Institute of Standards and Technology (NIST) ; `http://duc.nist.gov`

[8] Depuis 2006, le DEFT (DÉfi Fouille de Texte), `http://deft.limsi.fr`, propose des campagnes d'évaluation dans le domaine du TALN. L'édition 2008 concerne la classification en thème et en genre de textes. L'annexe E présente les résultats obtenus de nos méthodes sur cette tâche.

[9] *Text Retrieval Conference* (TREC). Lancé en 1992, son but était de soutenir la recherche au sein de la communauté de recherche d'information en fournissant les infrastructures nécessaires à l'évaluation des méthodes de recherche d'informations à grande échelle.

[10] Construit par F. Choi à partir du Brown Corpus, corpus standard en anglais sur l'actualité des États-Unis, `http://www.cs.man.ac.uk/$\sim$mary/choif/software.html`

1.8 Le système E-Gen

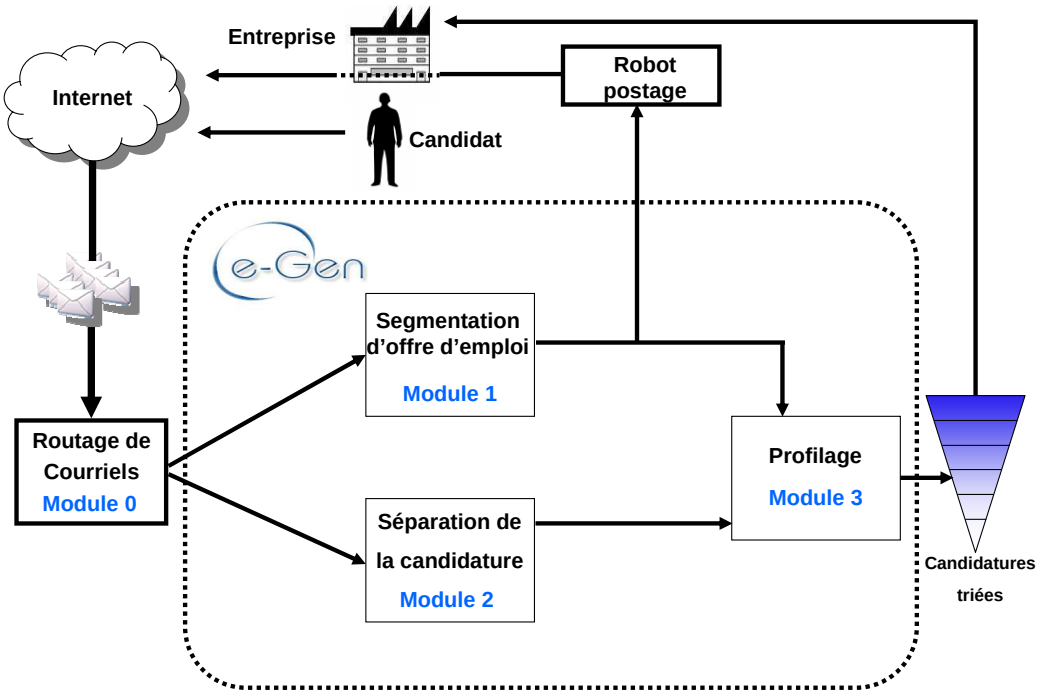


FIG. 1.2 – *Découpage en modules du système E-gen.*

Chapitre 2

Tour d'horizon

Que ce soit dans les réseaux d'entreprises ou directement sur le Web, la rapide augmentation de la quantité de données accessibles sous format électronique offre des mines d'or pour les méthodes numériques. Une grande partie de cette information est de nature textuelle, format naturel pour les humains mais d'exploitation bien plus difficile pour les systèmes d'information. Les technologies classiques de traitement de données se sont avérées insuffisantes pour leur exploitation et ont donné naissance à la recherche d'information. La technologie de Recherche documentaire ou de Recherche d'Information (RI) a été conçue dans les années 70. Une partie de celle-ci s'est orientée en Extraction d'Information (EI).Elle utilise des mesures simples de similarité entre documents textuels pour retrouver (voire catégoriser, ordonner, etc.) des documents pertinents. L'utilisateur exprime ses besoins par une requête en langage naturel, par mots-clés, sous un formalisme logique ou encore par un ensemble de documents pertinents (Salton et McGill, 1986). Dans les années 90, ces techniques se sont révélées insuffisantes pour l'exploitation de grandes bases de données textuelles, ce qui a conduit à une intensification de la recherche dans ce domaine. L'application des techniques de Traitement Automatique de la Langue Naturelle (TALN) et de l'Apprentissage Automatique (AA) ont enrichi les modèles de RI, en augmentant leur performance et en élargissant leur domaine d'application. La communauté scientifique très active dans ce domaine, continue à enrichir et à fiabiliser les outils tout en abordant de nouveaux domaines d'applications. Nous présenterons au cours de ce chapitre les différentes méthodes appliquées au routage de courriels et aux ressources humaines (RH). Nous détaillerons dans la section 2.1 les différents travaux de routage de courriel. La section 2.2 offrira un aperçu des différentes réflexions et approches consacrées aux traitements automatisés des ressources humaines. Nous présenterons enfin dans la section 2.4 les différentes approches en matière d'apprentissage automatique, celles-ci étant au cœur de nos travaux.

2.1 Routage automatique de courriels

Pour les entreprises, la gestion du courrier électronique est une activité importante aussi bien sur le plan technique que commercial. Quelques six trillions de messages électroniques sont échangés chaque année dont plus de 20% échangés à des fins commerciales, ce qui donne une indication du rôle important joué par ce mode de communication au sein des entreprises. Les flux sans cesse grandissants de courriels reçus de nos jours par chacun ont introduit de nouveaux besoins dans les domaines tels que : le routage qui implique de diffuser à des usagers ou des groupes d'usagers les informations correspondant à leurs profils d'intérêt, la détection d'opinion ou de satisfaction (Beauchene et al.), ou encore la lutte contre le *spam* qui vise à filtrer le flux entrant de courriels pour enlever les courriels indésirables ou encore les *hotlines* disponibles par courriels afin de décharger les centres d'appels téléphoniques. On distinguera cependant le filtrage (*filtering*) où il s'agit d'effectuer un choix binaire sur la pertinence (ou la non pertinence) d'un document pour une classe (comme dans le cas de la détection de *spam*), et le routage (*routing*) (Lewis et al., 1996) où il s'agit d'ordonner un ensemble de documents par rapport à leur pertinence pour différentes classes (Zaragoza, 1999). Les premiers travaux dans ce domaine suivent les recommandations de (Maes, 1994) de créer des agents intelligents et personnalisés afin de se décharger de cette tâche de classement jusqu'alors manuel. De même, différentes études (Jupiter, 2000; Banter, 2001) ont montré qu'il est essentiel pour les compagnies d'offrir un service de relation avec la clientèle qui traite les communications électroniques.
Différentes approches ont été proposées afin de répondre à cette problématique. La première est celle des systèmes basés sur des techniques issues du domaine de questions-réponses (QR). Le principe étant d'identifier la question, puis de rechercher dans une base d'information sa réponse. Ainsi, (Watanabe et al., 2004) proposent un système basé sur la collecte d'information contenue dans différentes listes de diffusion afin de déterminer une correspondance par mesure de la similarité entre la question de l'utilisateur et les réponses contenues dans les listes de diffusion. (Bélanger et Lapalme, 2004) proposent un système issu lui aussi de QR mais effectuant une analyse des courriels entrants à l'aide d'une grammaire mettant en jeux des patrons lexicaux. Ces patrons permettent d'identifier s'il s'agit d'une question ou une requête afin de pouvoir par la suite générer un courriel contenant la réponse.
On notera aussi les travaux de (Lamontagne, 2004) qui propose un système sur la base de raisonnement par cas (CBR). La seconde approche est celle basée sur les techniques d'apprentissage automatique. (Payne et Edwards, 1997) décrivent dans leurs travaux le système Magi (*Mail Agent Interface*) basé sur une combinaison de règles et d'apprentissage automatique avec l'algorithme CN2 (Clark et Niblett, 1989) et IBLP1 (Stanfill et Waltz, 1986), algorithme de type k plus proches voisins. (Segal et Kephart, 2000) avec le logiciel SwiftFile proposent une approche basée sur un apprentissage incrémental. SwiftFile effectue une représentation vectorielle des messages, puis il utilise la mesure de similarité AIM (Segal et Kephart, 1999) entre les classes. La principale conclusion apportée est qu'il n'est pas forcément nécessaire d'avoir une base d'apprentissage importante. (Zaragoza, 1999) propose le système de catégorisation et de surlignage de courriers électroniques, le *M*Boxer, avec un perceptron multi-couche (PMC)

couplé à un système de programmation dynamique. L'approche choisie pour la représentation est de prendre en compte l'ensemble des informations contenu dans le courrier électronique, en tête et corps du courrier. Les résultats obtenus, à partir du corpus de courriers électroniques *20-newsgroups* (Joachims, 1997) sont de bonne qualité (Précision / Rappel : 0,74 / 0,46) mais limités à une classification binaire (1 classe pertinente, les 19 autres non pertinentes) et inférieurs aux résultats obtenus sur le même corpus par le système *Rainbow* (Mccallum, 1998) fondé sur un classifieur Bayésien naïf. (Kiritchenko et Matwin, 2001a) proposent une méthode de classification de courriels basée sur un *co-training* (Blum et Mitchell, 1998) d'un classifieur Bayésien naïf et de machines à support vectoriel. Les résultats montrent une classification performante (entre 75 et 85% de classification correcte) à partir d'une dizaine d'itérations.

2.2 Ressources humaines, Internet et informatique

Depuis un certain nombre d'années, les ressources humaines ont fait l'objet de divers travaux dans le domaine de la recherche informatique. La gestion de ressources humaines étant souvent un processus long et coûteux pour les entreprises. Depuis les années 90, Internet joue un rôle croissant dans la coordination du marché du travail. D'abord mobilisée sur des segments spécifiques, son utilisation s'étend à mesure qu'augmente la part des internautes dans la population. (Fondeur, 2006) décrit l'évolution de ce marché au cours des dernières années et de ce qu'on appelle désormais l'*e-recrutement*.

Celui-ci repose sur l'émergence et le déploiement de sites Internet, organisés en plates-formes à deux versants, chargées de faire converger les offres et demandes de travail : les *job boards*. On citera à titre d'exemple Monster (`http://www.monster.fr/`), l'Anpe (`http://www.anpe.fr`) et HandiQuesta (`http://www.handiquesta.com`), un nouveau *job board* entièrement conçu et dédié aux personnes en situation de handicap en recherche d'emploi. D'autre part, les nouveaux acteurs que sont les agrégateurs d'offres d'emploi[1] permettent d'effectuer des recherches centralisées sur les différents *job boards* et les sites "carrières" des entreprises. À partir d'un point unique, l'accès est ouvert à un vaste éventail d'opportunités d'offres d'emploi. Les agrégateurs d'offres d'emploi ont amélioré l'accessibilité de l'information brute sur les opportunités d'emploi en diffusant des annonces d'offres d'emploi et en constituant des bases de données de curriculum vitae (CVthèques). C'est notamment en ce sens qu'une étude de l'APEC[2] (2006) conclut qu'avec Internet le marché du travail est « de plus en plus transparent ». Pour l'emploi des cadres, la part des recrutements réalisés sans publication d'offre d'emploi est passée de plus de la moitié en 1996 à seulement un tiers, dix ans

[1] Ce terme désigne les sites indexant des offres d'emploi en provenance du Web et permettant aux candidats d'effectuer des recherches centralisées. Pour la consultation des offres complètes, ces services redirigent la plupart du temps les internautes vers les sites émetteurs. Nous citons par exemple Keljob (`http://www.Keljob.com`), Optioncarriere (`http://www.optioncarriere.com/`), indeed (`http://fr.indeed.com/`) ou encore Simplyhired (`http://www.simplyhired.com/` en version francaise depuis peu à l'adresse `http://www.simplyhired.fr/`).

[2] Agence pour l'emploi des cadres, www.apec.fr

après, et toutes les offres feraient l'objet d'une diffusion Internet. Il va de soi qu'il faut relativiser ces chiffres en fonction de la profession des recrutés, l'utilisation d'Internet restant plus marginale dans le secteur de la grande distribution alimentaire contrairement à l'informatique où elle est le premier canal de recrutement.

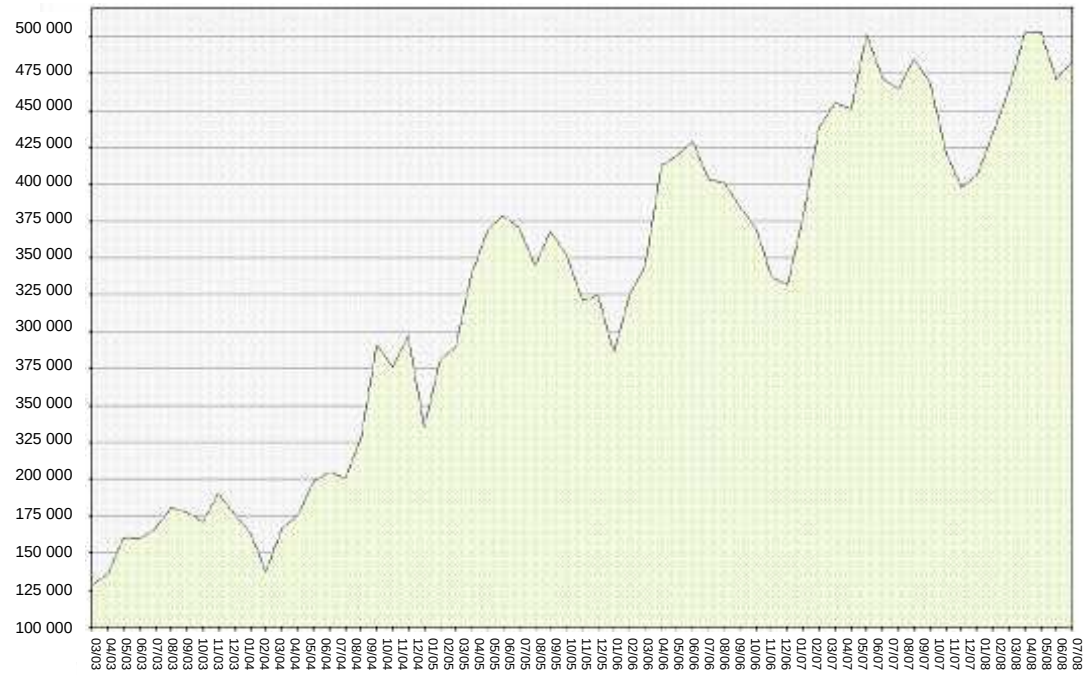


FIG. 2.1 – *Evolution du recrutement en ligne entre août 2003 et mai 2008*

Cette explosion d'informations (août 2003 : 177 000 offres, mai 2008 : 500 000 offres comme le montre la figure 2.1)[3] pose divers problèmes dans leur traitement. D'abord, l'audience élevée et hétérogène de ces « bourses à l'emploi électroniques » tend à induire un taux important de candidatures non pertinentes. Internet a engendré une banalisation de l'acte de candidature : la possibilité de se porter candidat en quelques clics a abaissé le niveau d'autocensure des candidats et a suscité un accroissement des candidatures dites « non qualifiées » (Fondeur, 2006). (Autor, 2001) avançait l'idée selon laquelle « *une conséquence naturelle de la baisse du coût de l'acte de candidature est que beaucoup de travailleurs vont postuler pour plus d'emplois. En fait, l'excès de candidatures apparaît être la norme pour les offres d'emploi déposées en ligne, avec des employeurs rapportant qu'ils reçoivent fréquemment des nombres ingérables de CV en provenance de candidats tant sur que sous-qualifiés, souvent de manière répétée* ». Ce « bruit » perçu par les recruteurs est aussi nourri par celui auquel sont confrontés les candidats face à l'accroissement du nombre d'offres d'emploi disponibles en ligne. Ces offres d'emploi publiées selon des standards différents, sans référence à des nomenclatures communes, perdent une grande partie de leur richesse lorsqu'elles sont agrégées. Il est difficile pour le candidat

[3] http://www.Keljob.com

de faire le tri parmi les annonces à partir de formulaires très largement fondés sur la recherche de mots-clés en plein texte et/ou dans le titre des postes comme le souligne (Mellet, 2006) dans son analyse des requêtes d'un agrégateur d'offres. (Beauvallet et al.) relatent ainsi dans leur étude les difficultés des internautes à trouver l'offre d'emploi du fait de la quantité d'information disponible et de son éparpillement.

2.2.1 Les approches

Nous présentons dans cette partie les différentes approches qui ont été proposées dans la littérature afin d'aborder ces problématiques, ainsi que les différentes solutions sur le marché qui commencent à apparaître.

Approches sémantiques

La spécificité des informations contenues dans les documents d'une candidature à une offre d'emploi a permis le développement d'approches sémantiques.

(Desmontils et al., 2002; Morin et al., 2004) proposent une méthode d'indexation sémantique de CV fondée sur le système BONOM (Cazalens et Lamarre, 2001). La méthode consiste à exploiter les caractéristiques dispositionnelles du document afin d'identifier chacune des parties et l'indexer en conséquence. Par ailleurs, une description d'une approche sémantique du processus de recrutements et des différents impacts économiques est proposée par (Bizer et Rainer, 2005; Tolksdorf et al., 2006) en partenariat avec le gouvernement allemand. (Rafter et al., 2000a) décrivent les lacunes des systèmes actuels face à la problématique de recherche d'emploi et proposent un système sur la base d'un filtre collaboratif (ACF) permettant d'effectuer des profilages automatiques sur le site JobFinder.

Ontologies et ressources humaines

L'idée d'utiliser des ontologies est née dans les années 90 au travers du projet *Knowledge Sharing Effort* ayant pour volonté de créer des Bases de Connaissances réutilisables et partageables. Celles-ci étant généralement un processus complexe, elles nécessitent un temps considérable à leur mise en place. Au milieu des années 90, un courant de recherche se développe autour de la terminologie textuelle. Elle préconise la construction de terminologies à partir de textes, et qui sollicite le TAL pour des méthodes et outils d'analyse de corpus (Slodzian, 2000). On appellera ces méthodes Ingénierie des Connaissances (IC). L'IC a ambition de résoudre les difficultés de construction des systèmes experts. Elle propose des concepts, méthodes et techniques permettant d'acquérir et de modéliser les connaissances dans des domaines se formalisant peu ou pas. L'IC s'intéresse au processus de construction d'ontologies pour les systèmes à base de connaissances ou pour l'interopérabilité entre systèmes dans le Web sémantique (Bourigault et Aussenac-Gilles, 2003). Elle préconise que, dans certains contextes, ce

processus s'appuie sur l'analyse de corpus de textes. On définit une ontologie comme une spécification explicite d'une conceptualisation. Ce terme fait référence à un système de concepts, autrement dit à un ensemble structuré de concepts. L'expression « spécification explicite » signifie que la conceptualisation est représentée dans un langage. Ce langage peut être une langue naturelle ou un langage formel (logique du 1er ordre ou réseau sémantique). Une ontologie est donc un ensemble de termes reliés entre eux par des relations vérifiées (vraies en toutes circonstances). Les relations les plus classiques sont les relations d'héritage (sorte-de, est-un, a-pour-spécialité, a-pour-instance), les relations de composition (est-composé-de, composant-de) ou d'autres moins explicites (spatiales, qualitatives, sémantiques etc.). La figure 2.2 présente un exemple simple d'ontologie avec une organisation hiérarchique de métiers. Ainsi le métier *Développeur* appartient à la catégorie *Informatique d'étude* , qui regroupe un certain nombre de métiers lié à l'*informatique*, etc.

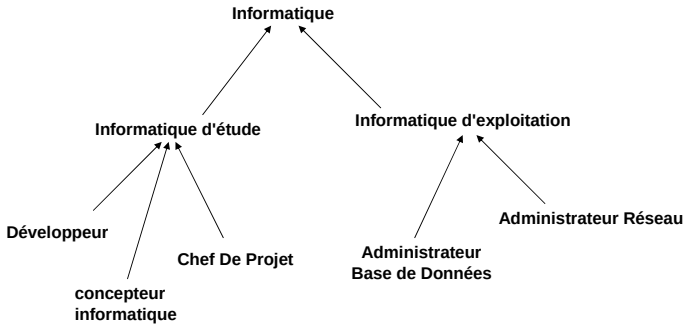


FIG. 2.2 – *Exemple d'ontologie.*

Même si l'application cible n'est pas une application textuelle, s'il s'agit de construire une ontologie pour un système informatique dont les données d'entrée sont numériques (par exemple des résultats de mesures de capteur) l'analyse de textes et la description du vocabulaire sont néanmoins primordiales pour la construction de l'ontologie. En effet, l'analyse des textes sert d'indicateur à l'organisation d'un système conceptuel et donc à la mise en relation de concepts. Par ailleurs, le choix des étiquettes de concepts doit être judicieux pour assurer l'interprétabilité et l'intelligibilité du système, ainsi que la maintenance de l'ontologie (Bachimont, 2000). Différents travaux de recherche ont été menés sur le développement d'outils d'aide à la construction d'ontologies à partir de texte, tel que l'outil TERMINO (David et Plante, 1990) précurseur dans le domaine, Xtract (Smadja, 1993) et ANA (Enguehard et Pantera, 1995) basé sur des méthodes statistiques d'extraction de collocations, ACABI (Daille, 2003) qui effectue une analyse linguistique ou encore Upery (Bourigault, 2002) basé sur une analyse distributionnelles et l'analyseur syntaxique SYNTEX (Bourigault et al., 2005).

Dans le domaine des ressources humaines, (Mocho et al., 2006) décrivent l'impor-

tance d'une ontologie commune (*HR ontology*) ainsi qu'un guide pour mettre en place ce type d'application. (Bourse et al., 2004) décrivent un modèle de compétence et un processus dédié à la gestion des compétences dans le cadre du *e-recrutement* (principalement des CV ou des offres d'emploi). De la même façon, s'appuyant sur la technologie HR-XML (Allen et Pilot, 2001), (Dorn et al., 2007; Dorn et Naz, 2007) décrivent un prototype de méta-moteur spécifique à la recherche d'emploi. Celui-ci privilégie la récolte des informations importantes (catégorie de l'emploi, lieu du travail, compétences recherchées, intervalle de salaire, etc.) sur un ensemble de sites Web (Jobs.net, aftercollege.com, Directjobs.com, etc.). Plus récemment, dans le cadre du projet Prolix, (Trog et al., 2008) proposent une ontologie de ressources humaines basée sur le cas de British Telecom. Ils proposent une architecture en plusieurs niveaux en fonction des compétences, des intéractions et du contexte.

Analyse de CV

L'étude du document principal d'une candidature, le CV ou curriculum vitae, a fait l'objet de différents travaux pour l'analyser automatiquement. (Clech et Zighed, 2003) décrivent une approche de fouille de données avec des automates capables d'apprendre à identifier des typologies de CV, de profils de candidats et/ou de postes. Les travaux présentent une approche limitée à la catégorisation de CV de cadres et de CV non cadres. La méthode s'appuie sur l'extraction de termes spécifiques permettant une catégorisation à l'aide de C4.5 (Quilan, 1993) et un modèle à base d'analyse discriminante. La spécificité de certains termes ou concepts (tels que le niveau d'étude, les compétences mises en avant) afin d'effectuer cette classification est mise en évidence mais reste décevante au niveau des résultats : 50-60% de CV correctement classés. (Roche et Kodratoff, 2006; Roche et Prince, 2008) décrivent une étude d'extraction de terminologie spécifique sur un corpus de CV[4]. Leur approche permet d'extraire un certain nombre de collocations contenues dans les CV sur la base de patrons (tels que *Nom-Nom*, *Adjectif-Nom, Nom-préposition-Nom, etc.*) et de les classer en fonction de leur pertinence en vue de la construction d'une ontologie spécialisée.

Analyse de Lettre de motivation

Peu de travaux ont traité des lettres de motivation (abrégée en LM par la suite) jusqu'a présent. La LM est généralement considérée comme un exercice de style (Knouse, 1988) et un complément d'informations du CV. Elle est généralement consultée uniquement dans des cas particuliers par les recruteurs (parcours atypique, choix entre plusieurs candidats très proches, etc.). On notera les travaux de (Audras et Ganascia, 2006) sur les erreurs usuelles dans le passage à l'écrit d'une population d'apprenants. L'approche proposée est la détection de motifs syntaxiques propres à une catégorie d'apprenants, et qui se trouvent absents ou peu usités chez les locuteurs natifs. L'étude

[4]Corpus fourni par la société Vedior Bis (http ://www.vediorbis.com).

porte en partie, sur la rédaction de LM par une population apprenant la langue française. L'analyse des résultats montre la présence d'automatismes morphosyntaxiques dans les lettres de motivation. Ces automatismes concernent aussi bien des textes d'apprenants du français que ceux des francophones, et forment des « matrices d'écriture », permettant de repérer les erreurs. On notera aussi l'étude de (Amadieu, 2007) sur un petit échantillon de CV/LM. Il conclut qu'il n'existe pas de critère discriminatoire lors d'un recrutement dû à la faible différence de traitement entre les candidats dans les entreprises testées.

Solutions sur le marché

L'ensemble de ces solutions étant commerciales, l'évaluation ou les tests n'ont pas toujours été possibles. Parmi les différentes solutions que l'on trouve sur le marché, on citera Candidatus.com[5]. Ils proposent une solution de gestion de candidatures comprenant une *indexation full text* des candidatures, et une recherche par mots-clés. De la même façon, Easy Recruitment[6] récupère le contenu des candidatures afin de constituer une base d'information permettant d'effectuer des requêtes. CVTracker, de la société Sémantis[7], est équipé de SEMIS un moteur d'analyse sémantique qui extrait automatiquement les CV qui lui sont soumis (quel que soit leur format : pdf, Word, Html, Rtf) puis catégorise chaque information. CVTracker effectue un rapprochement automatique entre les données issues des CV et des appels d'offres. Malgré de bonnes performances globales lors de l'intégration de CV ou d'offres d'emploi, on observe cependant certaines difficultés inhérentes à la sémantique dans l'interprétation (par exemple : *La société X basée à Lyon et à Paris recherche pour son bureau de Lyon* conduit le système à proposer Paris comme lieu de travail ; *enseignement en DEUG* est interprété comme une formation, etc.) ou lors d'absence d'étiquette structurant les CV (Formation, Experiences, etc.). La société RHTech[8] propose RH Premium, combinant un site de recrutement, un outil de gestion de flux de courriels de CV couplé avec le logiciel *Just a clic !* permettant la saisie automatique de CV et l'interrogation d'un moteur de recherche sémantique. ELISE de la société Wcc[9] se diversifie en plusieurs types de produits, allant de la reconnaissance d'identité à la veille d'informations (immobiliers, ventes de voitures, agences de voyages) en passant par une version RH, basée sur la correspondance entres les offres d'emplois et les candidatures. On notera aussi les travaux de Lingway[10] avec LINGWAY e-RH Applications[11] (LEA) découpé en 3 modules : LEA CV qui propose une analyse et extraction automatique des informations importantes contenues dans les CV. LEA Search permet d'effectuer une requête qui est enrichie par un thésaurus spécialisé et une ontologie puis d'interroger la base de CV ou différentes CVthèques sur Internet. LEA Offres permet d'effectuer une analyse de l'offre, permettant de re-

[5] http://www.candidatus.com/
[6] http://www.agl-informatique.fr
[7] http://www.semantis.fr
[8] http://www.rhtech.com
[9] http://www.wcc-group.com
[10] Nous tenons à remercier M. de Mazancourt pour l'ensemble des informations fournies.
[11] http://www.lingway.com

pérer les informations importantes de l'annonce et de vérifier sa conformité selon les normes actuelles (pas de discrimination de sexe ou d'âge dans le contenu de l'offre). En association avec Jobetic[12] (site d'information dédié aux personnes en recherche d'emploi), ils proposent depuis 2008 de tester la compatibilité de votre CV électronique avec une technologie de traitement automatique permettant de les intégrer dans des bases de données via le site `http://jobetic.lingway.info/`. On notera aussi Twitter[13] qui lance le site de recherche d'emploi `www.twitterjobsearch.com` basé le concept de message court (moins de 140 caractères). Le site compte environ 150 000 offres d'emploi par mois. Les internautes naviguent au fil des annonces où il est possible de les affiner par date de publication, intitulé du poste, tranche de salaire, compétences ou encore par le type de contrat souhaité. Le service propose essentiellement des offres d'emploi provenant des États-Unis mais devrait se développer en France dans un futur proche. Une solution originale et intéressante dans l'intégration de CV est proposé par ZaPoint[14] avec la génération d'une courbe pour l'expérience professionnels, personnels et la formation du candidat. Ce concept, appelé Lifechart, permet d'avoir une visualisation extrêmement rapide du parcours du candidat comme le montre la figure 2.3. On regrette cependant le manque de clarté dans la définition de l'axe des ordonnées.

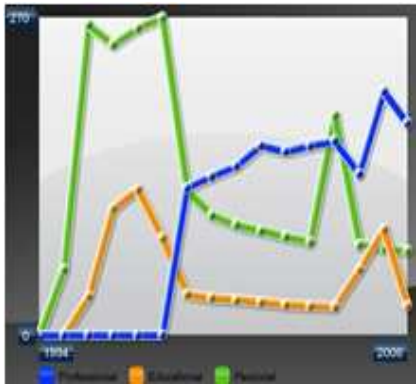


FIG. 2.3 – *Zapoint : courbes de Lifechart*

2.3 Représentation numérique des textes

Nous avons choisi de notre coté d'appliquer des méthodes numériques, pour leur rapidités et leurs efficacités. Pour appliquer ces techniques, les textes doivent être transformés en une représentation permettant de faire des calculs. La représentation numérique que nous avons utilisée au cours de notre travail, est le modèle vectoriel de (Salton et McGill, 1986). Dans cette section nous expliquerons brièvement comment utiliser ce modèle pour transformer les textes en vecteurs et comment faire face aux problèmes inhérents à de telles représentations.

[12] http ://www.jobetic.net/
[13] `http://twitter.com`
[14] `http://www.zapoint.com/`

2.3.1 Le modèle vectoriel

Le premier pas vers l'application du modèle vectoriel sur un corpus est le choix des unités textuelles ou termes d'indexation (termes, n-grammes de termes ou de caractères, expressions). Ces termes constitueront le vocabulaire. À chaque élément du vocabulaire est associé un index unique. Ensuite, on accorde un vecteur v à chaque segment de texte (une phrase, un paragraphe, un document). La dimension de ce vecteur correspond à la taille du vocabulaire et chaque composante v_i associe un poids au terme d'indice i (par exemple la fréquence d'apparition du terme i dans le segment). Dans cet espace, chaque document est représenté par un vecteur calculé à partir des unités textuelles les plus significatives présentes dans chaque segment. Du choix des termes et des segments dépend le niveau de l'analyse. Ainsi on peut souhaiter, par exemple, la comparaison entre paragraphes à partir des phrases qu'ils contiennent, ou des phrases à partir de leurs termes ou à comparer des termes à partir de leurs caractères.

Pour illustrer la démarche, nous prenons le document du tableau 2.1. Nous choisissons comme vocabulaire les termes séparés par des espaces en blancs. Dans la représentation vectorielle, chacune des trois phrases sera un vecteur dont les composantes indiquent la fréquence (n) ou l'absence (0) d'un terme dans la phrase (voir tableau 2.2). L'arrangement consécutif des vecteurs forme la matrice termes-phrases où l'on a perdu l'ordre des termes dans les phrases. C'est pourquoi cette représentation est aussi connue comme « sac de mots » (*bag of words*). La grande taille des collections de documents, ainsi que la variabilité et la complexité des informations textuelles, ne permettent généralement pas d'utiliser une représentation sophistiquée des documents, ni des modèles complexes pour traiter automatiquement et rapidement de grosses masses de données. Cette représentation rudimentaire bien que largement utilisée dans le domaine de la RI ne permet pas de "comprendre" le sens des textes. Par contre, elle présente l'avantage d'être facilement manipulable et exploitable par des techniques statistiques génériques. Par ailleurs, perdre l'ordre des termes dans les textes peut devenir gênant quand on est intéressé par des tâches comme la traduction automatique. Cependant, pour faire des calculs de fréquence ou des distributions, il est très utile d'avoir une correspondance entre les termes et les composantes des vecteurs phrases.

2002 : Baccalauréat scientifique au lycée agricole à Nîmes
Commercial terrain : Gestion d'un portefeuille de 1500 clients
(vente conseil, prospection nouveaux clients, vente additionnelle)

TAB. 2.1 – *Extrait de CV en français.*

Cet exemple utilise jusqu'à présent des poids *tf* (*tf* signifie « term frequency »). La mesure du nombre d'occurrences d'un terme dans la collection (*tf*) ne permet pas de capturer sa spécificité. Or un terme commun à de nombreux documents est moins utile qu'un terme commun à peu d'entre eux. Nous utiliserons par la suite dans certaines tâches le *Tf-idf* de (Salton et McGill, 1986) qui est une combinaison du nombre d'occurrences du terme dans le document et de la valeur inverse du nombre de documents dans lesquels il est présent (*idf* signifie «inverted document frequency») tel que défini

Phrase	2002	:	Baccalauréat	scientifique	au	lycée	agricole	à	Nîmes	Commercial	terrain	Gestion	d'un	portefeuille	de	1500	clients	(vente	conseil	prospection	nouveaux	clients,	vente	additionnelle)
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

TAB. 2.2 – *Matrice de termes-segments pour le texte du tableau 2.1. Chaque phrase devient un vecteur de présences/absences.*

dans (Manning et Schütze, 1999) :

$$Tf \cdot idf_i = \frac{n_{i,j}}{\sum_j n_j} \times \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \tag{2.1}$$

où $|D|$ est le nombre total de documents dans le corpus, $n_{i,j}$ le nombre d'occurrences du terme t_i dans le document d_j, n_j le nombre d'occurrences de tous les mots du document, et $|\{d_j : t_i \in d_j\}|$ est le nombre de documents où le terme t_i apparaît au moins une fois (c'est-à-dire tels que $n_{i,j} > 0$).

2.3.2 Réduction dimensionnelle : pré-traitements linguistiques

Une approche classique pour définir les unités textuelles dans un corpus est d'utiliser les "mots" pouvant être produits par des techniques simples de segmentation automatique. Cependant, ces unités élémentaires peuvent également faire l'objet de traitements additionnels permettant l'intégration de connaissances linguistiques plus sophistiquées dans les représentations. Dans la recherche documentaire, nous sommes intéressés par des mots *discriminants*, c'est-à-dire des mots utiles à la recherche d'information dans ces documents. Le lexique est de ce fait une composante importante de la matrice, nous utilisons divers processus afin d'amoindrir la malédiction dimensionnelle[15] qui pose de très sérieux problèmes de représentation et de traitement dans le cas des grandes dimensions. Les méthodes les plus classiques sont :

- **Uniformiser la casse** : Transformation des majuscules en minuscules ;
- **Filtrage** : suppression des verbes et des mots fonctionnels (être, avoir, pouvoir, falloir,...), des expressions courantes (par exemple, c'est-à-dire, chacun de, ...), des nombres (écrits en chiffres ou en lettres) et des symboles (comme $, #, *, etc.).
- **Racinisation** : Ce traitement peut entraîner une réduction importante du lexique. La racinisation simple consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et/ou féminins au masculin singulier[16] avant de leur associer un nombre d'occurrences. La racinisation permet donc de diminuer le nombre de termes qui définiront les dimensions de l'espace de représentation de termes ou espace vectoriel ;

[15] *The curse of dimensionality.*

[16] Ainsi les mots *développe, développent, développé, développeront, développement* et éventuellement *développeur* seront ramenés à la même forme.

– **Détection des mots composés** : Les mots composés sont repérés automatiquement puis transformés en un terme unique racinisé[17] ;

Même si une perte d'information existe lors de ces opérations (perte de structures comme les énumerations, le rapprochement délicat de certain terme ("La Poste recherche ... pour un poste de facteur"), ces opérations permettent de réduire considérablement la dimension de l'espace tout en augmentant la fréquence des termes canoniques. Le tableau 2.3 montre les trois phrases du tableau 2.1 après le pré-traitement. La matrice correspondante est montrée au tableau 2.4. La diminution dimensionnelle est évidente. Divers pré-traitements spécifique à chaque tâche ont été effectués et seront discutés par la suite.

Phrase 1 : baccalauréat scientifique lycée agricole nîmes
Phrase 2 : commercial terrain gestion portefeuille client
Phrase 3 : vente conseil prospection nouveau client vente additionner

TAB. 2.3 – *texte du tableau 2.1 après pré-traitements linguistiques.*

Phrase	baccalauréat	scientifique	lycée	agricole	nîmes	commercial	terrain	gestion	portefeuille	client	vente	conseil	prospection	nouveau	additionner
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	2	1	1	1	1

TAB. 2.4 – *Matrice réduite de termes-segments après les opérations de pré-traitement.*

On notera cependant que le mot en tant qu'unité textuelle n'est pas toujours le choix le plus approprié et il est parfois plus efficace de chercher des combinaisons de mots afin de retrouver des expressions composées. Ainsi dans notre exemple 2.1, "terrain" et "agricole" ne prenne leur sens qu'avec le mot précédent "lycée agricole" et "Commercial terrain". Il peut donc également être utile de prendre en compte des unités plus larges constituées de plusieurs mots. Cependant l'utilisation de techniques d'extraction plus sophistiquées pour les unités servant à la représentation des textes présuppose la disponibilité des ressources linguistiques nécessaires (qui ne sont pas forcément disponibles pour toutes les langues) et, de plus, augmente de façon sensible le nombre total d'unités à prendre en compte dans les étapes ultérieures de traitement. Le choix de la nature des constituants des représentations est de ce fait un compromis nécessaire entre, d'une part, la qualité et la disponibilité des outils de pré-traitement linguistique et, d'autre part, les contraintes (tailles des données, temps de traitement, ressources informatiques disponibles, etc.) imposées par les calculs à effectuer.

2.3.3 La similarité vectorielle

À partir d'une représentation matricielle, une manière de calculer la proximité entre les unités textuelles, est d'utiliser les mesures communes de similarité vectorielle comme

[17] *Date de naissance* et *expériences professionnelles* deviennent ainsi **date_de_naissance** et **expérience_professionnel**

par exemple le cosinus :

$$Cos(D_\mu, D_\nu) = \frac{\left|D_\mu \cdot D_\nu\right|}{\left\|D_\mu\right\| \left\|D_\nu\right\|} \tag{2.2}$$

où D_μ et D_ν sont les vecteurs obtenus pour les documents μ et ν. Plus l'angle qui les sépare est petit, plus l'information qu'ils portent est proche. La figure 2.4 présente un exemple avec 4 documents, D_1, D_2, D_3 et D_4. Ainsi l'angle θ_t montre des documents D_1 et D_2 relativement proches tandis que les documents D_1 et D_4 sont éloignés d'un angle θ_1 dans l'espace considéré.

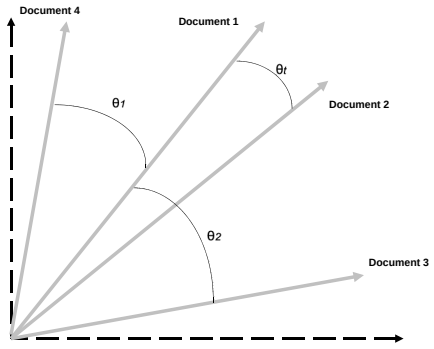


FIG. 2.4 – *Espace Vectoriel et représentation de documents.*

L'utilisation des mesures de similarité vectorielle dans l'analyse textuelle est une pratique très courante. Par exemple, (Rajman et Lebart, 1998) utilisent la distance du chi-deux (χ^2), le cosinus et la mesure de Kullback-Leibler afin d'explorer un corpus textuel. À l'aide d'une combinaison des mesures de similarité (Jaccard, Minkowski,χ^2 et cosinus), (Pincemin, 2002) proposent une application de diffusion ciblée en comparant les profils des utilisateurs avec les documents à diffuser. (Chauché et al., 2003) proposent une classification d'articles de journaux en fonction de leur thématique à l'aide du cosinus. De même, le système de résumé proposé par (Boudin et al., 2008) se base sur des variantes de la mesure cosinus et de la mesure Jaro-Winkler. On notera l'approche par mesure de similarité et logique flou de (Julio et Gil-Lafuente, 2009) dans le choix de recrutement de joueurs dans le domaine sportif en tenant compte de différents paramètres évalués par des experts.

2.4 Apprentissage automatique[18]

Cette section est un survol de ce qui ce trouve dans la littérature concernant les modèles d'apprentissage que nous avons utilisé dans notre travail. Nous distinguerons trois familles de méthodes d'apprentissage : l'apprentissage non supervisé (*clustering*), sans apport d'information *a priori*, en opposition avec l'apprentissage supervisé qui requiert un étiquetage préalable des données des classes afin que le modèle puisse s'entraîner sur celles-ci. Le mode d'apprentissage semi-supervisé manipule conjointement

[18] Cette section s'appuie principalement sur les thèses de (Amini, 2001) et (Zaragoza, 1999).

des données étiquetées et non étiquetées. La figure 2.5 illustre la sélection des données effectuée en fonction du type d'apprentissage. (Bondu et al., 2007) proposent un apprentissage « actif » combinant les différentes méthodes en interaction avec un expert humain, dans le but d'accélérer l'apprentissage.

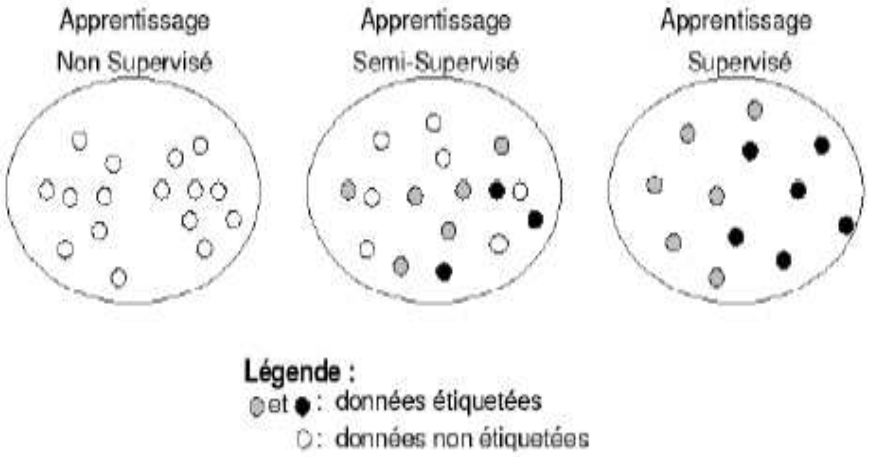


FIG. 2.5 – *Sélection des données en fonctions du type d'apprentissage.*

2.4.1 Approche non-supervisée

Lors d'une approche dite non-supervisée, le modèle utilise des données démunies d'étiquettes. Il existe différentes raisons qui justifient le choix de ce type d'apprentissage tels que le coût en temps d'élaboration d'un étiquetage manuel, la recherche des caractéristiques discriminatoires lors d'une première étude ou encore des caractéristiques qui évoluent dans le temps. L'apprentissage non-supervisé est souvent traité comme un problème d'estimation de densité. Malheureusement, c'est un problème difficile en particulier en grande dimension comme dans le cas des données textuelles. Cela a amené les chercheurs à reformuler le problème sous la forme d'un problème de partionnement (ou *clustering* (Jain et al., 1999)) des données en sous-groupes (ou *clusters*), qui est plus simple. La figure 2.6 illustre ce principe. Les données d'entrées sont présentées en 2.6(a) tandis que la figure 2.6(b) montre le partionnement souhaité.

Parmi les différents algorithmes de classification non supervisée, on citera par exemple l'algorithme des k-moyennes (MacQueen, 1967) qui a vu de nombreuses améliorations jusqu'à récemment (Huang et al., 2005; Laszlo et Mukherjee, 2006). Cette approche produit une classification « dure » des données, conduisant à un manque de précision, notamment dans le cas d'un chevauchement des classes. Son équivalent flou est l'algorithme des c-moyennes floues (fcm) (Bezdek, 1981), avec lequel chaque document est associé à chaque classe avec différents degrés d'appartenance.

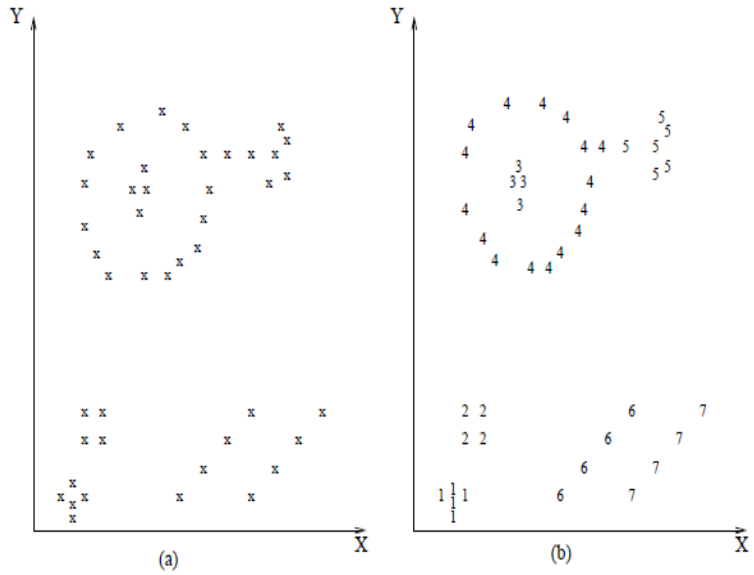


FIG. 2.6 – *Partitionnement de données en clusters.*

2.4.2 Approche supervisée

L'apprentissage supervisé est une technique d'apprentissage automatique où l'on cherche à produire automatiquement des règles à partir d'une base de données d'apprentissage contenant des exemples de cas déjà traités. Son but est donc d'utiliser cette base d'apprentissage afin de déterminer une représentation compacte d'une fonction de prédiction, qui à une nouvelle entrée x associe une sortie $g(x)$. Le but d'un algorithme d'apprentissage supervisé est donc de généraliser pour des entrées inconnues ce qu'il a pu "apprendre" grâce aux données déjà traitées par des experts. Nous présenterons ici les trois grandes approches d'apprentissage supervisé que sont les réseaux de neurones, les Modèles de Markov Cachés et les Machines à Vecteurs Support. Les premiers travaux dans le domaine sont les réseaux de neurones qui mettent en œuvre le principe de l'induction, c'est-à-dire l'apprentissage par l'expérience. Un réseau de neurones est, en général, composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente comme le montre la figure 2.7. Chaque couche i est composée de N_i neurones, prenant leurs entrées sur les N_{i-1} neurones de la couche précédente. À chaque synapse est associé un poids synaptique, de sorte que les N_{i-1} sont multipliés par ce poids, puis additionnés par les neurones de niveau i, ce qui est équivalent à multiplier le vecteur d'entrée par une matrice de transformation. Mettre les différentes couches d'un réseau de neurones l'une derrière l'autre reviendrait à mettre en cascade plusieurs matrices de transformation. Cela pourrait se ramener à une seule matrice s'il n'y avait à chaque couche, la fonction de sortie qui introduit une non linéarité à chaque étape. Les premiers réseaux de neurones à une seule couche sont le perceptron (Rosenblatt, 1958) et adaline (Widrow et Hoff, 1960), mais le plus important reste les perceptrons multicouches (PMC) qui constituent une part importante des applications des réseaux de neurones.

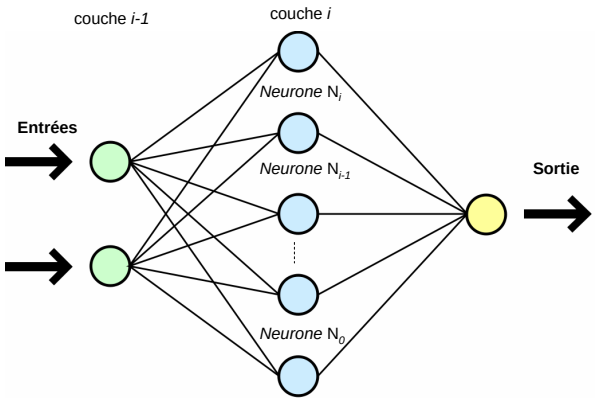


FIG. 2.7 – *Exemple d'un réseau de neurones artificiels.*

La théorie des Modèles de Markov Cachés (MMC) a été élaborée vers la fin des années soixante par (Baum et Petrie, 1966; Baum et Egon, 1967). Ces modèles ont été très étudiés pour la reconnaissance de la parole depuis le début des années soixante-dix et de façon plus récente dans des applications liées au traitement de la langue (Amini, 2001; Illouz, 2000; Kerloch et Gallinari, 2005). Un MMC est défini par deux processus stochastiques : une chaîne de Markov définie par un ensemble d'états et les transitions entre ces états, des probabilités dites d'émission associées à chaque état. Nous nous intéresserons principalement au processus décisionnel (Sigaud, 2007) qui se décrit par les éléments suivants :

- un ensemble fini S d'états discrets notés s ;
- un ensemble fini A d'actions notées a ;
- une fonction de transition $P : S \times A \rightarrow \Pi(S)$ où $\Pi(S)$ est l'ensemble des distributions de probabilité sur S. Une distribution de probabilité particulière $Pr(s_{t+1}|s_t, a_t)$ indique les probabilités pour que l'automate se retrouve dans les différents états s_{t+1} possibles quand il fait l'action a_t dans l'état s_t ;

Le formalisme de ce processus permet de décrire la structure d'un ensemble de problèmes auxquels est confronté un automate. Il permet uniquement de déterminer quelle sera la situation future de l'automate selon qu'il effectue telle ou telle action dans tel ou tel état. Les processus décisionnels de Markov définissent donc un cadre général pour représenter un problème auquel est susceptible d'être confronté un automate réalisant des actions discrètes dans un environnement discret. Pour que le problème représenté soit bien un Modèle de Markov, il faut que la fonction de transition vérifie une hypothèse connue sous le nom d'hypothèse de Markov, qui stipule que la distribution de probabilité spécifiant l'état s_{t+1} dépend uniquement de s_t et de l'action a_t, mais pas du passé. On a donc $P(s_{t+1}|s_t, a_t) = P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \cdots, s_0, a_0)$. Cela revient à dire que, si l'hypothèse de Markov est vérifiée, une connaissance des actions réalisées précédemment ou des états parcourus dans le passé ne permet pas de déterminer plus précisément avec quelle probabilité l'automate se trouvera dans chaque état possible à l'issue de son action. Le comportement de l'automate est décrit par une *Matrice de Markov* qui indique pour chaque état s selon quelle distribution de probabilité, l'automate

doit choisir les différentes actions possibles dans cet état.

Pour une tâche d'apprentissage donnée, avec un nombre fini de données d'apprentissage, la meilleure performance en généralisation (c'est-à-dire sur des données non observées pendant l'apprentissage) est obtenue si on arrive à faire un bon compromis entre la nature des données et la capacité du système choisi à apprendre ces données. L'exploration et la formulation de ces concepts ont donné naissance à une théorie dans le domaine de l'apprentissage statistique (Vapnik, 1982) d'où sont issues les Machines à Vecteurs Support (*Support Vector Machine*, SVM). Elles offrent, en particulier, une bonne approximation du principe de minimisation du risque structurel. La méthode repose sur les idées suivantes :

- les données sont projetées dans un espace de grande dimension par une transformation basée sur un noyau linéaire, polynomial ou gaussien comme le montre la figure 2.8 ;
- dans cet espace transformé, les classes sont séparées par des classifieurs linéaires qui maximisent la marge (distance entre les classes) ;
- les hyperplans peuvent être déterminés au moyen d'un nombre de points limités qui seront appelés les « vecteurs de support ».

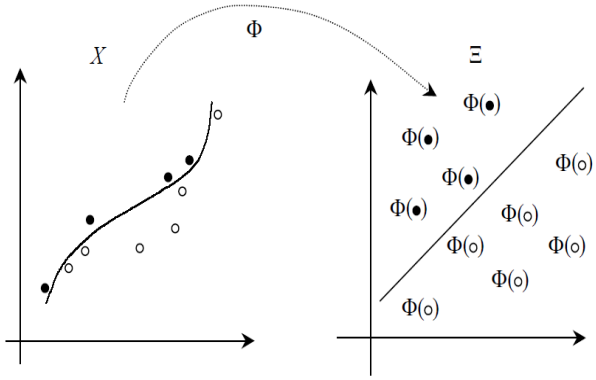


FIG. 2.8 – *À gauche, nous avons un ensemble de données non linéairement séparables dans un espace bidimensionel (x,y). À droite, au moyen d'une fonction noyau φ, ces données sont projetées dans un espace multidimensionnel, où un hyperplan sépare correctement les deux classes.*

La complexité d'un classifieur MVS va donc dépendre non pas de la dimension de l'espace des données, mais du nombre de vecteurs de support nécessaires pour réaliser la séparation. Les SVM ont déjà été appliquées au domaine de la classification du texte dans plusieurs travaux (Joachims, 1998; Grilheres et al., 2004; Vinot et al., 2003). Le classifieur MVS décrit ici est un classifieur binaire, mais il est facilement transposable pour traiter le cas multi-classes. On utilise pour cela le principe *One-against-the-Rest*, où chaque classe est comparée à l'ensemble des autres afin de trouver un hyperplan séparateur. Le cas à C classes est habituellement traité comme C classifieurs à 2 classes : pour $i = 1, \cdots, C$ on prend la classe i contre les autres classes.

Nous terminerons par le principe de Boosting qui regroupe de nombreux algorithmes qui s'appuient sur des ensembles de classifieurs binaires afin d'optimiser leurs performances. Celui-ci est issu de la combinaison de classifieurs (appelés également hy-

pothèses). Par itérations successives, la connaissance d'un classifieur faible, *weak classifier* est ajoutée au classifieur final, *strong classifier*. À chacune des exécutions de l'algorithme, une nouvelle distribution de probabilité est recalculée en fonction des résultats de l'algorithme à l'exécution précédente sur les exemples d'apprentissage. Introduit par (Freud et Schapire, 1997), l'algorithme standard s'appelle AdaBoost (*Adaptive Boosting*) devenu BoosTexter (Schapire et Singer, 2000). Une version *open source* existe aussi, icsiboost, développée par le laboratoire ICSI[19]. Il aurait été intéressant d'explorer les possibilités de celui-ci dans le cadre de ces travaux comme le montre les résultats obtenus lors de DEFT2008 (voir annexe E).

2.4.3 Approche semi-supervisée

Jusqu'à la fin des années 90, la plupart des techniques développées pour la classification reposaient sur l'apprentissage supervisé. Le principal problème que suscite l'automatisation du traitement de grandes quantités de données et le développement d'outils génériques, est la constitution et l'exploitation des bases de données nécessaires pour apprendre. Alors que les données sont extrêmement abondantes, la constitution de bases de données étiquetées nécessaires à l'apprentissage supervisé est extrêmement coûteuse et longue. Ceci est préjudiciable au développement rapide de systèmes génériques permettant avec un minimum d'interaction le traitement de différents corpus et capable de s'adapter en continu aux flux observés. Pour réduire ce temps de développement et exploiter les informations disponibles sans passer par une phase coûteuse d'ajout de connaissances, l'utilisation des données non-étiquetées est devenue primordiale.

Les premiers travaux dans ce cadre ont été motivés par le développement du Web qui a entraîné une production massive de données textuelles très hétérogènes. Ces masses de données sont généralement livrées sous forme brute, sans étiquetage *a priori* et pour les exploiter on était alors réduit à utiliser des techniques non-supervisées . C'est ainsi qu'est né l'apprentissage semi-supervisé qui a pour but d'apprendre à effectuer des tâches génériques de l'apprentissage supervisé tout en exploitant de petites quantités de données étiquetées, simultanément à de grandes masses de données brutes. Deux idées principales ont été développées. La première a été mise au point dans le cadre du traitement de signal adaptatif : il s'agit d'utiliser dans un contexte non-supervisé les sorties prédites par le système lui-même pour construire des sorties désirées. Celles-ci seront ensuite utilisées pour apprendre à partir d'une technique supervisée. Cette approche est connue sous le nom de *décision dirigée*. La deuxième idée, qui est, elle, issue de la communauté apprentissage, repose sur l'utilisation simultanée de deux classifieurs. Ceux-ci jouent alternativement le rôle de maître et d'élève dans un algorithme d'apprentissage itératif : la sortie calculée par l'un sera prise comme sortie désirée par l'autre et réciproquement, jusqu'à la convergence. Le critère d'apprentissage est ici d'optimiser la cohérence entre les deux classifieurs. Cette approche est connue sous le nom d'auto-supervision. Dans le cadre de l'apprentissage semi-supervisé, cette idée

[19] http://www.icsi.berkeley.edu/

a été reprise dans des algorithmes plus sophistiqués comme le Co-Boosting (Abney, 2008) et le Co-Training (Blum et Mitchell, 1998).

2.5 Conclusion

Nous avons présenté au cours de ce chapitre un tour d'horizon des différentes approches dans la littérature afin de résoudre les problématiques auxquels nous sommes confrontés. Nous nous sommes inspirés d'une partie de ceux-ci et éloigner d'autres afin de développer les différentes parties du projet E-Gen comme nous le verrons dans les chapitres suivants. Nous avons décrit par la suite succinctement les différentes techniques que nous avons utilisés dans nos travaux.

Chapitre 3

Routage de courriel dans une entreprise

3.1 Introduction

Les nouvelles formes de communication écrite (courriels, forums, chats, SMS, etc.) posent des défis considérables aux systèmes de traitement automatique de la langue. Ces données présentent des phénomènes linguistiques bien particuliers : messages trop courts, très bruités, etc. Il est possible d'observer des phénomènes linguistiques bien particuliers comme les émoticônes[1], les acronymes, les fautes (orthographiques, typographiques, mots collés, etc.) d'une très grande morpho-variabilité et d'une créativité explosive. Ces phénomènes doivent leur origine au mode de communication (direct ou semi-direct), à la rapidité de composition du message ou aux contraintes technologiques de saisie imposées par le matériel (terminal mobile, téléphone, etc.). Nous désignons par **phonécriture** ou **phonécrit** toute forme écrite qui utilise un type d'écriture phonétique sans contrainte ou avec des règles établies par l'usage[2]. Les travaux ci- dessous sont destinés à créer des outils et des ressources génériques pour la classification de courriels. La capacité d'une entreprise à gérer efficacement et à moindre coût, ces flux d'informations devient un enjeu majeur de nos jours pour la satisfaction des clients. Nous nous attacherons à traiter dans ce chapitre des problèmes posés par le routage précis de courriels.

Le traitement automatique des courriels est extrêmement difficile à cause de son caractère imprévisible (Beauregard, 2001; Kosseim et Lapalme, 2001; Kosseim et al., 2001; Cohen, 1996a) : des textes trop courts (moyenne de 11 mots par courriel dans nos corpus), souvent mal orthographiés et/ou pauvres. Ceci impose donc d'utiliser des outils de traitement automatique robustes et flexibles. Les méthodes d'apprentissage automatique à partir de textes (fouille de documents), permettent d'apporter des solutions par-

[1] Symboles utilisés dans les messages pour exprimer les émotions, par exemple le sourire **:-)** ou la tristesse **:-(**

[2] Par exemple **kdo** à la place de cadeau, **10ko** pour dictionnaire, **A+** à plus tard, **@2m1** à demain, etc.

tielles aux tâches évoquées. Elles semblent bien adaptées aux applications de filtrage, de routage, de recherche d'information, de classification thématique et de structuration non supervisée de corpus. Ces méthodes présentent, de surcroît, l'intérêt de fournir des réponses adaptées à des situations où les corpus sont en constante évolution ou bien contiennent de l'information dans des langues étrangères. Ce chapitre propose l'application des méthodes d'apprentissage afin d'effectuer la classification automatique de courriels visant leur routage, en combinant techniques probabilistes et machines à vecteurs de support (*Support Vector Machines* MVS). Des approches fondées sur des mots et des *n*-grammes de caractères (Jalam et Chauchat, 2002; Miller et al., 2000) ont été testées. La catégorisation thématique est au cœur de nombreuses applications de traitement de la langue. Ce contexte fait émerger un certain nombre de questions théoriques nouvelles, en particulier en relation avec la problématique du traitement d'informations textuelles incomplètes et/ou très bruitées (Kosseim et al., 2001). En premier, nous présentons en section 3.2 le positionnement du problème, puis par la suite les différents pré-traitements linguistiques utilisés afin d'obtenir une représentation vectorielle de chaque courriel. Ensuite, nous présentons les différents algorithmes d'apprentissages utilisés avant de conclure.

3.2 Positionnement du problème

Nous nous plaçons dans le cas où dans une entreprise une boîte aux lettres reçoit un grand nombre de courriels correspondant à plusieurs thématiques. Une personne doit lire ces courriers et les rediriger vers le service concerné (les courriels de problèmes techniques vers le service technique, ceux pour le service après vente seront redirigés vers ce dernier, etc.). Il s'agit donc de développer un système pour automatiser cette tâche.

Il existe des approches pour effectuer le traitement automatique de courriers électroniques écrits en anglais (Kiritchenko et Matwin, 2001b; Kosseim et al., 2001; Cohen, 1996b; Brutlag et Meek, 2000), ainsi qu'une approche à base de règles s'inspirant d'une méthode de question-réponse (Bélanger et Lapalme, 2004). Cependant, comme il s'est avéré difficile de trouver des travaux sur les courriels ou de corpora en français (des corpus en anglais existent cependant pour de la classification de *spams*). Pour générer les ensembles d'apprentissage et de test, nous avons donc décidé de créer une adresse électronique et de l'abonner à diverses listes de diffusion[3] ou newsletters[4] de thèmes variés. Il faut noter que l'évaluation de notre système s'effectue en fonction de la liste de diffusion émettrice. De ce fait, nous obtenons directement un partitionnement de référence où le message ne peut appartenir à plusieurs thématiques. Les corpus réalistes qui ont été ainsi collectés (libres de *spams*) présentent un certain nombre de caractéristiques particulières qui sont reportées dans le tableau 3.1. On remarque que la majorité des courriels sont des messages courts ($\approx$ 70%) et que la répartition du nombre d'auteurs est assez importante (70% des auteurs ont envoyé moins de 6 courriels).

[3]Football, jeux de rôles, ornithologie, cinéma, jeux vidéo, poème, humour, etc.
[4]Sécurité informatique, journaux, matériel informatique, etc.

Nb. total de courriels	P=2000	
Nb. total de mots bruts	956757	
Nb. de courriels avec pièce jointe	140	7%
Nb. de courriels court ($\leq$ 11 mots)	1379	69%
Nb. de courriels long (> 11 mots)	621	31%
Taille moyenne d'un courriel en mots		11
Nb. d'auteurs différents	298	
Auteurs ayant émis moins de 2 courriels	125	42%
Auteurs ayant émis entre 2 et 5 courriels	83	28%
Auteurs ayant émis entre 6 et 10 courriels	33	11%
Auteurs ayant émis plus de 10 courriels	57	19%

TAB. *3.1* – *Statistiques du corpus de courriels.*

3.3 Méthodes

3.3.1 Suppression de microspams et phonécriture

Avant d'appliquer les pré-traitrements classiques (2.3.2), un processus se charge de supprimer automatiquement la micro publicité (microspams) qui n'apporte aucune information permettant de catégoriser le courriel mais, au contraire, ajoute du bruit risquant de gêner cette catégorisation. Il s'agit en général (à plus de 95% de cas) de publicités rajoutées au bas des courriels par les fournisseurs de service de messagerie électronique comme le montre l'exemple suivant :

```
___[ Pub ]_____________________________________________________
Inscrivez-vous gratuitement sur Siteweb, Le site de rencontres!
http ://www.site.com/index.php?origine=4
```

Nous avons supprimé la micro publicité propre au corpus, celle-ci se présentant, la plupart du temps, sous la forme de liens HTML vers des pages Internet. À l'aide d'un dictionnaire constitué à partir de sites[5] et décrivant les divers termes de phonécriture, nous remplaçons ceux-ci par leurs équivalents en langue française. Cette étape de « traduction » est réalisée avant la suppression de la ponctuation et la suppression des majuscules car beaucoup de termes phonécrits sont composés à l'aide de ponctuation (en+ pour En plus, @2m1 pour À demain) ou de combinaisons de majuscule/minuscule (NRV pour Énervé, Xcellent pour excellent). Il subsiste cependant un certain nombre d'ambiguïtés difficilement détectables sans le contexte de la phrase et en utilisant uniquement un dictionnaire, tel que « 10Ko » pour dictionnaire, mais qui peut être aussi une taille de fichier, ou encore certains chiffres comme le 7 utilisé soit comme nombre

[5] http ://www.mobimelpro.com/portail/fr/my/dictionnaire_sms.asp
http ://www.mobilou.org/10kosms.htm
http ://www.affection.org/chat/dico.html

soit pour le déterminant démonstratif « cette », 9 pour le chiffre ou l'adjectif qualificatif « neuf » etc.

3.3.2 Dictionnaire avec et sans accents

Dans un premier temps, nous avons effectué l'ensemble des filtrages avec un dictionnaire avec accents. L'usage des caractères accentués n'étant pas systématique dans les courriels, nous avons adapté en conséquence notre dictionnaire. Nous avons amélioré la qualité de notre filtrage en utilisant un dictionnaire composé des deux précédents (avec et sans accent) : sur un corpus de 2000 messages, $\approx$ 43 000 lemmatisations contre $\approx$ 40 000 lemmatisations précédemment. L'ensemble de ces opérations de prétraitement ayant réduit la taille du corpus d'environ 70% (sur un corpus d'environ 950 000 occurences au départ, nous obtenons un corpus final de $\approx$ 250 000 occurences après ces opérations) et les performances ont été legèrement améliorées.

3.3.3 Calcul de la distance entre vecteurs

Afin d'effectuer la classification, nous calculons la distance entre les vecteurs et les centroïdes. Nous avons utilisé, pour cela, la distance de Minkowski (3.1). Nous avons fait une implantation de celle-ci avec $\lambda = 1$ (distance de Manhattan), cependant les résultats obtenus étant décevants, nous avons utilisé $\lambda = 2$ (distance euclidienne). Chacune de ces distances ayant leurs avantages et leurs inconvénients en fonction des données à traiter.

$$d^{\lambda}(a,b) = \left(\sum_i ||a_i - b_i||^{\lambda} \right)^{1/\lambda} \tag{3.1}$$

3.4 Observation de la matrice

La figure 3.1 présente une répartition des termes en fonction des courriels. Les classes ont été affichés selon leur distribution originale dans le corpus, ce qui explique la division de certaines classes. L'axe des ordonnées représente la quantité de nouveau termes apparaissant dans le corpus tandis que chaque courriel est représenté par son numéro sur l'axe des abscisses. Les droites délimite l'appartenance des courriels en fonction de leur classe respective $C1$,$C2$,$C3$,$C4$. Sur ce graphique, on observe que la catégorie 4 est divisée en 2 parties (courriels 1 à 25 puis courriels 100 à 125). La densité des 1000 premiers termes redevient importante entre le 100ème et le 125ème termes, mais ailleurs elle reste assez faible. Ceci est observable aussi pour la catégorie 2 (entre le courriel 25 et 50 puis entre le courriel 125 et 150) où l'on remarque très bien l'absence de termes des catégories 1, 3 et 4 dans le second morceau (faible densité des termes 1500 à 2500). On observe par ailleurs que lors de l'apparition d'une nouvelle catégorie, les nouveaux termes sont en forte densité dès le début de celle-ci.

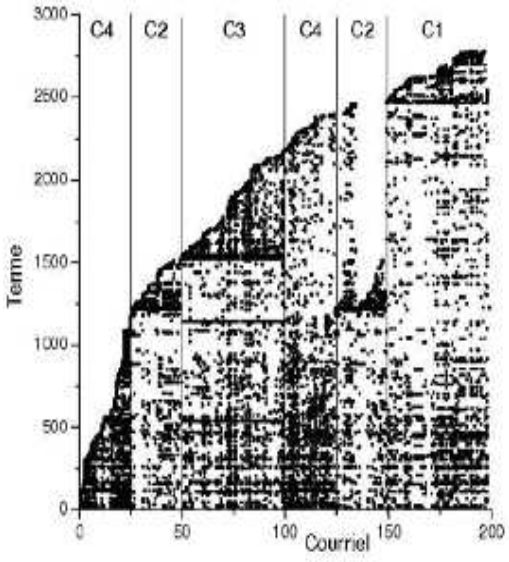


FIG. 3.1 – *Répartition des termes dans les catégories C_i en fonction des courriels.*

3.5 Apprentissage non supervisé

L'algorithme *Fuzzy k-means* (Bezdek, 1981; deGruijter et McBratney, 1988) permet d'obtenir un regroupement des éléments par une approche floue avec un certain degré d'appartenance, où chaque élément peut appartenir à une ou plusieurs classes, à la différence de *k-means*, où chaque exemple appartient à une seule classe (partition dure). *Fuzzy k-means* minimise la somme des erreurs quadratiques avec les conditions suivantes :

$$\sum_{k=1}^{c} m_{\mu k} = 1; \quad \mu = 1, \cdots, P \tag{3.2}$$

$$\sum_{k=1}^{c} m_{\mu k} > 0; \quad m_{\mu k} \in [0,1]; \quad k = 1, \cdots, c; \quad \mu = 1, \cdots, P \tag{3.3}$$

On définit alors la fonction objectif :

$$J = \sum_{\mu=1}^{P} \sum_{k=1}^{c} m_{\mu k}^{f} d^{\lambda}(\Gamma^{\mu}, \beta^{k}) \tag{3.4}$$

où P est le nombre de données dont on dispose, c est le nombre de classes désirées, β^k est le vecteur qui représente le centroïde (barycentre) de la classe k, Γ^μ est le vecteur qui représente chaque exemple μ et $d^\lambda(\Gamma^\mu, \beta^k)$ est la distance entre l'exemple Γ^μ et β^k en accord avec la distance (3.1) et que nous écrirons $d^\lambda_{\mu k}$ afin d'alléger la notation. f est un paramètre, avec une valeur comprise dans l'intervalle [2,∞] qui détermine le

degré de flou (*fuzzyfication*) de la solution obtenue *in fine*, contrôlant le degré de recouvrement entre les classes. Avec $f = 1$, la solution deviendrait une partition dure (du style *k-means*). Si $f \to \infty$ la solution approche le maximum de flou et toutes les classes risquent de se confondre en une seule. La minimisation de la fonction objectif J fournit la solution pour la fonction d'appartenance $m_{\mu k}$:

$$m_{\mu k} = \frac{d_{\mu k}^{\lambda/(f-1)}}{\sum_{j=1}^{c} d_{\mu j}^{\lambda/(f-1)}} \; ; \; \beta^k = \frac{\sum_{\mu=1}^{P} m_{\mu k}^f \Gamma^\mu}{\sum_{\mu=1}^{P} m_{\mu k}^f} ; \; \mu = 1, \cdots, P; \; k = 1, \cdots, c \tag{3.5}$$

L'intérêt d'utiliser *Fuzzy k-means* dans le cadre de la classification thématique de courriers électroniques consiste à router un message vers un destinataire prioritaire (celui avec le degré d'appartenance le plus élevé) et en copie conforme (Cc) ou cachée (Bcc) vers celui (ou ceux) dont le degré d'appartenance dépasse un certain seuil empirique établi à l'avance. L'annexe G présente l'algorithme *Fuzzy k-means*.

3.6 Apprentissage supervisé

Nous avons testé plusieurs implantations différentes des machines à vecteurs de support (SVMTorch[6], WinSVM[7], M-SVM[8]) ainsi qu'un système d'arbre à classification sémantique (Lia_sct (Béchet et al., 2000)) afin de pouvoir sélectionner la plus efficace. Nous avons finalement décidé d'utiliser une implantation de l'algorithme de Collobert (Collobert et Bengio, 2000), SVMTorch, qui permet une approche multi-classes des problèmes de classification, qui est portable et qui a posé le moins de soucis techniques lors de l'intégration. On utilise pour cela une fonction appelée noyau, qui permet de projeter les données dans un espace de grande dimension de la façon suivante : sous les conditions de Mercer (Vapnik, 1995), le produit scalaire dans le nouvel espace peut être réécrit au moyen d'une fonction noyau $K(a, b)$ telle que $K(a, b) = (\Phi(a)) \bullet (\Phi(b))$. SVMTorch permet de tester plusieurs types de fonctions noyaux :

- Linéaire (polynôme de premier degré) ;
- Polynomial de degré d $(a, b) \to (a \cdot b)^d$;
- Gaussiennes à base radiale (FBR) $(a, b) \to \exp\left(-\frac{||a-b||^2}{2\sigma^2}\right)$;
- Fonctions d'activation sigmoïdales $(a, b) \to \tanh(sa \cdot b + r)$.

Les résultats des tests présentés dans la section 3.9, ont été obtenus à l'aide d'une fonction à noyau linéaire, qui s'est avérée la plus performante. Lors des séries de tests que nous avons effectués, les erreurs d'apprentissage et de test ont été généralement plus importantes avec les autres fonctions noyaux.

[6] http://www.idiap.ch/
[7] http://liama.ia.ac.cn/PersonalPage/lbchen/winsvm.htm
[8] http://www.loria.fr/~guermeur/

3.7 Initialisation aléatoire ou semi-supervisée ?

K-means et *Fuzzy k-means* sont des algorithmes performants mais fortement dépendants de l'initialisation (Fred et Jain, 2003). Nous étions donc confrontés au problème de l'initialisation des centroïdes β^k. Nous avons d'abord testé la méthode avec des initialisations aléatoires, mais l'erreur d'apprentissage, ϵ_a, était autour de 25% dans le meilleur des cas (voir figure 3.2). Ceci est dû au fait que l'algorithme semble piégé dans des minima locaux. Nous avons donc décidé d'initialiser, de façon semi-supervisée, en prenant un petit nuage d'exemples (avec leur classe) afin d'avoir des points de départ mieux situés pour nos centroïdes. Nous avons fait une étude de cette initialisation semi-supervisée. Sur la figure 3.2, sont illustrés les résultats que nous avons obtenus sur 10 ensembles d'apprentissage tirés au hasard. La comparaison entre l'initialisation aléatoire et celle semi-supervisée montre que cette dernière est nettement supérieure. On remarque que l'initialisation aléatoire a des résultats irréguliers contrairement à celle semi-supervisée. Il est cependant important de rappeler que l'apprentissage avec *k-means* est toujours non supervisé, et qu'il suffit d'initialiser avec un nombre d'exemples entre 10% et 20% pour obtenir $\epsilon_g < 10\%$.

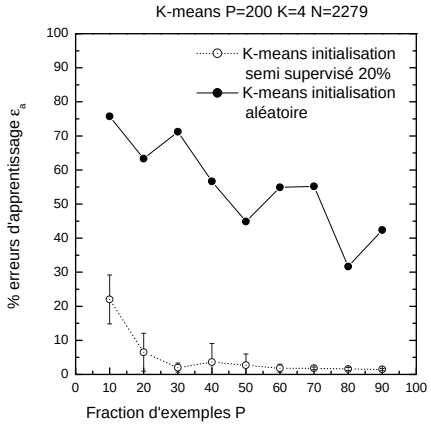


FIG. 3.2 – *Comparaison entre l'initialisation aléatoire et l'initialisation semi-supervisée.*

Nous avons voulu par ailleurs, connaître l'incidence du paramètre de flou f afin d'améliorer les résultats. Nous avons donc effectué une série de tests en ne faisant varier que ce paramètre, f allant de 2 à 50. Les résultats de la figure 3.3 montrent qu'au delà d'une valeur de 10, les variations sur ϵ_g sont négligeables. Nous avons retenu $f = 6$ comme valeur finale. Cette valeur reste cependant empirique et dépendante des données.

3.8 La méthode hybride

Nous avons décidé de combiner les méthodes d'apprentissage supervisé et non supervisé afin de tirer parti des avantages de chacune d'entre elles. En effet, *k-means* étant

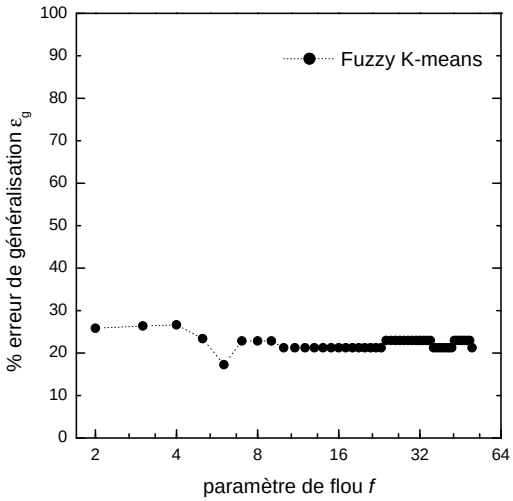


FIG. 3.3 – *Incidence du paramètre de flou f sur l'erreur de généralisation ϵ_g. $P = 160$ courriels en apprentissage,* 40 *en test.*

un algorithme exploratoire, il permet de ne pas avoir besoin d'un corpus étiqueté mais présente l'inconvénient de résultats en dent de scie en fonction de l'initialisation. D'un autre côté, l'apprentissage avec les MVS est supervisé et comme on le sait, il est très coûteux d'avoir de grands ensembles de données étiquetées. Nous avons donc mis en place une méthode hybride combinant les deux classifieurs : avec un très petit échantillon (une vingtaine de courriels), nous calculons les centroïdes de *k-means*. Nous effectuons ensuite un apprentissage non supervisé avec *k-means* sur la matrice γ_1[9] qui fournit la classe prédite pour chaque courriel. La deuxième étape consiste à présenter γ_1 à la machine à vecteurs de support, celle-ci pouvant dès lors effectuer un apprentissage supervisé à l'aide des étiquettes fournies par *k-means*. La généralisation est effectuée par la méthode MVS sur l'ensemble γ_2 à partir des vecteurs de support trouvés précédemment. La figure 3.4 montre la chaîne de traitement complète. Les premiers résultats ont montrés que la qualité des vecteurs supports obtenus était fortement dépendant des centroïdes initiaux de *k-means* : une mauvaise initialisation (problèmes de minima locaux) entraîne un mauvaise étiquetage de la matrice γ_1 par *k-means* et de ce fait un mauvaise apprentissage des MVS et des taux d'erreurs sur γ_2 extrêmement important. L'initialisation des centroïdes a permis de compenser cela comme le montre les résultats en section 3.9. la figure 3.9 montre l'influence de la taille de cette échantillon d'initialisation.

[9]Un tirage aléatoire a été réalisé afin de constituer les matrices d'apprentissage γ_1 et de test γ_2. Ainsi, plusieurs tests statistiquement indépendants ont été effectués.

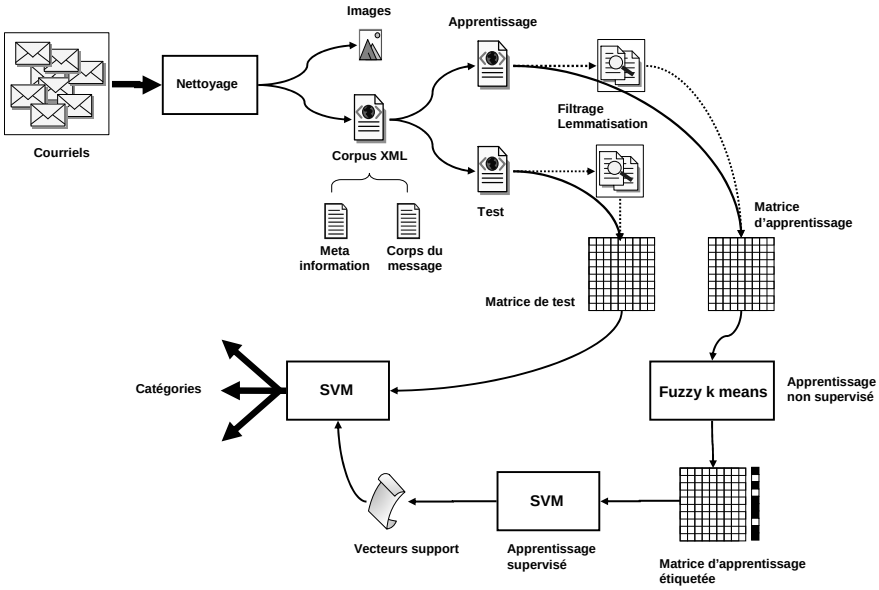


FIG. 3.4 – *Méthode hybride, chaîne de traitement complète : classification d'un corpus de courriels en k catégories.*

3.9 Résultats sur le routage de courriels

Nous avons travaillé avec des corpus de N termes contenus dans $P = \{200, 500, 1000, 2000\}$ courriels ayant $k = 4$ classes prenant leur valeur dans l'ensemble {**football, jeux de rôles, cinéma, ornithologie**}. Chacun des tests a été effectué à 50 ou 100 tirages aléatoires, dans le cas de n-grammes de caractères ou de mots, respectivement.

Soit Λ un ensemble de P exemples avec leur classe d'appartenance τ^μ. Soit γ_1 un sous-ensemble d'apprentissage de P_1 exemples tirés au hasard, et γ_2 le sous-ensemble de test indépendant de P_2 exemples, tel que $\Lambda = \gamma_1 \cup \gamma_2$ et $P = P_1 + P_2$. Un exemple mal classé par un classifieur est un exemple dont la classe attribuée τ^μ n'est pas correcte. Ce type d'erreur peut être estimé en phase d'apprentissage ϵ_a ou de généralisation ϵ_g. Ce sont des valeurs comprises entre 0 et 1, ou exprimées en pourcentage. Pour mesurer les performances des algorithmes de classification, nous avons utilisé les expressions suivantes afin d'estimer ϵ_a et ϵ_g :

$$\epsilon_a = \frac{\text{Nb. exemples } \in \gamma_1 \text{ mal classes}}{card\{\gamma_1\}}; \epsilon_g = \frac{\text{Nb. exemples } \in \gamma_2 \text{ mal classes}}{card\{\gamma_2\}} \tag{3.6}$$

où $card\{\bullet\}$ représente le nombre d'éléments d'un ensemble. Ces mesures sont utilisées pour l'ensemble des tests concernant la classification de courriels.

3.9.1 Résultats

La figure 3.5 présente les résultats obtenus sur un corpus de $P = 500$ courriels. À gauche, nous montrons l'erreur d'apprentissage ϵ_a avec une initialisation semi-supervisée pour *k-means*. À droite, nous montrons l'erreur en généralisation ϵ_g de MVS avec un apprentissage supervisé. Bien sûr, le taux d'erreur ϵ_g est faible : inférieur à 10% au delà de 50% des exemples appris. Cela correspond à la meilleure situation possible en apprentissage, mais les MVS nécessitent une classification préalable des exemples de l'ensemble d'apprentissage, ce qui n'est toujours pas disponible.

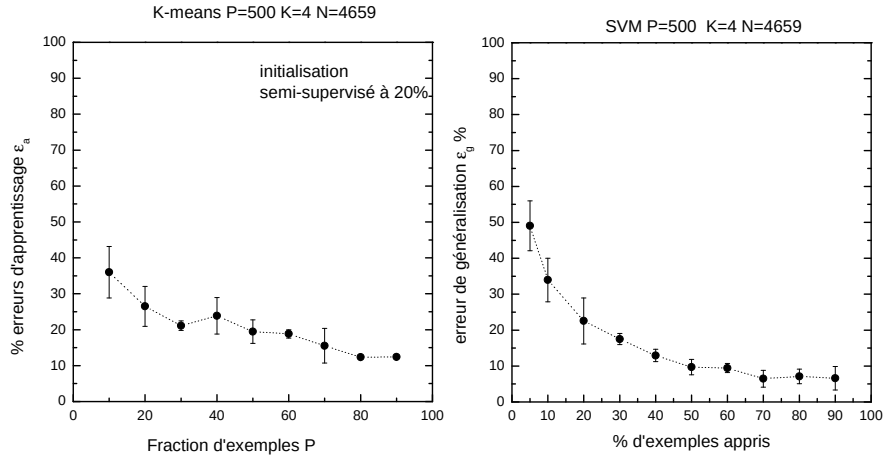


FIG. 3.5 – *Erreur d'apprentissage ϵ_a pour k-means à gauche et de généralisation ϵ_g MVS à droite, $P = 500$ courriels. k=4 classes, N dimension des matrices.*

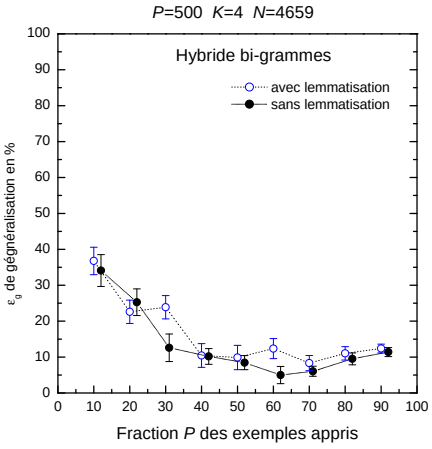


FIG. 3.6 – *Erreur de généralisation pour la méthode hybride avec bi-grammes de lettres (50 tirages aléatoires) sur un corpus de $P = 500$ courriels lemmatisés et non lemmatisés.*

La figure 3.6 présente les résultats obtenus sur un corpus de $P = 500$ courriels et un découpage en n-grammes de caractères, avec et sans lemmatisation. Nous avons

effectué des tests avec $n = \{2, 3, 5\}$ et nous avons trouvé que les bi-grammes produisent les meilleurs résultats avec nos corpus et sensiblement comparables à ceux obtenus avec un modèle uni-gramme de mots. Nous montrons l'erreur en généralisation ϵ_g pour la méthode hybride.

Les figures 3.7 et 3.8 comparent les résultats de la méthode hybride et des MVS sur des corpus de $P = \{200, 500\}$ et de $P = \{1000, 2000\}$ courriels respectivement. Dans le cas hybride, nous avons combiné un apprentissage non supervisé par *k-means* (initialisation semi-supervisée de $0.05P$ ou $0.2P$ courriels) et supervisé pour SVM. Nous constatons que la performance ne se détériore pas en augmentant la taille du corpus. Il est possible de voir aussi que les performances en généralisation de la méthode hybride sont très proches de celles des MVS en apprentissage supervisé. Notons que dans toutes les courbes des figures 3.9 et 3.10 l'unité de base est un uni-gramme de mot.

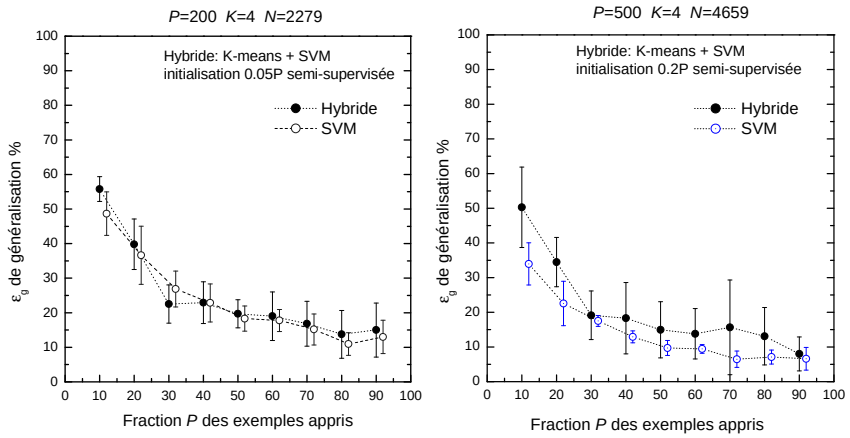


FIG. 3.7 – *Méthode hybride vs. MVS, $P = 200$ courriels à gauche et 500 courriels à droite. 100 tirages aléatoires. k=4 classes, N=dimension des matrices.*

3.9.2 Discusion

Nous avons étudié en détail les messages mal catégorisés afin de comprendre pourquoi le système les a mal classés. Une analyse *a posteriori* a montré que les messages trop courts, présentant de termes communs à plusieurs thématiques ou une combinaison de ces deux caractéristiques sont souvent mal catégorisés.

Nous présentons ci-dessous, deux exemples de messages qui n'ont pas été bien classés par notre système. Le premier étant trop court et trop vague pour permettre une catégorisation adéquate, tandis que le second appartient à la catégorie **Football** mais il comporte plusieurs termes (**mauvais_film**, **mauvais_scénario**, **action**, **la_fin**) de la catégorie **Cinéma**. Ceci illustre une partie des difficultés inhérentes à cette tâche, qui n'est pas évidente, même pour des êtres humains. Les sujets des courriels ont été retirés pour effectuer la classification, ceux-ci indiquant clairement la classe du courriel.

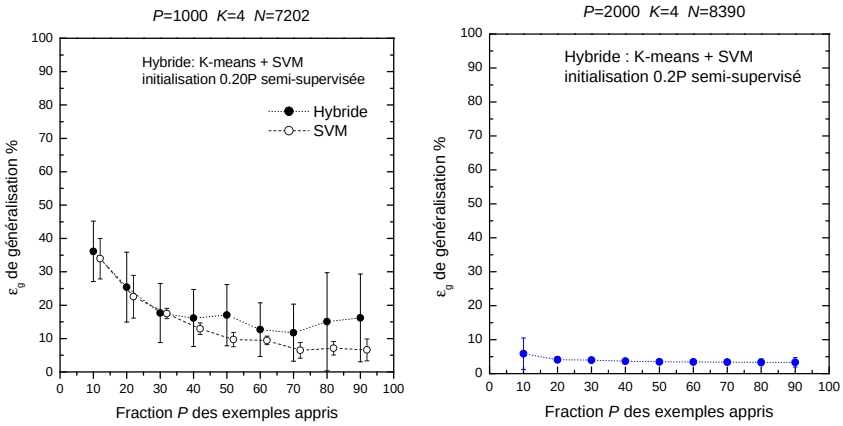


FIG. 3.8 – *Méthode hybride vs. MVS. À gauche $P = 1000$, à droite 2000 courriels. L'erreur est négligeable au délà de 20% des exemples appris. Par souci de clarté seulement la courbe de la méthode hybride est presentée, celle-ci se confondant avec la courbe SVM. k=4 classes, N=dimension des matrices.*

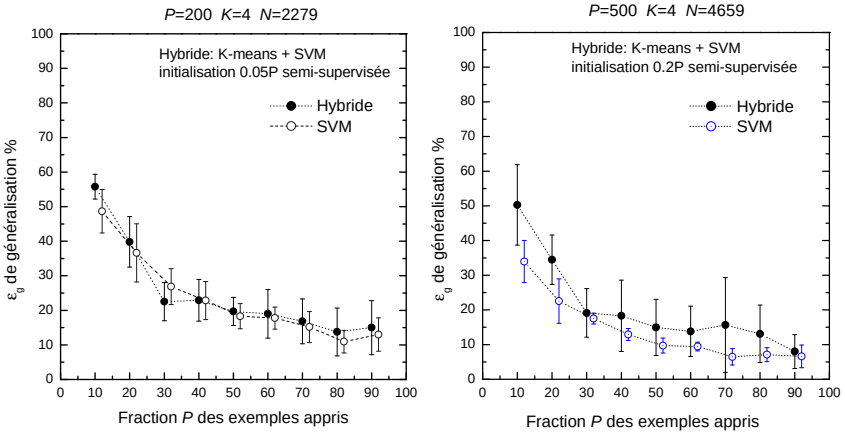


FIG. 3.9 – *Méthode hybride vs. MVS, $P = 200$ courriels à gauche et 500 courriels à droite. 100 tirages aléatoires. k=4 classes, N=dimension des matrices.*

Catégorie : **Jeux de rôles**, classé **Football**

```
From - Sun May 02 14:40:16 2004
Received: from [266.28.166.929] by n2.grp.scd.yahoo.com
To: <Shadowrun-france@yahoogroupes.fr>
In-Reply-To: <c70ti1+c4d8@eGroups.com>
From: Valerie <val@unicom.net>
Mailing-List: list <Shadowrun-france@yahoogroupes.fr>;
Date: Sun, 2 May 2004 05:24:15 -0700 (PDT)
Subject: [Shadowrun-france] Re: bonsoir_le_groupe
Reply-To: <Shadowrun-france@yahoogroupes.fr>

t'as raison... c'est pas brillant... au secours !!!!!!!!!!!!!!!
belle soirée Valérie
```

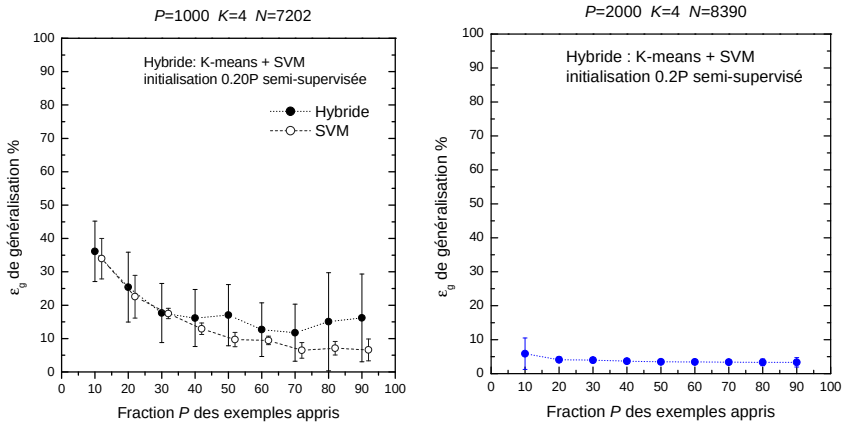


FIG. *3.10* – *Méthode hybride vs. MVS. À gauche $P = 1000$, à droite 2000 courriels. L'erreur est négligeable au delà de 20% des exemples appris. Par souci de clarté seulement la courbe de la méthode hybride est presentée, celle-ci se confondant avec la courbe MVS. k=4 classes, N=dimension des matrices.*

Catégorie : **Football**, classé **Cinéma**

```
From - Thu Mar 18 12:49:15 2004
Received: (qmail 64796 invoked from network); 18 Mar 2004 11:32:10 -0000
To: france-foot@yahoogroupes.fr
From: Jean LE COUTEAUX <jean@net.com>
Mailing-List: list france-foot@yahoogroupes.fr;
Date: Thu, 18 Mar 2004 12:33:14 +0100
Subject: [france-foot] CDF - 1/4
Reply-To: france-foot@yahoogroupes.fr

Nantes  (L1)    0-0     Rennes  (L1)
Ôh purée, j'y etais et une fois de plus j'ai été décu..
On aurait dit un mauvais film avec
de mauvais acteurs.. C'était mou, il n'y avait pas
d'action...Le scénario était couru d'avance, chacun en défense
et on attend la fin...
```

3.10 Conclusion

La tâche de classification de courriels est assez difficile en raison des particularités de cette forme de communication. Il s'agit d'une tâche où l'on travaille avec des évènements rares. Nous avons effectué des processus de pré-traitement (filtrage, traduction du phonécrit, lemmatisation) afin de représenter les courriels via un modèle vectoriel. Nos travaux concernant les dictionnaires avec ou sans accents ont permis d'accroître les performances du filtrage. À partir de cette représentation des données, nous avons réalisé une étude de différentes méthodes d'apprentissage automatique. Cette étude nous a permis de mieux connaître les caractéristiques de ces méthodes et leur comportement sur les données que nous avons collectées. Ainsi, l'apprentissage supervisé permet de mieux classer des nouveaux courriels mais demande une classification préalable qui

n'est pas toujours facile à mettre en œuvre. Par contre, bien que l'erreur d'apprentissage de *Fuzzy k-means* avec initialisation aléatoire semble importante, nous l'avons diminuée avec une initialisation semi-supervisée impliquant un faible nombre d'exemples. De plus, cette méthode n'a pas besoin de données étiquetées. La méthode hybride, qui permet de combiner les avantages de l'apprentissage non supervisé de *k-means* pour pré-étiqueter les données, et du supervisé avec MVS pour trouver les séparateurs optimaux, a donné des résultats intéressants : avec un 0,05% d'exemples appris (20 courriels), le taux d'erreur est inférieur à 20% dans le cas d'un ensemble d'apprentissage de 2000 courriels. Deux méthodes hybrides, découpage en mots et en n-grammes de caractères, ont été dévéloppées et testées. Les résultats avec les bi-grammes semblent être similaires à ceux avec des mots. Nous avons confirmé que la non lemmatisation produit des meilleurs résultats. Ceci va dans le même sens que les conclusions tirées par (Sahami, 1999). Des combinaisons des n-grammes, avec un lissage approprié, pourrait être aussi envisagées. Nous nous sommes principalement intéressés à l'amélioration de l'apprentissage non-supervisé, celui-ci ayant les résultats les plus bas au départ. Nos résultats montrent que la performance du système hybride est proche de celle des MVS. Ces travaux ont été publiés dans (Kessler et al., 2004b) et (Kessler et al., 2006).

Chapitre 4

E-Gen : Analyse automatique de courriels d'offre d'emploi

4.1 Introduction

La croissance exponentielle de l'Internet a permis le développement d'un grand nombre de *job boards* (Bizer et Rainer, 2005; Rafter et al., 2000a; Rafter et Smyth, 2000) et d'un marché du recrutement en ligne en pleine explosion. Beaucoup d'entreprises utilisent ce type de média de nos jours afin d'effectuer leurs recrutements. Le temps étant un facteur déterminant dans ce domaine, il était important de trouver un processus capable d'intégrer rapidement l'offre d'emploi de l'entreprise d'une manière automatique ou assistée et pouvoir la diffuser rapidement après validation par l'entreprise recruteuse. Le LIA et Aktor Interactive, agence de communication française spécialisée dans l'*e-recruiting*, ont développé le système E-Gen pour résoudre ce problème.

Le système E-Gen se compose de trois modules principaux :

1. Module d'extraction de l'information à partir de corpus des courriels provenant d'offres d'emploi extraites de la base de données d'Aktor.
2. Module pour analyser les réponses des candidats (afin de distinguer lettre de motivation et curriculum vitae).
3. Module pour analyser et calculer un classement de pertinence du profil des candidats (lettre de motivation et curriculum vitae).

Afin d'extraire l'information utile, le premier module analyse le contenu des courriels d'offres d'emploi. Cette étape présente des problèmes intéressants liés au TAL : les textes des offres sont écrits dans un format libre, sans structure, avec certaines ambiguïtés et des erreurs typographiques. Une des principales activités de l'entreprise est la publication d'offres d'emploi sur les sites d'emploi en ligne pour les sociétés ayant un besoin en recrutement. Face à la grande quantité d'informations disponibles sur In-

ternet et au nombre important de *job boards* (spécialisés[1], non spécialisés[2] ou locaux[3]), Aktor a besoin d'un système capable de traiter rapidement et efficacement ces offres d'emploi avant leur diffusion. Pour cela, Aktor utilise un système automatique pour envoyer les offres au format XML (*Robopost Gateway*). Au cours de cette première étape, il est donc nécessaire d'identifier les différentes parties de l'offre d'emploi et de plus d'extraire certaines informations pertinentes (type de contrat[4], salaire, localisation, etc.). Auparavant, cette première étape était une tâche manuelle : on demandait aux utilisateurs de copier et coller les offres d'emploi dans le système d'information de l'entreprise. Nous nous intéresserons principalement dans ce chapitre au premier module du système E-Gen et ses performances sur les tâches d'extraction ainsi que de classification. Nous aborderons l'architecture globale du premier module du système E-Gen avant de détailler la modélisation adoptée pour notre problème. Puis nous présentons les différents algorithmes de classification mis en place et les résultats.

4.2 Architecture

Nous avons choisi de développer un système répondant aussi rapidement et judicieusement que possible au besoin de la société Aktor, et donc aux contraintes du marché de recrutement en ligne. Dans ce but, une adresse électronique a été créée afin de recevoir les courriels (parfois avec un fichier attaché) contenant les offres d'emploi et une librairie afin de les traiter, celle-ci est présentée en annexe B. Après l'identification de la langue, E-Gen analyse le message afin d'extraire le texte de l'offre d'emploi du message ou du fichier attaché. Un module externe, wvWare traite le document MS-Word et produit une version texte du document decoupé en segments[5]. La segmentation de textes MS-Word étant un vrai casse-tête, nous avons opté pour un outil existant. Dans la majorité des cas, il sectionne en paragraphes ou en phrases le document. Nous ramenons de ce fait la problématique de segmentation à un découpage préfixées. Après l'étape de filtrage et lemmatisation, nous utilisons la représentation vectorielle pour chaque segment afin de lui attribuer une étiquette en fonction de son rôle dans le texte. Par la suite, cette séquence d'étiquettes, qui donne une représentation de l'enchaînement des différentes parties du texte de l'annonce, est traitée par un processus correctif qui la valide ou qui propose une meilleure séquence. À la fin du traitement, un fichier XML est généré et envoyé au système d'information d'Aktor. Lors de la publication d'une offre d'emploi, un certain nombre d'informations est requis par le *job board*. Ainsi, il faut trouver ces champs dans l'annonce afin de les incorporer dans notre fichier XML. Nous avons donc mis en place différentes solutions à base de règles écrites à la main afin de localiser des informations telles que salaires, lieu de travail, noms d'entreprises, contrat, référence, durée de la mission (voir annexe A). La chaîne de traitement du premier module d'E-Gen est représentée dans la figure 4.1.

[1] `http://www.admincompta.fr` (comptabilité), `http://www.lesjeudis.com` (informatique)
[2] `http://www.monster.fr`, `http://www.cadremploi.fr`, `http://www.cadresonline.com`
[3] `http://www.emploiregions.com`, `http://www.regionsjob.com`
[4] CDI, CNE, CDD, stage, intérim, etc.
[5] `http://wvware.sourceforge.net.`

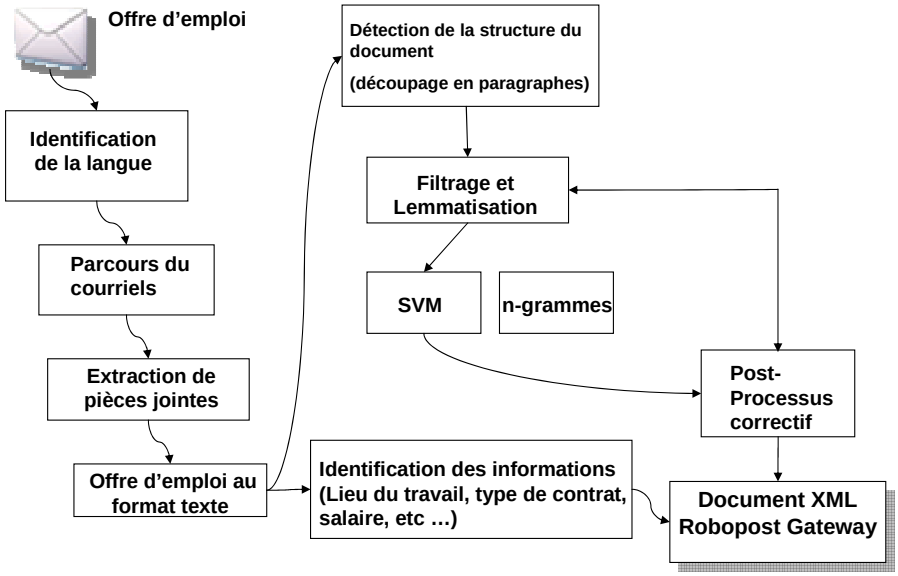


FIG. 4.1 – *Architecture E-Gen Module 1.*

4.3 Analyse d'une offre d'emploi

L'extraction à partir de la base d'informations d'Aktor a permis d'avoir un corpus de taille relativement importante, sans catégorisation manuelle, nommé par la suite *Corpus d'Offres d'Emploi*. Quelques statistiques du corpus sont rapportées dans le tableau 4.1 et des exemples d'offres d'emploi sont présentés en annexe D. Notre analyse a montré que les offres d'emploi se composent souvent de blocs d'information thématiquement proche qui demeurent, cependant, fortement non structurés. Une offre d'emploi est composée de quatre blocs :

1. **DESCRIPTION** : bref résumé de l'entreprise qui recrute ;
2. **TITRE** : titre de l'emploi ;
3. **MISSION** : courte description de l'emploi ;
4. **PROFIL** : qualifications et connaissances exigées pour le poste. Les contacts sont généralement inclus dans cette partie.

Nombre d'offres d'emploi	$D = 1000$	
Nombre total de segments	$P = 15621$	
Nombre de segments étiquetés **Titre**	1 000	6,34 %
Nombre de segments étiquetés **Description**	3 966	25,38 %
Nombre de segments étiquetés **Mission**	4 401	28,17%
Nombre de segments étiquetés **Profil**	6 263	40,09 %

TAB. 4.1 – *Statistiques du Corpus d'Offres d'Emploi.*

On peut en déduire que dans une offre il peut y avoir plusieurs segments de même type. Ceux-ci pouvant être consécutif ou non.

4.4 Une approche par classification thématique à l'aide de dictionnaire spécifique

L'observation de chaque annonce a révélé l'utilisation d'un vocabulaire spécifique, comme le montre le tableau 4.2 qui regroupe un certain nombre d'exemples de termes propres à chaque partie de l'annonce. Chacune des annonces étant en format libre, les termes ne sont pas répétés pour chaque annonce, cependant la taille de notre corpus a permis d'en recenser un nombre important.

Partie de l'annonce	Exemples de termes spécifiques
Titre	homme/femme, infographiste, développeur, commercial, consultant, chef_de_projet
Description	société, établissement, nous, entreprise, recruter, filiale, liste_clients[5]
Mission	vous_aurez_pour_mission, mise_en_place, organiser, responsabilité, principale_mission, participer, gérer
Profil	compétence, bac+X, disponible, mobile, autonome, expérience, motivation, autonome, profil_recherché, postuler

TAB. 4.2 – *Exemples de termes collectés en fonction de chaque partie d'une annonce.*

Nous avons donc tenté une première approche basée sur une collecte manuelle de mots contenus dans le *Corpus d'Offres d'Emploi* spécifique à chaque partie de l'annonce. Nous avons par la suite effectué une pondération empirique de chacun des termes, certains étant extrêmement discriminants pour une catégorie (*recruter* $\in$ **Description**, *homme/femme* $\in$ **Titre**, *vous_aurez_pour_mission* $\in$ **Mission**, *postuler* $\in$ **Profil**) tandis que d'autres peuvent être présents dans plusieurs parties de l'annonce (*vous* $\in$ **Mission** $\cup$ **Profil** par opposition au *nous* que l'on trouve uniquement dans la partie **Description** ; *français* $\in$ **Description** $\cup$ **Profil**, par exemple *Groupe français recherche* et *Langues : français/anglais*. Nous effectuons par la suite pour chaque segment la somme des pondérations en fonction de la présence/absence de chacun des termes et le segment sera associé à la catégorie ayant obtenue le score le plus haut. L'analyse des premiers résultats obtenus montrent une classification parfois pertinente, mais relativement faible sur une série de tests (environ 55% des segments correctement classés) compte tenu de la faible taille de nos dictionnaires.

[5]Un dictionnaire listant les différentes sociétés clientes de la société Aktor a été crée

4.5 Classification par MVS et *n*-grammes de mots

Nous avons choisi les MVS pour cette tâche suite aux résultats lors de travaux précédents sur la classification de courriels (voir section 3.6). Nous avons cependant utilisé l'implémentation LibSVM (Fan et al., 2005), plus appropriée à cette tâche (SVMTorch ne permettant pas de faire de prédiction sur des données non préalablement étiquetées, ce qui était nécessaire pour la version finale sur le serveur). Les résultats obtenus montrent une classification performante des segments individuellement pour chacun des classifieurs (voir Figure 4.2), mais les premiers tests d'intégration d'une offre d'emploi complète ont montré une chute importante des performances due au grand nombre d'annonces avec un ou deux segments mal classés. La figure 4.2 présente le pourcentage de segments en erreur pour chaque classifieur sur une série de 50 tests avec tirage aléatoire sur le *Corpus d'Offres d'Emploi*. On observe que malgré de bonnes performances globalement ($\approx$ 20% en moyenne), le classifieur à base de bi-grammes reste inférieur aux MVS ($\approx$ 13% en moyenne) sur l'ensemble de tests.

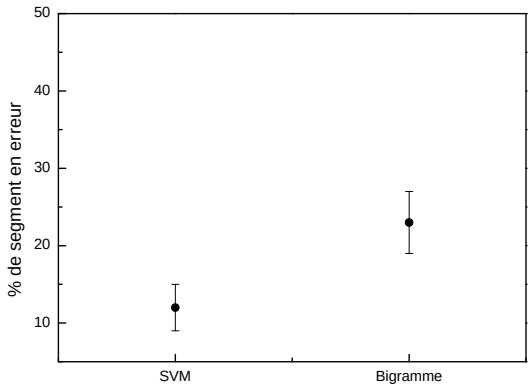


FIG. 4.2 – *Segments en erreurs pour les MVS et les bi-grammes de mots.*

La figure 4.3 résume les différents résultats obtenus en fonction de différents noyaux des MVS en fonction du pourcentage de segments reconnus. La figure ne présente que les meilleurs résultats obtenus pour chaque noyau, un certain nombre de tests ayant été effectués en jouant sur les différents paramètres des MVS. Le corpus utilisé a été une sélection de 4476 segments répartis proportionnellement dans chacune des classes (25%). On observe que les noyaux polynomial, sigmoïde et radiale obtiennent toujours des résultats inférieurs. Au vu de ces résultats préliminaires, nous avons décidé de réaliser un modèle SVM, utilisant un noyau linéaire.

4.6 Modélisation

La catégorisation de segments sans considérer leur position dans l'offre d'emploi peut être une source d'erreurs. En effet, nous avons constaté que les MVS produisent globalement une bonne classification des segments individuels, mais les segments d'une

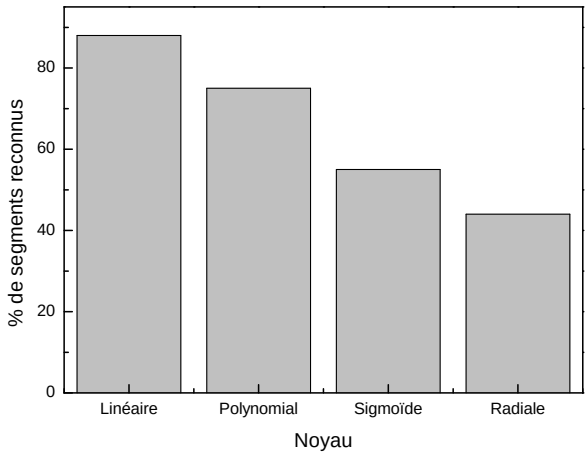


FIG. 4.3 – *Pourcentage de segments reconnus en fonction de chaque noyau MVS.*

même offre d'emploi sont rarement tous correctement étiquetés. En raison d'une grande variété dans les paramètres (texte libre, découpage incertain, délimiteur varié), il s'est avéré difficile de traiter ce type de documents avec des expressions régulières. Une des erreurs la plus fréquente et la plus visible rencontrée lors de ces tests était une mauvaise catégorisation des segments **Titre**. Ce segment est en général très court, mais contient des informations importantes sur l'offre d'emploi comme le montre l'exemple 4.3.

```
<segment class="description" >Fort d'une expérience de plus de dix
ans dans le domaine du bio-nettoyage et des services hôteliers en
milieu de santé. Nous nous positionnons en leader sur ce marché en
pleine expansion.</segment>
<segment class="title" >coordinateur travaux</segment>
<segment class="mission" > Sous la responsabilité du Responsable du
département hygiène technique vous aurez comme mission la gestion
des plannings d'intervention, des approvisionnement et le suivi de la
facturation et de l'administration du personnel. </segment>
<segment class="profil">Agé(e)s de 30-35 ans vous bénéficiez
d'une expérience réussie de 3 à 5 ans dans le bio-nettoyage et la
décontamination des réseaux aérauliques ou dans le BTP. </segment>
<segment class="title">Poste basé a Toulon </segment>
```

TAB. 4.3 – *Exemple d'offre d'emploi.*

Le dernier segment est catégorisé **Titre**, mais cela est impossible. Afin de corriger ce problème, nous avons calculé dans un premier temps la probabilité $P(titre|position)$ que le segment soit un titre en fonction de sa position x dans l'annonce tel que défini en 4.1, suivant le principe d'une distribution selon une loi exponentielle comme le montre l'estimation obtenu en figure 4.4.

$$P(title|position) = y_0 + Ae^{(-\frac{1}{2}(\frac{x-x_c}{w})^2)} \tag{4.1}$$

où $y_0 = -0,003; x_c = -0,889; w = 5,026$ *et* $A = 3,468$.

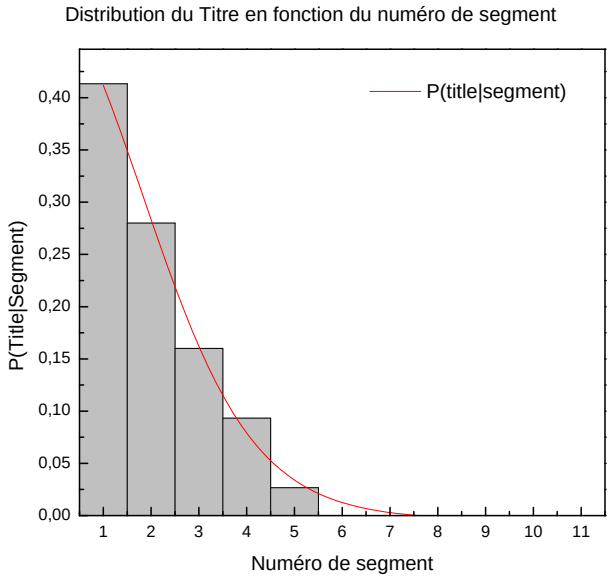


FIG. 4.4 – *$P(title|segment_i)$ distribution selon une loi exponentielle de la position du* ***Titre*** *en fonction du numéro i du segment.*

Le modèle semblait augmenter les performances du système sur un petit échantillon d'exemples, mais sans réelles amélioration sur une série de test complète. L'estimation présentée avec la figure 4.4 montre cependant qu'un segment **Titre** à une probabilité nulle d'être au-delà du cinquième segment. Ces observations nous ont fait opter pour un automate de Markov, permettant ainsi de gérer la position de chaque élément de l'annonce vis à vis des autres. Le modèle proposé a six états différents : **Début (S)**, **Titre (1)**, **Description (2)**, **Mission (3)**, **Profil (4)** et **Fin (E)**. Nous avons donc représenté une offre d'emploi comme une succession d'états dans cette machine. Chaque état ayant la possibilité d'émettre un segment ou de passer à l'état suivant en fonction d'une certaine probabilité. Nous avons donc parcouru l'ensemble du corpus de référence afin de déterminer les probabilités de transition entre les états. Le tableau 4.4 montre la matrice de probabilités obtenue.

L'observation de cette matrice nous renseigne sur la structure d'une offre d'emploi. Ainsi, celle-ci a une probabilité $p = 0,99$ de commencer par le segment **Description** mais il est impossible de commencer par **Mission** ou **Profil**. De la même manière, un segment **Mission** peut seulement être suivi soit d'un segment **Mission** soit d'un segment **Profil**. Ceci nous a permis d'en déduire l'automate représenté sur la figure 4.5. Celui-ci possède 4 états différents en plus de l'état initial et final. Chaque état pouvant boucler sur lui même ou passer à l'état suivant en fonction d'une certaine probabilité

	Début	Titre	Description	Mission	Profil	Fin
Début	0	0,01	0,99	0	0	0
Titre	0	0,05	0,02	0,93	0	0
Description	0	0,35	0,64	0,01	0	0
Mission	0	0	0	0,76	0,24	0
Profil	0	0	0	0	0,82	0,18
Fin	0	0	0	0	0	0

TAB. 4.4 – *Matrice de Markov.*

déterminée par la matrice 4.4. Le processus correctif décrit en 4.6.1 est piloté par cet automate.

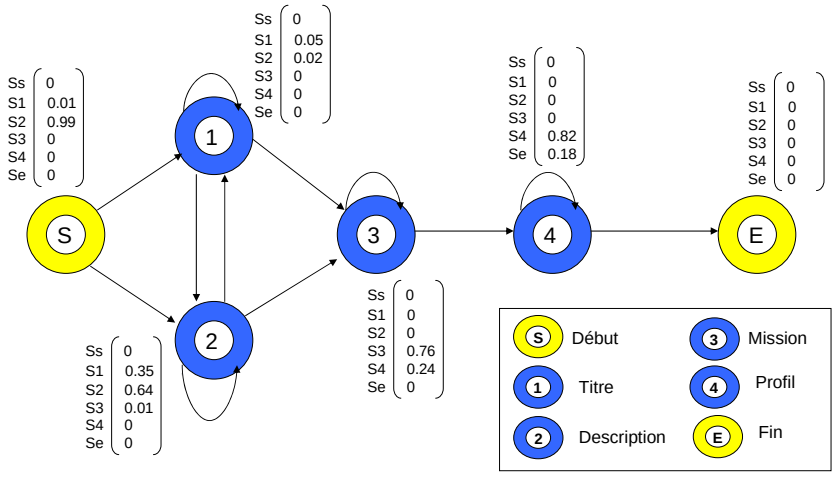


FIG. 4.5 – *Automate de Markov utilisée pour corriger les étiquettes des segments mal classés.*

4.6.1 Processus correctif

Lors de la classification d'une offre d'emploi complète, quelques segments sont classés incorrectement, sans un comportement régulier (un segment **Description** a été détecté au milieu d'un **Profil**, le dernier segment de l'offre d'emploi a été identifié comme **Titre**, etc.). Afin d'éviter ce genre d'erreurs, nous avons appliqué un post-traitement inspiré de l'algorithme de Viterbi (Manning et Schütze, 1999; Viterbi, 1967). La classification par MVS donne à chaque segment individuellement une classe. Une offre complète est une succession de segment de texte. Chaque segment pouvant être répété (dans le cas ou la segmentation du document a été mauvaise). Par exemple, une annonce qui donnerait le découpage en segment suivant :

Description↦Description↦Titre↦Mission↦Mission↦Profil

donnerait la séquence suivante :

S↦2↦2↦1↦3↦3↦4↦E

L'algorithme classique de Viterbi calculera la probabilité de la séquence. Si la séquence est probable, l'automate renvoie cette probabilité et la séquence est proposée comme découpage de l'offre. Si la séquence est improbable, Viterbi renvoie 0, la séquence est rejeté et le processus correctif est interrogé afin de transmettre la séquence avec un nombre d'erreurs minimales (comparé à la séquence produite par les MVS) et une probabilité maximale. Ce processus parcourt l'arbre des solutions possibles en calculant la probabilité de la séquence ainsi que le nombre de différence par rapport à la séquence produite par les MVS.

Algorithme (1) présente le détail de cette procédure. Le schéma 4.6 présente un exemple de déroulement du post-processus correctif avec la sous-séquence hypothétique suivante 3↦4↦3↦3↦4↦5 (d'après l'automate 4.5, il est impossible de passer de l'état 4 (**Profil**) à l'état 3 (**Mission**)). Le post-processus propose donc différentes solutions : Les 3 premières ayant 2 différences par rapport à la séquence MVS, la dernière n'aboutissant pas à l'état final (5), la séquence retenue sera la 4ème 3↦3↦3↦3↦4↦5, celle-ci ayant une seule erreur et la probabilité la plus importante.

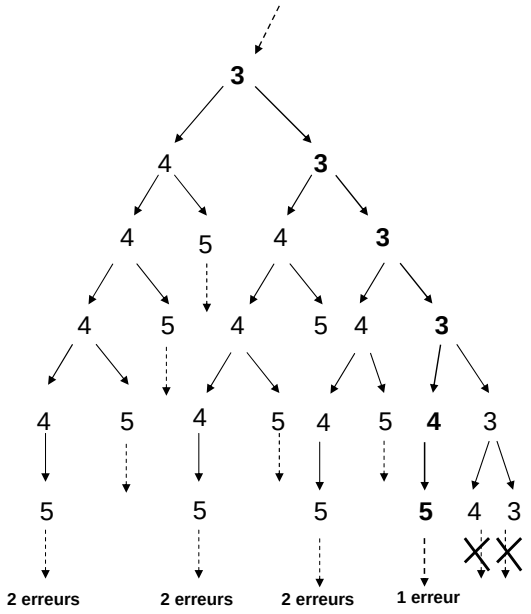


FIG. 4.6 – *Déroulement du post-processus correctif.*

Les premiers résultats étaient intéressants, mais avec des temps de traitement assez grands lorsque le nombre de segments était important (plusieurs heures de calcul pour

Calcul symbole suivant()
Calcul de la séquence en cours (Viterbi) : ajout du nouveau symbole, calcul de la probabilité de la séquence et du nombre d'erreurs.
if *Nombre d'erreurs de la séquence en cours > Maximum d'erreurs trouvé* **then**
  return `séquence en cours`
end
if *le symbole courant est le dernier de la séquence* **then**
  if *nombre d'erreurs de la séquence en cours < Maximum d'erreurs trouvé* **then**
    `Maximum d'erreurs trouvé = Nombre d'erreurs de la séquence en cours;`
  end
  return `séquence en cours;`
end
else
  foreach *symbole de la séquence* **do**
    Séquence en cours = **Calcul symbole suivant()**
    if *la séquence en cours est la meilleure séquence* **then**
      `Meilleure séquence = séquence en cours;`
      if *nombre d'erreurs de la séquence en cours < Maximum d'erreurs trouvé* **then**
        `Maximum d'erreurs trouvé = Nombre d'erreurs de la séquence en cours;`
      end
    end
  end
end

Algorithm 1: Algorithme du post-processus correctif avec la méthode Branch and Bound.

une annonce d'une cinquantaine de segments). Nous avons introduit une amélioration en utilisant un algorithme *Branch and Bound* (Land et Doig, 1960) pour élaguer l'arbre : dès qu'une première solution est trouvée, le nombre de différence par rapport à la séquence MVS et sa probabilité sont retenues et comparées chaque fois qu'une nouvelle séquence est traitée. Si la solution n'est pas meilleure (au sens nombre de segments différents et probabilité inférieur), le reste de la séquence n'est pas calculée. L'utilisation de cet algorithme permet d'obtenir une solution avec des temps très acceptable (le traitement de séquences contenant 50 symboles avoisine les 2 secondes) mais présente cependant quelques lacunes. Le post processus s'appuie sur la séquence transmise par le classifieur. Si celle-ci contient trop d'erreurs ou des erreurs autorisées par l'automate, celui-ci n'apporte aucune amélioration.

Nous avons effectué une évaluation selon deux niveaux de finesse : par segment et par offre d'emploi entièrement reconnue. La figure 4.7 à gauche montre une comparaison entre les résultats obtenus par les MVS et le processus correctif. Les courbes présentent le nombre de segments non reconnus en fonction de la taille du corpus d'apprentissage. La figure présente les résultats des MVS seules (ligne pointillée) sur la tâche de classification des segments. Les résultats sont bons et prouvent que même avec une petite fraction de patrons d'apprentissage (20% du total), le classifieur MVS obtient un faible taux de patrons mal classés (< 10% d'erreurs).

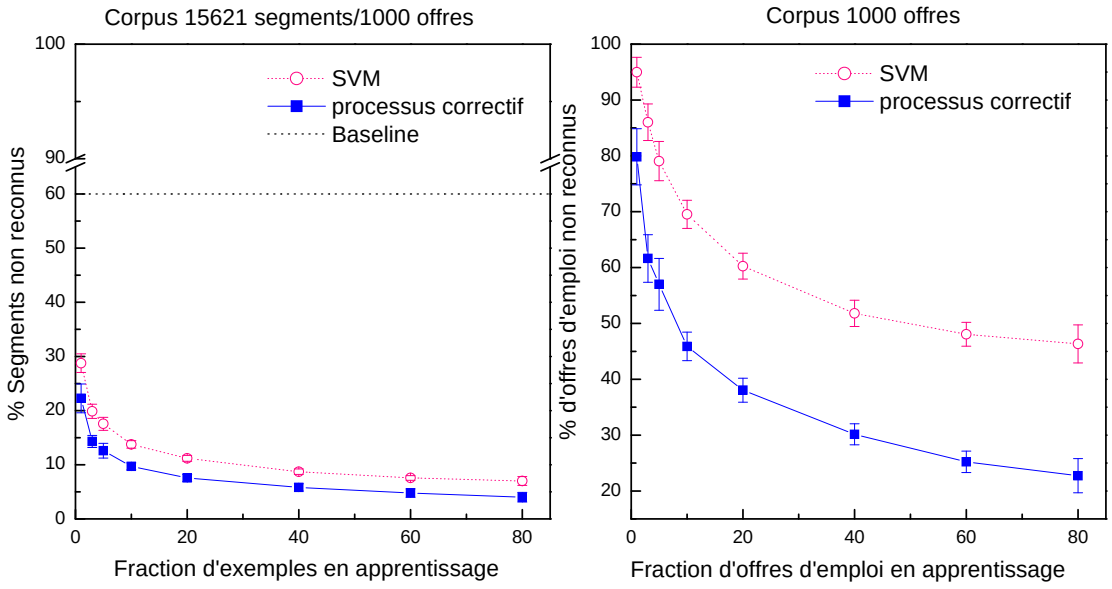


FIG. 4.7 – *À gauche, les taux d'erreurs en pourcentage des MVS et de l'algorithme correctif par rapport aux nombre de segments mal étiquetés. À droite, les taux d'erreurs en pourcentage des MVS et de l'algorithme correctif par rapport aux offres d'emploi reconnues de façon erronée.*

Le processus correctif (ligne continue) donne toujours de meilleurs résultats que les MVS quelle que soit la fraction d'exemples d'apprentissage. Pour comparaison, une classification *Baseline* avec la classe la plus probable (étiquette **Profil** avec environ 40% d'apparition sur le corpus) donne 60% d'erreurs calculée sur tous les segments. La

figure 4.7 à droite présente une comparaison entre les résultats obtenus par chaque méthode mais selon les offres d'emploi mal étiquetées. On observe une considérable amélioration du nombre d'offres d'emploi identifiées avec le processus correctif. MVS obtient un minimum d'environ 50% des offres d'emploi mal étiquetées, et le processus correctif en obtient 25%, donc une amélioration de plus du 50% du score des MVS.

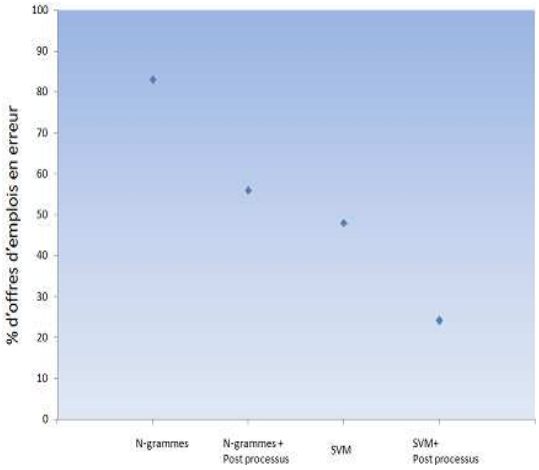


FIG. 4.8 – *Comparaison des résultats obtenus avec un apprentissage fixé à 80%pour les différentes méthodes de classification avec ou sans processus correctif.*

La figure 4.8 montre une comparaison entre les résultats obtenus pour chaque méthode, avec ou sans processus correctif, selon les offres d'emploi partiel ou totalement mal reconnues. On observe que le processus correctif améliore les résultats quel que soit l'algorithme de classification (amélioration d'environ 30% pour les bi-grammes et d'environ 20% pour les MVS). L'ensemble des tests montre également que la classification par MVS obtient des résultats de meilleure qualité que la classification par bi-grammes.

4.6.2 Discussion

Une analyse des offres d'emploi mal étiquetées, montre qu'environ 10% d'entre elles contiennent une ou deux erreurs. Ces segments mal classés, correspondent généralement au bloc frontière entre deux catégories différentes (El-Bèze et al., 2007) tel que montré dans la figure 4.9.

FIG. 4.9 – *Erreurs de blocs frontière.*

Ainsi, le classifieur donne comme séquence 0↦2↦1↦3↦3↦3↦4↦4↦5 alors que la séquence correcte est 0↦2↦1↦3↦3↦4↦4↦4↦5. Le segment dont l'étiquetage est faux, est reproduit dans la figure 4.10.

*De formation superieure Ecole d'ingenieur Chimiste (CPE) option chimie analytique environnementale, vous avez deja conduit un **projet** de **transfert d'activite**.*

FIG. *4.10* – *Exemple de segment mal classé.*

Une analyse des exemples avec des erreurs aux blocs frontières montre que des termes importants présents dans deux catégories différentes amènent à une classification incorrecte. En particulier, des termes tels que **projet** et **transfert d'activité** correspondent aux catégories **Mission** et **Profil**. Le segment est classé en tant que **Profil**. En fait, ce segment se trouve à la frontière entre les blocs **Mission** et **Profil**, la séquence étant probable (la probabilité de Viterbi n'est pas nulle), cette erreur n'est pas corrigée par le processus correctif.

4.7 Conclusion

Le traitement des offres d'emploi est une tâche difficile car l'information y est toujours fortement non structurée. Ce chapitre a mis en avant le module de catégorisation, premier composant d'E-Gen, système pour le traitement automatiquement des offres d'emploi. Les premiers résultats obtenus par les MVS étaient très intéressants environ 10% de segments mal étiquetés pour un corpus d'apprentissage de 80%. Le processus correctif améliore ces résultats d'environ 50% pour chaque méthode de classification (MVS et bi-gramme) et diminue considérablement les erreurs de segments isolés incorrectement classés, tout en restant dans des temps de calcul très raisonnables. L'extraction des informations telles que le salaire (salaire minimum, maximum et devise), le lieu de travail et la catégorisation de l'emploi sont correctement détectés et permettent une meilleure caractérisation des offres sur les sites d'emploi, critères importants pour l'intégration dans le système d'information de l'entreprise. Ce module d'E-Gen est actuellement sur le serveur d'Aktor et permet un gain de temps considérable dans le traitement des offres d'emploi avec un coût minimal en terme d'intervention humaine. Ces travaux ont été publiés dans (Kessler et al., 2004b) et (Kessler et al., 2006).

Chapitre 5

E-Gen : Évaluation des réponses à une offre d'emploi

5.1 Introduction

Internet est devenu essentiel dans le processus de recrutement, car il permet une meilleure diffusion de l'information, que ce soit par les sites de recherche d'emplois ou par les réponses à ceux-ci par courriels. Cependant, ce phénomène pose divers problèmes dans leur traitement dû à des volumes d'information (majoritairement sous la forme de texte libre) intraitables manuellement pour les entreprises. En outre, si le navigateur Web est devenu un outil universel, facile à employer pour les utilisateurs, la nécessité fréquente pour les internautes d'entrer des données dans les formulaires Web à partir de sources papier, de « copier et coller » de données entre différentes applications, est symptomatique des problèmes d'intégration de données communes. Par ailleurs, le flux de réponses à une offre d'emploi entraîne un long travail de lecture des candidatures par les recruteurs. En conséquence, il est nécessaire de traiter cette masse de documents d'une manière automatique ou assistée. Nous souhaitons pour cela mettre en place un système capable de fournir une première évaluation automatisée des candidatures selon divers critères. Nous présentons dans ce chapitre les travaux concernant les problématiques de séparation de CV/LM ainsi que de classement de candidatures.

5.2 Vue d'ensemble

Lors de la publication d'une offre d'emploi, la société Aktor génère une adresse électronique afin de répondre à cette offre. Chaque courriel est ainsi redirigé vers un logiciel de ressources humaines, Gestmax[1] afin d'être lu par un consultant en recrutement. Lors de la réception d'une candidature, le système extrait le corps du message, ainsi que les différentes pièces jointes. Une version texte des différents documents contenus

[1] http ://www.gestmax.fr

dans la candidature est alors produite. Différents processus de filtrage et racinisation permettent au système d'identifier à l'aide de MVS et de règles le type du document (CV et/ou LM présents dans le corps du courriel ou dans les pièces jointes (module 2)). Une fois le CV et la LM identifiés, le système effectue un profilage automatisé de cette candidature à l'aide de mesures de similarité en s'appuyant sur un petit nombre de candidatures préalablement validées comme candidatures pertinentes par un consultant en recrutement (module 3). La figure 5.1 présente une vue d'ensemble des deux modules.

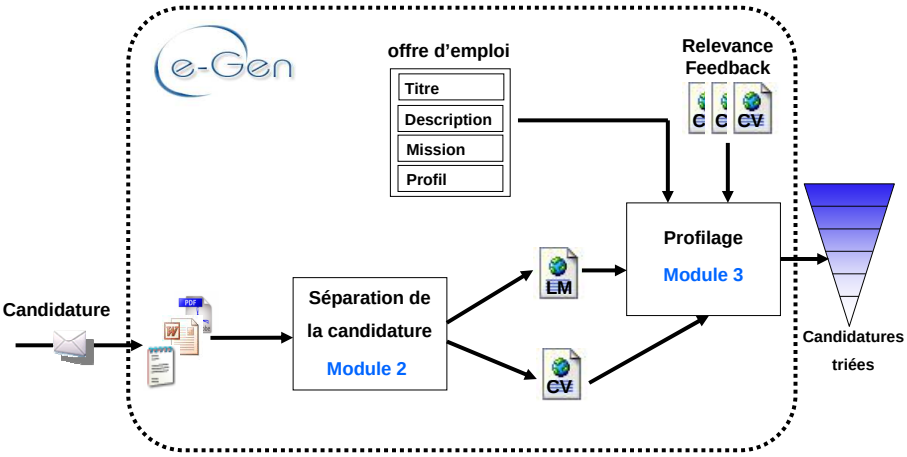


FIG. 5.1 – *Vue d'ensemble du module 2 et 3 du système E-Gen.*

5.3 Corpus et analyse de candidatures

Le corpus regroupe plusieurs missions[2] d'Aktor Sourcing & Selection[3] ainsi que les diverses réponses à ces offres d'emplois étiquetées par un recruteur du cabinet. Il regroupe un ensemble d'offres d'emploi avec des thématiques différentes (emplois en comptabilité, entreprise, informatique, etc.). Afin de simplifier le problème nous l'avons ramené à deux étiquettes avec les valeurs **retenues** ou **non retenues**. Une valeur **retenue** correspond à un candidat potentiellement intéressant pour un emploi donné et une valeur **non retenue** a été attribuée à une candidature non pertinente, selon l'avis d'un consultant en recrutement. Ce regroupement nous a permis d'équilibrer un peu

[2]Mission désigne la pré-sélection effectuée par le cabinet de recrutement pour une offre d'emploi.
[3]http ://www.aktor-selection.fr

le corpus, celui-ci étant majoritairement composé de candidatures étiquetées **non retenue** comme le montre le tableau 5.1. Les missions peuvent être rédigées en différentes langues, mais nous conserverons pour notre étude les offres et les réponses en français (le marché français représente l'activité principale d'Aktor). Ce sous-ensemble, nommé *Corpus Mission* a donc permis d'obtenir un corpus de réponses classées en fonction de la mission ainsi que du jugement d'un recruteur sur les candidatures.

Nombre total de missions	25
Nombre de missions avec moins de 10 réponses	2
Nombre de missions avec plus de 10 réponses	8
Nombre de missions avec plus de 50 réponses	6
Nombre de missions avec plus de 100 réponses	9
Nombre total de candidatures	1586
Nombre de candidatures **retenue**	160
Nombre de candidatures **non retenue**	1426

TAB. *5.1* – *Statistiques du Corpus Mission.*

Nous avons effectué une analyse fine de chacune des missions afin de connaître le nombre de candidatures par mission, la longueur en mots de l'offre d'emploi ainsi que la taille moyenne (LM et CV) des candidatures, comme le montre le tableau 5.2. On observe que certaines missions ont un nombre de candidatures vraiment faible (par exemples les missions 00010, 32559 et 32560) et d'autres pour lesquelles l'annonce est relativement courte (31702 : 63 mots, 33553 : 81 mots, et 31702 : 84 mots). La longueur moyenne des CV étant d'environ 200 mots contre 60 pour les LM. L'annexe C présente les intitulés des offres d'emploi.

Nous avons effectué un pré-traitement des CV et LM permettant leur anonymisation (noms des candidats, suppression des adresses, des courriers électroniques et des noms de villes). La suppression des accents est également effectuée. Nous avons, par la suite, utilisé les processus de filtrage et racinisation présentés en 2.3.2 afin de réduire le bruit dans le modèle à base de mots[4].

5.4 Séparation de CV/Lettre de motivation par MVS

Nous avons tenté une classification simpliste en se basant uniquement sur les noms des fichiers. Cependant ceci s'est avéré insuffisant[5] en raison de la diversité des noms de fichiers[6]. Nous avons 2 classifieurs naïfs basés sur les longueurs moyennes des phrases et sur le nombre de mots dans chaque type de documents. On notera la faible différence au niveau de la longueur moyenne des phrases, les LM étant généralement des

[4]Ces pré-traitements ne sont pas appliqués lors de la représentation en n-grammes de caractères décrites en 5.8.

[5]Le système constituait un corpus tronqué à 1725 CV et 910 LM.

[6]Par exemple : PierreDurand.doc, Durand.pdf, Aktor.doc, 13042007.doc, V3.doc, etc.

Mission	Nb candidatures	Longueur offre	Taille moyenne CV	Taille Moyenne LM
32559	3	123	347,5	70,25
00010	6	184	241,48	108,14
32560	6	135	390,38	65,63
32854	9	128	211	50,13
34692	11	187	272,88	72,31
33867	12	127	115,39	73
00014	18	85	153,63	59,93
34990	19	91	187,16	48,06
35044	19	151	238,81	65,44
34861	29	147	194,9	81,23
31273	37	164	184,12	69,37
31248	41	183	221,33	70,80
33633	42	173	232,63	74,48
34865	44	72	163,48	50,72
31702	45	63	152,56	59,78
32461	49	84	264,05	69,70
00015	54	112	178,02	48,47
33746	60	123	157,83	36,86
34783	61	91	148,44	32,44
33725	63	154	137,74	46,72
33553	74	81	134,06	44,28
31022	132	117	193,70	59,78
31274	200	111	143,46	62,67
34119	206	100	177,25	69,30
31767	346	111	144,33	69,27
Moyenne	63,44	123,88	199,45	62,35

TAB. 5.2 – *Statistiques des candidatures par mission, triées par nombre de candidatures.*

documents courts et les CV des documents de synthèse présentant peu de caractères délimitant les phrases ("'."', "' :"', etc.).

	CV	LM
Nombre de documents	2165	2165
Nombre total de phrases	45655	20658
Longueur moyenne des phrases	17,07	18,97
Nombre total de mots	922103	412008
Moyenne de mots par document	425,91	190,30

TAB. 5.3 – *Statitistiques à la base des classifieurs naïfs.*

Nous avons ainsi construit un premier classifieur naïf *N1* qui décide en fonction du nombre de phrases ($p < 18$ =>CV, $p > 18$ =>LM) et un second *N2* en fonction du nombre de mots ($m < 200$=>LM, $m > 200$ =>CV). Cependant les résultats (tab. 5.4), montrent les limites de ce genre de méthode.

Le tableau 5.4 présente les résultats obtenus avec chacun des classifieurs naïfs, *N1* et *N2*. Le classifieur *N1* sépare l'ensemble des documents comme des CV et aucun dans la classe LM. Les résultats obtenus par le classifieur *N2* sont plus mitigés mais res-

Classifieur	Précision	Rappel	Fscore
N1	0,66	1	0,75
N2	0,35	0,26	0,30

TAB. 5.4 – *Précision, Rappel, Fscore obtenus par les deux classifieurs naïfs.*

tent décevants. Nous expliquons cela par l'hétérogénéité des données (des CV parfois très courts et des LM parfois extrêmement longues) qui viennent fausser les moyennes. Nous avons donc choisi les MVS, présentées en section 3.6 pour cette tâche compte-tenu des bons résultats obtenus dans les travaux précédents en catégorisation de texte. Afin de régler les paramètres et tester nos méthodes, nous avons scindé le *Corpus Mission* en cinq sous-ensembles approximativement de la même taille $A_i\,; i = 1, \cdots, 5$. avec une répartition aléatoire mais équilibrée des candidatures dans chaque sous-corpus. Le protocole expérimental a été le suivant : nous avons concaténé quatre des cinq sous-ensembles comme ensemble d'apprentissage et gardé le cinquième pour le test (par exemple, les sous-ensembles d'apprentissage A_1, A_3, A_4 et A_5 valident le sous-ensemble de test A_2). Cinq expériences ont été ainsi effectuées à tour de rôle. Nous avons choisi d'effectuer ce découpage afin d'éviter de régler les algorithmes sur un seul ensemble d'apprentissage (et un seul ensemble de test), ce qui pourrait conduire à deux travers, le biais expérimental et/ou le phénomène de sur-apprentissage (Torres et al., 2007). Les algorithmes ont été évalués sur des corpus de test en utilisant la mesure Fscore (5.1) des documents bien classés, moyennée sur toutes les classes (avec $\beta = 1$ afin de ne privilégier ni la précision ni le rappel) (Goutte et Gaussier, 2005).

$$\text{Fscore}(\beta) = \frac{(\beta^2 + 1) \times \langle \text{Précision} \rangle \times \langle \text{Rappel} \rangle}{\beta^2 \times \langle \text{Précision} \rangle + \langle \text{Rappel} \rangle} \tag{5.1}$$

où la précision moyenne et le rappel moyen sont calculés comme :

$$\langle \text{Précision} \rangle = \frac{\sum_{i=1}^{n} \text{Précision}_i}{n}; \langle \text{Rappel} \rangle = \frac{\sum_{i=1}^{n} \text{Rappel}_i}{n} \tag{5.2}$$

Etant donné pour chaque classe i :

$$\text{Précision}_i = \frac{\{\text{Nombre de documents correctement attribués à la classe} i\}}{\text{Nombre de documents attribués à la classe } i} \tag{5.3}$$

$$\text{Rappel}_i = \frac{\{\text{Nombre de documents correctement attribués à la classe } i\}}{\text{Nombre de documents appartenant à la classe } i} \tag{5.4}$$

Le tableau 5.5 présente les résultats obtenus par les MVS sur la tâche de classification de CV/Lettre de motivation. On observe d'excellents résultats en Précision, Rappel et Fscore sur chaque sous-ensemble (entre 0,95 et 0,98) avec un apprentissage à partir de 40%. La matrice de confusion présentée en 5.6 montre que seulement 18 documents sur 867 sont mal classés.

	$A1$	$A2$	$A3$	$A4$	$A5$	Moyenne
Précision	0,98	0,98	0,97	0,98	0,99	0,98
Rappel	0,95	0,95	0,97	0,95	0,97	0,96
Fscore	0,97	0,97	0,97	0,97	0,98	**0,97**

TAB. 5.5 – *Précision, Rappel, Fscore obtenu par les MVS sur la classe CV pour la classification de CV/LM.*

	Documents type CV	Documents type LM
Documents classés CV	**421**	12
Documents classés LM	6	**428**

TAB. 5.6 – *Matrice de confusion MVS.*

Une analyse des CV/Lettre de motivation mal étiquetés montre deux types de CV mal classés : l'exemple du tableau 5.7 montre le cas d'un mauvais étiquetage dans le *Corpus de référence*, puisque le document contient plus vraisemblablement une lettre de motivation et un lien vers le CV. L'exemple au tableau 5.8, étiqueté LM, est un message généré automatiquement par des sites d'emploi, ceux-ci contenant des versions très courtes de CV avec un lien vers une version complète.

Mr ARVAUX Pierre
45 rue DE CHANTECLAIR 69440 VANNES. Tél 06.06.06.06.06
A la recherche d'un autre emploi, je me permets de vous adresser ma candidature pour le poste de Directeur d'hôtel car je pense correspondre au profil souhaité.En effet j'ai acquis une solide expérience en ma qualité de Responsable de Centre de Profit ainsi que Directeur de Cafétéria. reconnu,homme de terrain, j'ai un sens du commerce trés prononcé, j'ai managé jusqu'à 5O collarateurs.Je vous laisse le soin d'étudier ma candidature et me tiens à votre disposition pour de plus amples renseignements.
Le CV du candidat est consultable à l'adresse suivante :$http : ///CV?code = 3D - 1789031296_1 9543181$

TAB. 5.7 – *Premier exemple de CV mal classé.*

M. Zidounet Albert 4 rue de la Corniere 42490 Fraisage
akzeddoun@yahoo.fr Portable : 0606060606
Salaire souhaité : 21,000.00 EUR par an
Type d'emploi :Temps Plein Mobile géographiquement : non
Niveau d'études : Maîtrise, IEP, IUP, Bac + 4
Dernière expérience professionelle : 2002 à 2004 : Cabinet d'Expertise Comptable "Cofis" - Assistant en comptabilité
Le CV du candidat est consultable à l'adresse suivante :$http : ///CV?code = 130493543$

TAB. 5.8 – *Deuxième exemple de CV mal classé.*

Une fois la problématique de tri de CV/LM résolue, nous avons pu générer le *Corpus Mission* présenté par la suite et qui a servi aux expériences pour l'ensemble du troisième module et de la tâche de classement de candidatures.

5.5 Approche par classification de CV

Le CV étant la base d'une candidature, nous avons décidé dans un premier temps de les classer par leur contenu. Nous avons pris en considération le genre donné par le *Corpus de référence* (CV ou LM) afin de ne garder que les documents étiquetés CV. Après pré-traitement, nous avons effectué un premier apprentissage par les MVS. Les premiers résultats étant mitigés nous ont conduit à envisager une classification par n-grammes de mots. Nous avons extrait les uni-grammes et les bi-grammes de chaque classe (**OUI**/**NON**) avec leur probabilité p, puis nous avons calculé $\tilde{t}$ des n-grammes pour un document D, avec W les termes du document D et $\tilde{t} \in$ [**OUI**,**NON**] :

$$\tilde{t} = ArgMax_t p(t|W) = ArgMax_t \frac{p(W|t)p(t)}{p(W)} = ArgMax_t p(W|t)p(t) \tag{5.5}$$

Les deux dernières égalités proviennent de l'application du théorème de Bayes. En prenant comme hypothèse, compte-tenu de la sous-représentation de la classe **OUI** :

$$p(\tilde{t}) = 1 \; \forall \; \tilde{t} \tag{5.6}$$

on obtient :

$$\tilde{t} \approx ArgMax_t p(W|t) \cong ArgMax_t \prod_{i=1}^{|D|} p_t(W_i|W_{i-1}^1) \tag{5.7}$$

avec comme seconde hypothèse, pour obtenir des estimations fiables, malgré la faible taille des corpus disponibles :

$$p_t(W_i|W_{i-1}^1) \approx \lambda p_t(W_i|W_{i-1}) + (1-\lambda)p_t(W_i) \tag{5.8}$$

Le tableau 5.9 présente les résultats obtenus par les différents noyaux sur le *corpus OUI/NON* ainsi qu'un test sans racinisation. Les meilleurs résultats sont obtenus avec un noyau linéaire avec *Gamma* à 64, C à 8 et un type $C - SVC$. Le tableau 5.10 présente les résultats obtenus par les MVS et les n-grammes de mots.

Noyau MVS	Précision	Rappel	Fscore
Linéaire	0,60	0,61	**0,61**
Polynomiale	0,57	0,57	0,57
Radiale	0,57	0,55	0,56
Sigmoïdale	0,54	0,54	0,55
Linéaire sans racinisation	0,57	0,58	0,58

TAB. 5.9 – *Précision, Rappel, Fscore obtenus par les MVS en fonction du noyau sur le corpus OUI/NON.*

L'observation de ces résultats nous a poussé à envisager un combinaison de classifieurs, les MVS ayant de meilleures performances sur la classe NON alors que les n-grammes de mots classent mieux la classe OUI. Le mixage des deux techniques a

permis une très légère amélioration. Nous avons par ailleurs testé l'influence du métier sur les caractéristiques de la candidature (c'est-à-dire, les CV sont-ils indépendants du métier ?) sans observer d'amélioration particulière. L'ensemble des résultats concernant ces travaux sont présentés dans le tableau 5.10. Afin de tester la robustesse de ces méthodes, nous avons participé au DÉfi Fouille de Texte (DEFT) édition 2008 (voir annexe E). Nous avons par ailleurs effectué une série de tests avec l'application Weka (Witten et Frank, 1999), celle-ci permettant d'exécuter plusieurs d'algorithmes de *data-mining*. Weka a permis de tester différents outils tels que *J*48, une implémentation de l'algorithme *C*4.5 (Quinlan, 1993), *IB*1[7], *KStar*[8], un classificateur bayésien naïf *BayesNet* ainsi que deux méthodes de *clustering Cobweb*(Fisher, 1987), *FarthestFirst*[9] et *SimpleKmeans*, une implémentation des *k* moyennes (MacQueen, 1967). Les résultats obtenus, présentés en annexe F, sont de très bonne qualité mais l'ensemble de test est différent et ce dernier présente un déséquilibre entre les classes. Ce déséquilibre conduit les algorithmes à classer un maximum de candidatures en **non retenues**.

Afin d'évaluer nos méthodes de classification d'une candidature, et plus particulièrement les CV, nous avons effectué une scission du *Corpus de référence* en plusieurs sous-corpus : un sous-corpus contenant les CV classés en fonction d'une évaluation **OUI**/**NON** (désigné comme *Corpus OUI/NON*) ainsi que deux sous-corpus thématiques, afin de tester l'influence du métier sur les caractéristiques de la candidature. Ceux-ci contiennent l'ensemble des CV répondant à des missions de type **commercial** (nommé *Corpus commercial*, avec 715 CV) et **comptable** (*Corpus comptable*, avec 1546 CV).

	Précision		Rappel		Fscore	
Classifieur	MVS	*n*-grammes	MVS	*n*-grammes	MVS	*n*-grammes
Corpus OUI/NON	0,62	0,62	0,62	0,59	0,61	0,61
corpus commercial	**0,66**	0,62	**0,64**	0,57	0,58	0,58
corpus comptable	0,57	0,61	0,59	0,63	**0,64**	**0,64**

TAB. 5.10 – *Précision, Rappel, Fscore obtenus sur les différents corpus par les MVS et les n-grammes de mots.*

Les tableaux 5.11 et 5.12 montrent la répartition des erreurs pour chacun des classifieurs. L'observation de ces résultats nous a poussé à envisager une combinaison de classifieurs, les MVS ayant de meilleures performances sur la classe NON (375 documents bien classés contre 115 pour la méthode probabiliste) alors que les *n*-grammes de mots classent mieux la classe OUI (107 documents bien classés contre 40 pour les MVS).

	type OUI	type NON
classés OUI	40	50
classés NON	85	375

TAB. 5.11 – *Matrice de confusion des MVS.*

	type OUI	type NON
classés OUI	106	310
classés NON	19	115

TAB. 5.12 – *Matrice de confusion des n-grammes de mots.*

[7]http ://weka.sourceforge.net/doc/weka/classifiers/lazy/IB1.html
[8]http ://bio.informatics.indiana.edu/ml_docs/weka/weka.classifiers.kstar.KStar.html
[9]http ://www.lri.fr/ pierres/donn%E9es/save/these/weka-3-4/doc/weka/clusterers/FarthestFirst.html

Le tableau 5.13 présente le résultat d'un mixage simple entre les MVS et les *n*-grammes de mots sur la base d'un vote (N'ayant que deux systèmes, en cas de désaccord on privilégie le système ayant le meilleur résultat pour chaque classe). On observe une très légère augmentation des performances globales (Fscore de 0,66 pour le mixage contre 0,61 pour chaque méthode individuellement) mais sans véritable amélioration.

Classification par Mixage	(OUI)	(NON)	(Toutes classes)
Précision	**0,53**	0,83	**0,68**
Rappel	0,38	0,90	0,64
Fscore	0,44	**0,90**	0,66

TAB. 5.13 – *Précision, Rappel, Fscore obtenu par mixage des MVS et n-grammes de mots.*

5.6 Comparaison Candidature/Offre d'emploi par mesure de similarité

Les méthodes de classification ayant obtenues des résultats assez mitigés, nous avons opté pour une approche par mesure de similarité afin d'obtenir un classement des candidatures. Chaque document a été transformé en un vecteur avec des poids représentant la fréquence des termes (*Tf*) et par la suite le *Tf-idf*. Nous avons mis en place une approche par mesures de similarité, afin d'ordonnancer automatiquement l'ensemble des candidatures par rapport aux offres d'emploi proposées.

La première utilisée a été l'énergie textuelle (Enertex). Celle-ci a été utilisée avec succès dans différentes tâches du TAL, telles que le résumé automatique et la segmentation de paragraphes (Fernandez et al., 2007). Basée sur le modèle magnétique d'Ising, Enertex considère un document de n termes comme une chaîne de n unités binaires appelées *spins*. Les *spins* ***up*** étant les mots présents et les *spins* ***downs*** ceux absents. Dans notre modèle, nous sommes intéressés par le calcul de l'énergie textuelle entre l'offre d'emploi et chaque candidature :

$$J^{i,j} = \sum_{k=1}^{P} s_k^i s_k^j \tag{5.9}$$

$$E_{\mu,\nu} = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} s_\mu^i \, J^{i,j} \, s_\nu^j \tag{5.10}$$

où s_μ^i est le terme i d'une offre d'emploi μ de N termes, s_ν^j est le terme j d'une candidature ν de M termes, $J^{i,j}$ est l'intéraction entre les termes i et j d'un document de P phrases.

La deuxième mesure utilisée est l'algorithme de Needleman-Wunsch, couramment employé en bio-informatique pour aligner des séquences de nucléotides ou de protéines (Needleman et Wunsch, 1970). Nous considérons le vecteur « offre d'emploi » j

et le vecteur « candidature » d comme des séquences de termes et nous recherchons le meilleur score D entre les 2 séquences tels que :

$$D(j,d) = \begin{cases} D(j-1,d-1) + d(s_j,t_d) & //\text{substitution} \\ \min D(j-1,d) + 1 & //\text{insertion} \\ D(j,d-1) + 1 & //\text{délétion} \end{cases} \tag{5.11}$$

Les autres mesures de similarité que nous avons utilisées dans nos travaux sont décrites dans (Bernstein et al., 2005) : le cosinus (5.12), qui permet de calculer l'angle entre l'offre d'emploi et la réponse de chaque candidat, les distances de Minkowski (5.13) ($p = 1$ pour Manhattan et $p = 2$ pour la distance euclidienne), et la distance de Recouvrement (5.14). Une autre mesure testée est Okabis (5.15) (Bellot et El-Bèze, 2001), fondée sur la formule Okapi (Robertson et al., 1994), souvent utilisée en Recherche d'Information :

$$sim_{\text{cosine}}(j,d) = \frac{\sum_{i=1}^{n} j_i \cdot d_i}{\sqrt{\sum_{i=1}^{n} j_i^2 \cdot \sum_{i=1}^{n} d_i^2}} \tag{5.12}$$

$$sim_{\text{Minkowski}}(j,d) = \frac{1}{1 + \left(\sum_{i=1}^{n} |j_i - d_i|^p\right)^{\frac{1}{p}}} \tag{5.13}$$

$$sim_{Recouvrement}(j,d) = \frac{\sum_{i=1}^{n} j_i \cdot d_i}{Min\left(\sum_{i=1}^{n} |j_i|^2, \sum_{i=1}^{n} |d_i|^2\right)} \tag{5.14}$$

où j est l'offre d'emploi, d la candidature et i un des n termes.

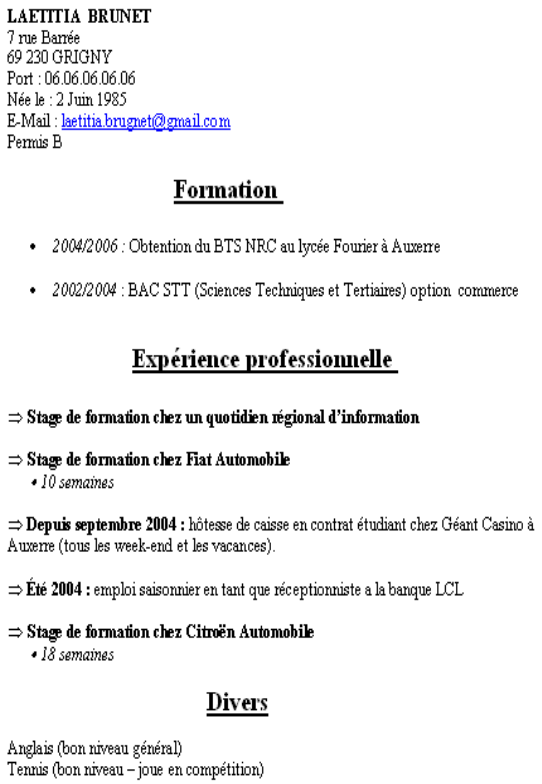
$$\text{Okabis}(d,j) = \sum_{i \in d \cap j} \frac{\sum_{i=1}^{n} j_i \cdot d_i}{\sum_{i=1}^{n} j_i \cdot d_i + \frac{\sqrt{|d|}}{M_d}} \tag{5.15}$$

Avec j une offre d'emploi, d la candidature, i un terme, j_i et d_i le nombre d'occurrences de i respectivement dans j et d et M_d leur taille moyenne.
Afin de combiner ces mesures, nous avons effectuer une normalisation de chaque mesure selon les minimums/maximums puis nous avons utilisé un algorithme de décision (AD) (Boudin et Torres Moreno, 2007) qui fusionne les valeurs λ obtenues par chaque mesure de similarité selon 2 tendances. Deux moyennes différentes sont calculées : la décision est calculée à partir de ces moyennes, la tendance positive (lorsque la mesure obtient un résultat $\lambda > 0{,}5$) et la tendance négative ($\lambda < 0{,}5$).

5.6.1 Exemple de candidatures

L'annexe D présente un exemple de découpage d'offre d'emploi tel que défini en 4.3. Le contenu des offres d'emploi est libre, mais présente des similarités dans leurs présentation et dans le vocabulaire employé (voir chapitre 4). Cette segmentation est

utilisée afin d'ordonnancer les candidatures comme nous le verrons dans la section 5.6.3.



LAETITIA BRUNET
7 rue Barrée
69 230 GRIGNY
Port : 06.06.06.06.06
Née le : 2 Juin 1985
E-Mail : laetitia.brugnet@gmail.com
Permis B

Formation

- *2004/2006* : Obtention du BTS NRC au lycée Fourier à Auxerre
- *2002/2004* : BAC STT (Sciences Techniques et Tertiaires) option commerce

Expérience professionnelle

⇒ **Stage de formation chez un quotidien régional d'information**

⇒ **Stage de formation chez Fiat Automobile**
- *10 semaines*

⇒ **Depuis septembre 2004 :** hôtesse de caisse en contrat étudiant chez Géant Casino à Auxerre (tous les week-end et les vacances).

⇒ **Été 2004 :** emploi saisonnier en tant que réceptionniste a la banque LCL

⇒ **Stage de formation chez Citroën Automobile**
- *18 semaines*

Divers

Anglais (bon niveau général)
Tennis (bon niveau – joue en compétition)

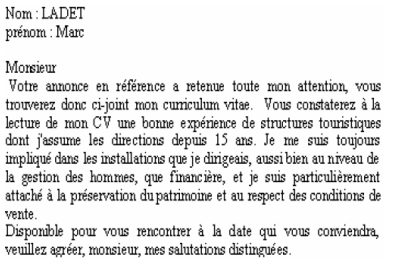
FIG. 5.2 – *Exemple de curriculum vitae en français.*

Les figures 5.2 et 5.3 présentent un exemple de CV et de LM[10]. De façon évidente, le style de chaque document est différent, la lettre de motivation se présentant comme un texte complet alors que le CV résume le parcours professionnel de la personne de façon concise. On observe par ailleurs que les CV, malgré un format libre, présentent des similarités entre eux du point de vue de leur contenu. On retrouve généralement les sections communes (« Expérience professionnelle », « Expérience personnelle », « Formation », « Divers » ou encore « Loisirs »), ainsi que certaines collocations pertinentes (« assistant commercial », « baccalauréat scientifique », etc.), comme décrit dans (Roche et Prince, 2008) et une présentation particulière (texte en gras ou en italique afin de définir chaque partie, indentation, etc. même si les différents outils que nous utilisons afin d'extraire le texte ne nous permettent pas de récupérer la structure du document (cf section 4.2).

5.6.2 Protocole expérimental

Nous souhaitons mesurer la similarité entre une offre d'emploi et ses candidatures. Le *Corpus Mission* contient 25 offres d'emplois associées à au moins quatre candida-

[10] Tous les documents ont été préalablement anonymisés.


Nom : LADET
prénom : Marc

Monsieur
Votre annonce en référence a retenue toute mon attention, vous trouverez donc ci-joint mon curriculum vitae. Vous constaterez à la lecture de mon CV une bonne expérience de structures touristiques dont j'assume les directions depuis 15 ans. Je me suis toujours impliqué dans les installations que je dirigeais, aussi bien au niveau de la gestion des hommes, que financière, et je suis particulièrement attaché à la préservation du patrimoine et au respect des conditions de vente.
Disponible pour vous rencontrer à la date qui vous conviendra, veuillez agréer, monsieur, mes salutations distinguées.

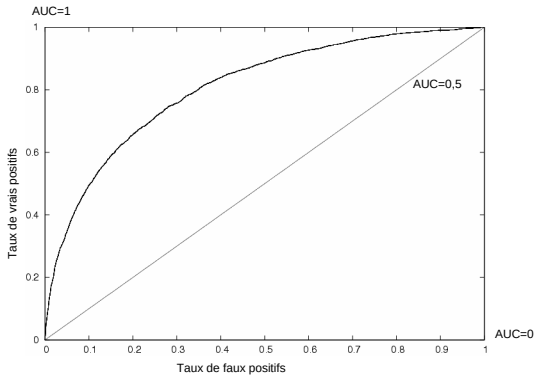


FIG. 5.4 – *Exemple de courbe ROC.*

tures. Nous transformons chaque document en une représentation vectorielle (Salton, 1991). Puis nous mesurons leur similarité à l'aide des mesures présentées en 5.6 : Enertex, cosinus, euclidienne, Manhattan, Recouvrement, Needleman-Wunsch (NW) et Okabis. Chacune de ses mesures produit un classement entre l'offre d'emploi et l'annonce. L'algorithme de décision combine l'ensemble de ces similarités afin d'ordonnancer les candidatures. Afin d'évaluer la qualité de l'ordonnancement obtenu nous utilisons les courbes *ROC* (*Receiver Operating characteristic*) (Ferri et al., 2002).

La courbe *ROC* est avant tout définie pour les problèmes à deux classes (positive et négative). Elle indique la capacité du classifieur à placer les exemples positifs devant les négatifs. Elle met en relation dans un graphique le taux de faux positifs (c'est à dire les candidatures **non pertinentes** mieux classées que les candidatures **pertinentes**) en abscisse et le taux de vrais positifs (c'est à dire les candidatures **pertinentes** classées en tête) en ordonnée. La surface sous la courbe *ROC* ainsi créée est appelée AUC (*Area Under the Curve*).

Dans notre problématique d'ordonnancement de candidatures, une courbe *ROC* parfaite (à savoir $AUC = 1$) reviendrait à obtenir toutes les candidatures pertinentes en tête de classement et toutes les candidatures non pertinentes à la fin. La ligne diagonale correspond à la performance d'un système aléatoire, le taux de vrais positifs est accompagné par une dégradation du taux de faux positifs. Cette situation correspond à $AUC = 0,5$. La figure 5.4 est un exemple de courbe *ROC* avec une ligne diagonale correspondant à une distribution aléatoire avec $AUC = 0,5$. Si les réponses

des candidats sont classées par ordre décroissant de pertinence (c'est-à-dire toutes les candidatures **retenues** après les candidatures **non retenues**), l'*AUC* est égal à 0. Pour qu'un système soit consideré comme efficace, il doit obtenir une courbe *ROC* avec la valeur la plus proche de 1. Le principal avantage des courbes *ROC* est leur résistance au déséquilibre dans le corpus (par exemple un déséquilibre entre les exemples **retenues** et **non retenues**). Le détail et les intérêts de cette mesure sont développés dans (Roche et Kodratoff, 2006). Pour chaque offre d'emploi du corpus, nous évaluons la qualité du classement obtenu avec cette méthode. Nous avons écarté lors de l'évaluation les candidatures où une pièce était manquante (CV ou LM).

5.6.3 Résultats

Nous nous sommes intéressés à la structure de nos données. Comme déjà mentionné en 4.3, une offre d'emploi est composée d'une brève description de l'entreprise (**D**), un titre (**T**), une mission (**M**) et un profil (**P**). Nous utilisons pour la suite deux combinaisons différentes de ce découpage :

- L'offre d'emploi complète (**DTMP**) ;
- L'offre d'emploi composée de son titre, sa mission et son profil (**TMP**).

	AUC	Enertex	Cosine	Minkowski	Manhattan	NW	Overlap	Okapi	Décision
DTMP	LM	0,524	0,567	0,561	**0,591**	0,481	0,573	0,521	0,596
	CV	0,524	**0,604**	0,510	0,503	0,532	0,543	0,541	0,562
	LM+CV	0,523	**0,621**	0,539	0,532	0,509	0,522	0,523	0,571
TMP	LM	0,524	0,560	0,559	0,580	0,473	0,562	0,513	**0,591**
	CV	0,523	**0,622**	0,508	0,501	0,544	0,538	0,542	0,561
	LM+CV	0,523	**0,622**	0,538	0,528	0,526	0,531	0,532	0,592

TAB. *5.14* – *AUC obtenu en fonction du découpage de l'offre d'emploi.*

La figure 5.14 présente les résultats obtenus en fonction de la candidature globale : le CV, la LM avec une offre d'emploi DTMP ou TMP. On observe que les meilleurs résultats sont obtenus en combinant les deux parties de la candidature (CV et LM) avec une offre d'emploi TMP, même si le CV reste le document majeur de la candidature avec un résultat très proche. La mesure cosinus obtient les meilleures résultats quelles que soient les approches (DTMP ou TMP) sauf lorsque l'on considère uniquement la LM où la mesure Manhattan obtient des résultats légèrement meilleurs. Nous supposons que les résultats obtenus par l'algorithme de décision sont bruités par la mauvaise performance de certaines mesures (Recouvrement ou Needleman-Wunsch). Nous avons retiré ces mesures par la suite. Nous observons que le CV contient plus d'informations pertinentes que la LM. Ceci confirmant notre intuition que le CV est le document principal de la candidature.

Le tableau 5.15 présente les résultats obtenus en effectuant un découpage des CV et LM en 3 parties afin d'identifier les morceaux contenant les informations les plus pertinentes. Nous obtenons des scores particulièrement bas dans les dernières parties des CV et des LM, qui permet de conclure que les informations les plus importantes afin

AUC	Enertex	Cosine	Minkowski	Manhattan	NW	Overlap	Okapi	Décision
CV_1/3	0,525	**0,589**	0,497	0,505	0,533	0,539	0,569	0,579
CV_2/3	0,524	**0,600**	0,524	0,520	0,515	0,577	0,560	0,580
CV_3/3	0,526	**0,526**	0,497	0,503	0,510	0,479	0,506	0,501
LM_1/3	0,527	0,573	0,561	**0,588**	0,480	0,571	0,528	0,580
LM_2/3	0,533	0,565	0,570	**0,578**	0,481	0,578	0,543	0,570
LM_3/3	0,516	0,447	0,528	**0,538**	0,416	0,446	0,439	0,470

TAB. 5.15 – *AUC obtenu en fonction du découpage des réponses.*

de déterminer si une candidature est pertinente ou pas, se situent dans les 2 premiers tiers de chaque document. Une étude détaillée du dernier tiers montre qu'il contient des informations rarement cruciales telles que « loisirs », « autres » pour le CV, ou encore les formules de politesse pour les LM (« je vous prie d'agréer », « en vous remerciant par avance de votre réponse », etc.).

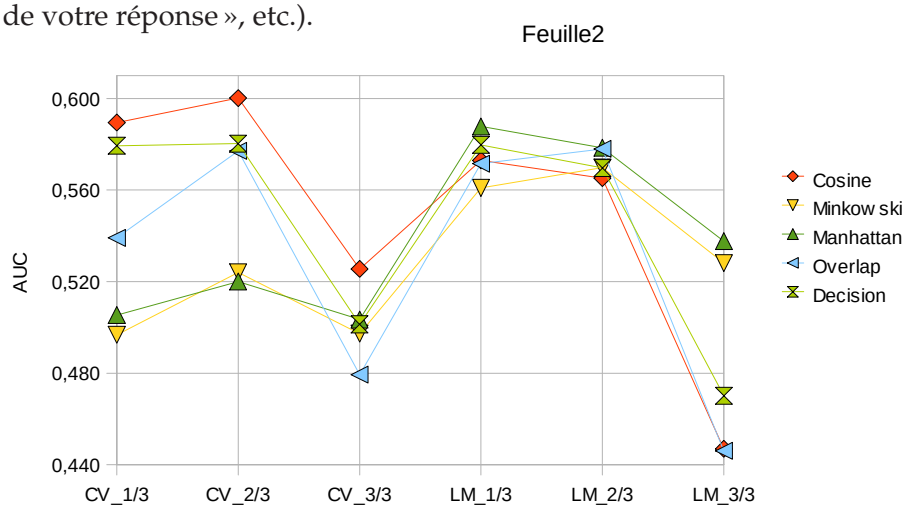


FIG. 5.5 – *Comparaison des résultats obtenus avec chaque mesure de similarité pour chaque tiers de document (CV et LM).*

La figure 5.5 présente, sous forme graphique, l'ensemble des résultats obtenus en fonction des découpages pour les mesures ayant obtenu les plus hauts scores (cosinus, Minkowski, Manhattan, Recouvrement et Décision).

5.6.4 Résultats sur la *Mission Virtuelle*

Nous présentons ici les résultats obtenus par notre système sur une offre d'emploi « *virtuelle* ». Il s'agit de regrouper dans cette mission une offre d'emploi et quelques candidatures de plusieurs missions (une vingtaine de candidatures prises dans quatre missions différentes avec des métiers différents), à thématique volontairement variées afin de tester le comportement de notre système face à une tâche simplifiée (par exemple une offre d'emploi pour un commercial associé a 4 candidatures de commerciaux, 4 candidatures de comptables, 4 candidatures de cuisiniers, etc.).Nous effectuons par la suite un nouvel étiquetage des candidatures, les candidatures appartenant à l'offre d'emploi

étant classées **retenue** et les autres **non retenues**. Le tableau 5.16 présente les résultats obtenus. On observe des résultats de bien meilleure qualité (entre 0,8 et 0,9), cependant la tâche est plus simple puisqu'il s'agit de catégoriser les candidatures en fonction de leur appartenance à un métier et non en fonction de leur pertinence. Les candidatures avec les termes les plus proches de l'offre d'emploi sont donc mieux classées.

	Enertex	cosinus	Minkowski	Manhattan	Recouvrement	Okapi	Décision
DTMP/tous	0,800	0,978	0,889	0,889	0,933	0,933	0,933
LM	0,800	0,733	0,689	0,689	0,667	0,711	0,622
CV	0,778	0,978	0,933	0,933	0,956	0,956	0,889
TMP/tous	0,778	0,933	0,800	0,800	0,889	0,889	0,889
CV	0,778	0,733	0,689	0,689	0,733	0,733	0,622
LM	0,756	0,956	0,822	0,822	0,933	0,933	0,867

TAB. *5.16* – *AUC obtenus sur le corpus Mission Virtuelle.*

5.7 Filtrage et pondération des mots selon leur étiquette grammaticale

Afin de tenter d'essayer d'améliorer les résultats obtenus par les mesures de similarité (section 5.6), nous avons effectué une extraction d'informations grammaticales du corpus à l'aide de TreeTagger[11] (Schmid, 1994). Nos observations du corpus montrent que les CV sont des documents textuels singuliers : généralement courts (inférieur à une page), syntaxiquement pauvres (peu de sujets et de verbes dans les phrases, phrases sous forme de résumé, nombreuses énumérations de noms et d'adjectifs, contenu fortement symbolique etc.) et avec une structure particulière (Roche et Kodratoff, 2006; Clech et Zighed, 2003). D'où la difficulté de traitement automatique de ces documents malgré le fait qu'il est relativement facile pour un lecteur d'identifier les données signalétiques, la formation, les expériences professionnelles et les loisirs du candidat sans pour autant être un spécialiste du domaine (Desmontils et al., 2002). Le tableau 5.17 présente les statistiques des catégories grammaticales du *Corpus Mission*. Les mots appartenant à des catégories grammaticales spécifiques peuvent donc être plus ou moins intéressants. Nous proposons donc de ne conserver que les mots suivants : **N** (Nom), **A** (Adjectif) et **V** (Verbe). Seulement ces mots seront la base de la représentation vectorielle des documents. Par ailleurs, différentes combinaisons (NV, NA,VA et NVA) et pondérations[12] (2N4A, 3N2A, 4N2A, 2N3A, 4N3A, 3N3A) ont été expérimentées.

Le tableau 5.18 présente les différents résultats obtenus en effectuant une extraction de terminologie sur le *Corpus Mission*. Différents tests ont été effectués en fonction de la terminologie conservée. Afin d'alléger les temps de traitements et d'améliorer les résultats de l'algorithme de décision, nous n'avons conservé que les mesures de simila-

[11] Treetagger est un système d'étiquetage grammatical des mots. http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[12] NA signifie que l'on combine les noms et les adjectifs, 2N3A signifie que l'on double le poids des noms combiné avec un poids triple pour les adjectifs, etc.

Catégorie	CV		LM		(CV+LM)		Mission	
Nom	27669	72,89%	111119	57,42%	388448	67,68	4617	63,16%
Verbe	43778	11,49%	55958	28,92%	99736	17,36%	1536	21,01%
Adjectif	59501	15,62%	26442	13,66%	85943	14,96%	1157	15,83%

TAB. *5.17* – *Statistiques des catégories grammaticales du Corpus Mission.*

rité ayant obtenues les meilleurs résultats lors de ces tests (cosinus, Minkowski, Manhattan, Recouvrement et Décision). Le tableau présente les AUC maximales obtenues toutes mesures de similarité confondues avec chacune des catégories grammaticales, **N** (Nom), **V** (Verbe), **A** (Adjectif) ou avec les combinaisons de catégories grammaticales, **NV**, **NA**, **VA** et **NVA**.

	A	**V**	**N**	**NV**	**NA**	**VA**	**NVA**
DTMP/CV+LM	**0,600**	0,563	0,594	0,610	0,595	0,532	**0,600**
LM	0,596	0,557	0,561	0,563	0,572	0,555	**0,577**
CV	0,596	0,524	0,572	0,577	**0,590**	0,519	0,575
CV1-3	0,533	0,555	0,565	0,550	**0,574**	0,524	0,563
CV2-3	**0,605**	0,563	0,566	0,575	0,567	0,535	0,566
CV3-3	0,530	**0,539**	0,501	0,511	0,520	0,512	0,516
LM1-3	0,568	0,532	0,557	0,533	**0,582**	0,542	0,574
LM2-3	0,562	0,526	0,535	**0,567**	0,542	0,546	0,559
LM3-3	**0,567**	0,559	0,501	0,521	0,500	0,529	0,527
TMP/CV+LM	0,576	0,555	**0,608**	0,584	0,581	0,527	0,606
LM	**0,589**	0,553	0,559	0,558	0,569	0,523	0,571
CV	0,564	0,507	0,564	0,565	0,591	0,511	**0,573**
CV1-3	0,532	0,555	0,555	0,538	0,571	0,504	**0,557**
CV2-3	**0,600**	0,565	0,575	0,579	0,571	0,534	0,567
CV3-3	**0,530**	0,534	0,517	0,510	0,524	0,500	0,512
LM1-3	0,583	0,523	0,553	0,519	0,563	0,538	**0,567**
LM2-3	0,523	0,533	0,532	**0,559**	0,544	0,545	0,553
LM3-3	**0,572**	0,553	0,521	0,526	0,511	0,528	0,541

TAB. *5.18* – *Résultats avec extraction de terminologie nominale sur le Corpus Mission.*

On observe que malgré quelques résultats approchant, principalement pour **A** (Adjectif) ou **NVA** (Nom-Verbe-Adjectif), l'ensemble des résultats restent en dessous des résultats obtenus précédemment (5.14). Nous avons tenté par la suite différentes pondérations pour chaque catégorie grammatical, **2N4A**, **3N2A**, **4N2A**, **2N3A**, **4N3A**, **3N3A**. Le tableau 5.19 résume l'ensemble de ces tests.

Notre approche utilise jusqu'à présent des poids *tf*. Le tableau 5.20 présente les résultats obtenus entre les deux mesures *Tf* et *Tf-idf* avec et sans lemmatisation. On observe une baisse de la performance globale lors de l'utilisation du *Tf-idf*. Nous attribuons la baisse de performance de notre système en utilisant *Tf-idf* à la faible taille de nos corpus et au faible nombre de candidatures dans certaines missions (tableau 5.2).

	2N4A	**3N2A**	**4N2A**	**2N3A**	**4N3A**	**3N3A**
DTMP/CV+LM	**0,637**	0,607	0,601	0,632	0,615	0,619
LM	**0,587**	0,578	0,575	0,582	0,578	0,580
CV	0,576	**0,606**	0,596	0,574	**0,606**	0,580
CV1-3	0,557	0,568	0,563	0,556	**0,575**	0,570
CV2-3	0,570	0,577	0,566	**0,579**	0,570	0,578
CV3-3	0,532	0,538	0,540	0,540	**0,555**	0,536
LM1-3	**0,594**	0,570	0,574	0,582	0,591	0,578
LM2-3	**0,578**	0,575	0,571	0,576	0,569	0,576
LM3-3	0,533	0,541	**0,549**	0,535	0,545	0,544
TMP/CV+LM	0,622	0,608	0,597	**0,627**	0,609	0,611
LM	**0,580**	0,571	0,573	0,578	0,570	0,573
CV	0,579	0,589	0,584	0,581	**0,602**	0,581
CV1-3	0,551	0,557	0,553	0,554	**0,570**	0,568
CV2-3	0,570	0,576	**0,580**	0,574	0,577	0,578
CV3-3	0,535	0,539	0,539	0,541	0,549	**0,560**
LM1-3	0,583	0,568	0,569	**0,587**	0,574	0,579
LM2-3	**0,578**	0,569	0,564	0,573	0,562	0,571
LM3-3	0,552	0,561	**0,567**	0,551	0,562	0,563

TAB. *5.19* – *Résultats avec pondération de la terminologie sur le Corpus Mission.*

On observe cependant une amélioration des résultats lors de l'utilisation de la lemmatisation (0,616 contre 0,607 et 0,602 contre 0,592).

	Tf avec lemmatisation	*Tf* sans lemmatisation	*Tf-idf* avec lemmatisation	*Tf-idf* sans lemmatisation
DTMP/CV+LM	0,621	**0,642**	**0,625**	**0,636**
LM	0,620	0,595	0,586	0,549
CV	0,603	0,589	0,601	0,607
TMP/CV+LM	**0,642**	0,624	0,614	0,618
CV	0,623	0,608	0,602	0,600
LM	0,590	0,584	0,586	0,547
Moyenne	0,616	0,607	0,602	0,592
Max	0,642	0,642	0,625	0,636

TAB. *5.20* – *Comparaison de résultats entre Tf et Tf-idf avec et sans lemmatisation.*

5.8 *N*-grammes de caractères

Utilisés principalement en reconnaissance de la parole, la notion de *n*-grammes de caractères prit davantage d'importance avec les travaux de (Damashek, 1995) sur le traitement de l'écrit. Ils montrent que ce découpage ne fait pas perdre d'information. De nombreux travaux depuis ont montré l'efficacité des *n*-grammes de carac-

tères comme méthode de représentation des textes que ce soit pour des tâches d'indexation (Mayfield et McNamee, 1998), d'identification de la langue d'un document (Juola, 1998), de reconnaissance d'auteurs (Khmelev et Tweedie, 2002), d'analyse exploratoire multidimensionnelle en vue d'une recherche d'information dans des corpus textuels (Lelu et al., 1998) ou encore dans différentes tâches de classification (Teahan, 2000; Teytaud et Jalam; Hurault-Plantet et al.).

Un n-gramme de caractères est une séquence de n éléments consécutifs. Pour un document quelconque, l'ensemble des n-grammes que l'on peut générer est le résultat que l'on obtient en déplaçant une fenêtre de n cases sur le corps du texte. Les fréquences des n-grammes trouvés sont calculées. Par exemple, la phrase "développeur php mysql" est représentée avec des 3-grammes par [dév, éve, vel, elo, lop, opp, ppe, peu, eur, ur_, r_p, _ph, php, hp_, p_m, _my, mys, ysq, sql]. Nous représentons les n-grammes en utilisant le caractère "_" pour caractériser les espaces. (Miller et al., 2000) montrent que des systèmes de recherches documentaires basés sur les n-grammes ont gardé leurs performances sur des textes avec des taux de déformations (erreurs typographiques, inversion de caractères, suppression de caractères, etc.) d'environ 30%. Par ailleurs, le choix des n-grammes de caractères apporte un autre avantage très important : il permet de contrôler la taille du lexique et de la maintenir à un seuil raisonnable pour de très larges corpus (Biskri et Meunier, 2002). En effet, un découpage en mots fait que la taille du lexique est d'autant plus grande que le corpus est grand. Cette limite subsiste malgré certains aménagements tels que le « nettoyage » des mots fonctionnels, la lemmatisation, etc. Un lexique obtenu suite à un découpage en n-grammes de caractères ne peut dépasser la taille de l'alphabet à la puissance n. Le choix d'un découpage en quadri-grammes pour une langue de 26 caractères donnerait un lexique de 456 976 quadri-grammes possibles. Si on élimine les combinaisons qu'il est impossible de rencontrer (p.ex. AAAA, ABBB, BBBA, etc.), ce nombre diminue de façon considérable. Toujours dans le but de diminuer ce lexique, nous effectuons aussi un filtrage sur les n-grammes de caractères obtenus en supprimant ceux rencontrés dans tous les documents. En contrepartie, à chaque fois que l'on augmente la taille de notre séquence, on augmente de façon conséquente celui-ci (5-grammes, lexique de 11881376 ; 6-grammes, lexique de 308915776, etc.).

Le tableau 5.21 présente un comparatif des résultats obtenus en fonction des différentes tailles de n-grammes de caractères. Nous avons testé différents n-grammes (3/4/5/6-grammes). Les meilleurs performances étant obtenues avec des 5-grammes.

	3-grammes tf	*4-grammes tf*	*5-grammes tf*	*6-grammes tf*
DTMP/tous	0,575	0,584	**0,601**	0,598
LM	**0,585**	0,573	0,561	0,574
CV	0,579	**0,585**	0,597	**0,599**
TMP/tous	0,572	0,575	0,598	0,587
CV	0,577	0,578	0,558	0,597
LM	0,447	0,572	0,597	0,573

TAB. 5.21 – *Comparaison de résultats entre les différentes tailles de n-grammes de caractères.*

5.9 Enrichissement sémantique de la mission

L'observation des mots ayant le plus d'influence lors du calcul de la mesure de similarité, nous a conduit à envisager un enrichissement du contenu de la mission à l'aide d'une ontologie obtenue à partir de la base ROME[13] de l'ANPE[14]. Ainsi, pour chaque mission, nous effectuons un enrichissement de celle-ci à l'aide des compétences et niveaux d'études nécessaires afin de remplir cette fonction[15]. Les résultats de ces tests sont présentés dans le tableau 5.24 sous l'appellation *Offre enrichie*. Ceux-ci restent relativement décevants cependant un travail plus en profondeur sur la collecte et l'organisation de cette enrichissement sémantique permettrait sans doute d'apporter des améliorations.

5.10 *Relevance Feedback*

L'enrichissement sémantique de la Mission obtenant des résultats inférieurs aux précédents, nous avons modifié le système afin d'intégrer un processus de retour de pertinence (*Relevance Feedback*) (Spärck, 1970). Le *Relevance Feedback* est une méthode classique de reformulation de requête afin d'améliorer les résultats obtenus au préalable. Par exemple, un utilisateur vérifie soigneusement la réponse d'un ensemble résultant d'une première requête, puis il reformule la requête en ajoutant les documents évalués pertinents pour améliorer le résultat de cette nouvelle requête. L'algorithme de (Rocchio, 1971) et ses différentes variations ont été largement utilisées dans le domaine de la recherche d'information (Frakes et Baeza-Yates, 1992; Leuski, 2000) et la catégorisation de texte (Joachims, 1997). Plus proche des Ressources Humaines, (Rafter et al., 2000b) proposent un système de *Relevance Feedback* afin de guider l'internaute dans sa recherche d'emploi à partir d'informations récoltées sur le site d'emploi JobFinder [16] . Dans notre système, la méthode *Relevance Feedback* permet de prendre en compte les choix du recruteur lors d'une première évaluation de quelques CV. Notre objectif n'étant pas un système capable de trouver la candidature idéale, mais un système capable de reproduire le jugement du consultant en recrutement. Il est extrêmement important pour un recruteur de pouvoir repêcher une candidature pertinente qu'il aurait évaluée de façon trop rapide. L'objectif de notre système est donc de l'aider à limiter ce genre d'erreur. Notre système va donc permettre d'effectuer un ordonnancement de façon assistée. Cette approche repose sur l'exploitation des documents retournés en réponse à une première requête pour améliorer le résultat de la recherche (Salton et Buckley, 1990). Dans notre contexte, nous effectuons un tirage aléatoire de quelques candidatures (de une à six dans nos expérimentations) parmi l'ensemble des candidatures étiquetées comme **pertinentes**. Celles-ci sont ajoutées à la Mission. Nous

[13] Répertoire Opérationnel des Métiers et des Emplois.

[14] http ://www.anpe.fr/espacecandidat/romeligne/RliIndex.do

[15] Exemple : 32321/développeur/**Bac+2** à **Bac+4** en **informatique CFPA**, **BTS**, **DUT** ; Participe au **développement** et à la **maintenance** des **applications informatiques**, l'**analyse fonctionnelle**, la **conception technique**, le **codage**, la **mise au point** et la **documentation** des **programmes**, etc.

[16] JobFinder (jobfinder.com)

enrichissons ainsi l'espace vectoriel par les termes appartenant à des candidatures jugées pertinentes par un consultant en recrutement. Ceci nous permet d'effectuer un nouveau calcul de similarité entre la candidature que nous évaluons et la Mission.

5.10.1 Expériences

Afin de pouvoir tester le principe de *Relevance Feedback*, il a été nécessaire de retirer certaines missions du *Corpus Mission*, celles-ci ne possédant pas un nombre de candidatures **retenues** suffisantes. Les missions avec moins de 5 candidatures **retenues** ont été retirées comme le montre le tableau 5.22 (les missions conservées sont grisées). Nous désignerons ce nouveau corpus comme *Corpus Mission RF*.

Numéro	Nombre de Candidatures	Candidatures	
		retenue	**non retenue**
32559	4	2	2
32560	8	3	5
34854	15	4	11
34692	16	4	12
33867	18	4	14
00010	21	1	6
00014	30	5	25
34990	32	4	28
34990	**32**	**7**	**25**
00015	36	1	35
34861	**40**	**14**	**26**
33273	51	1	50
31248	54	2	52
31702	**55**	**23**	**32**
32461	**60**	**7**	**53**
33633	**65**	**18**	**47**
34865	**67**	**10**	**57**
34783	**108**	**9**	**99**
33746	**116**	**60**	**56**
33553	**117**	**17**	**100**
33725	**118**	**43**	**75**
31022	**221**	**28**	**193**
31274	**224**	**26**	**198**
34119	**257**	**10**	**247**
31767	**437**	**51**	**386**
Total conservé	1917	323	1594

TAB. 5.22 – *Missions conservées (grisés) du Corpus Mission pour composer le Corpus Mission RF.*

Le tableau 5.23 présente les résultats obtenus pour chaque taille de *Relevance Feedback* (RF1 correspond à une seule candidature ajoutés a la mission, RF2 deux, etc.) et par rapport à la référence (TF). Nous utilisons un *residual ranking* (Billerbeck et Zobel, 2006) : les documents utilisés pour le *Relevance Feedback* sont retirés de la collection avant d'effectuer la requête reformulée. Chaque test a été effectué une centaine de fois avec une distribution aléatoire des candidatures **pertinentes** ajoutés au *Relevance Feedback*. On observe une progression positive du score *AUC* entre la référence et le RF7

pour 10 missions sur 12. L'étude détaillée des résultats montrent que la mission 33746 comporte quelques candidatures vides étiquetées **pertinentes**. Ce qui conduit le système à dégrader les résultats obtenus lorsque celles-ci sont sélectionnées. La mission 34783 obtient de bon score dès le départ mais comporte peu de candidatures évaluées pertinentes (7) dont des candidatures incomplètes (sans LM), ce qui entraîne une dégradation du score puisqu'elles ne sont pas prise en compte dans le résultat. La figure 5.6 illustre graphiquement les résultats des missions ayant eu une progression avec le retour de pertinence.

	LM/TF	RF1	RF2	RF3	RF4	RF5	RF6	RF7
Mission-31022	0,462	0,447	0,435	0,430	0,444	0,469	0,472	**0,503**
Mission-31702	0,414	0,514	**0,523**	0,518	0,517	0,486	0,477	0,483
Mission-31274	0,651	0,673	0,693	0,699	0,694	0,697	**0,700**	0,691
Mission-31767	0,461	0,488	0,501	**0,511**	0,503	0,504	0,507	0,501
Mission-33553	0,776	0,777	0,753	0,762	0,775	0,785	**0,793**	0,791
Mission-33633	0,561	0,555	0,574	0,586	0,574	**0,593**	0,579	0,571
Mission-33725	0,595	0,685	0,686	0,703	0,712	0,710	0,716	**0,728**
Mission-33746	**0,696**	0,612	0,594	0,582	0,575	0,566	0,563	0,570
Mission-34119	0,591	0,655	0,708	**0,696**	0,695	0,690	0,692	0,673
Mission-34783	0,827	**0,828**	0,809	0,816	0,807	0,796	0,793	0,741
Mission-34861	0,655	0,705	0,708	0,715	0,718	0,713	**0,735**	0,731
Mission-34865	0,903	0,904	0,899	0,894	0,901	0,911	**0,914**	0,911

TAB. 5.23 – *Comparaison entre les scores AUC obtenues en fonction de chaque taille de Relevance Feedback pour chaque mission, trié en fonction du numéro de la mission.*

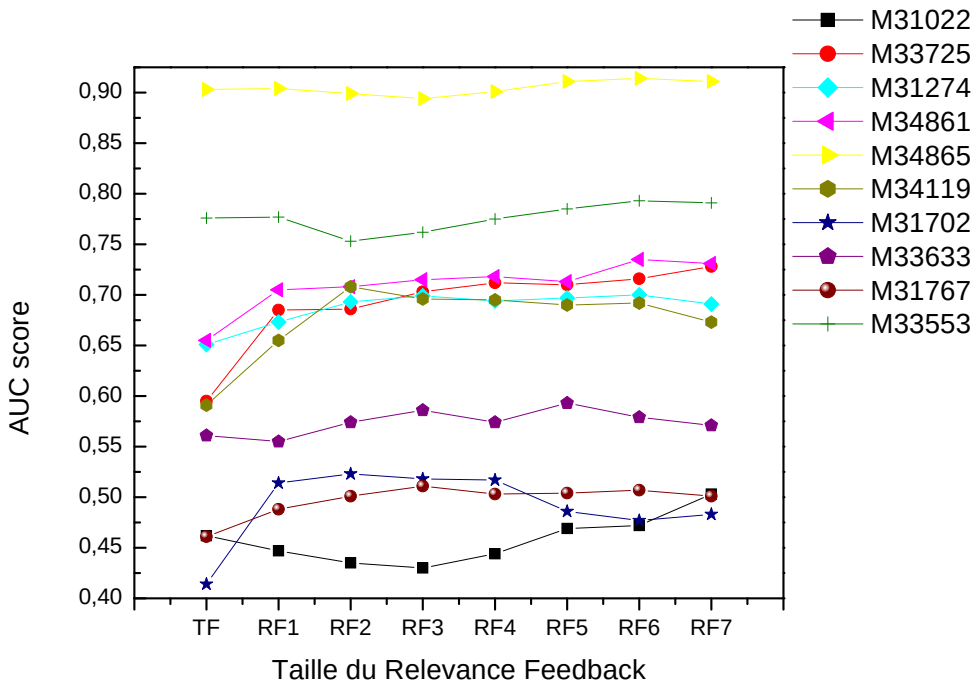


FIG. 5.6 – *Comparatif entre les tailles de RF et les résultats de référence (TF) pour chaque mission*

La progression n'est pas régulière mais le processus de Relevance Feedback améliore de façon certaine les résultats obtenus. Il serait intéressant de pouvoir tester celui avec un nombre plus important de candidatures ajoutées afin de connaître les limites de celui-ci, cependant cela nous obligerais a retirer à nouveau des missions du *Corpus Mission RF*.

5.11 Comparatif des résultats obtenus par chaque méthode

Le tableau 5.24 présente un comparatif des meilleurs résultats obtenus par chaque méthode. Nous effectuons une moyenne des *AUC* obtenues pour chaque mesure. *TF* sont les résultats obtenus à partir d'une représentation en fréquence de termes. *TF-IDF* utilise le produit de la fréquence des termes et de la fréquence inverse des documents.

	N-grammes	*Offre enrichie*	*TF*	*TF-IDF*	*Étiquettes grammaticales*	*Relevance Feedback*
Offre /CV et LM	0.60	0.62	0.64	0.64	0.64	**0.66**

TAB. 5.24 – *Comparaison entre les scores AUC obtenues par chaque méthode.*

Les représentations *TF* et *TF-IDF* donnent des résultats sensiblement similaires avec des scores *AUC* de 0.64, la taille réduite de nos corpus pouvant expliquer ces résultats. Les combinaisons et pondérations d'étiquettes grammaticales (voir section 5.7) ne semblent pas apporter d'amélioration (*Grammatical Labels*). Les résultats *N-grammes* présentés sont obtenus à partir de 5-grammes. Avec un score *AUC* de 0.6 au mieux (les scores des 3 et 4-grammes sont proches de 0,5), ceux-ci restent relativement faibles. Différents post-processus dans le but d'éliminer les séquences de caractères trop fréquentes ou non significatives ont été envisagés mais sans réellement d'amélioration majeure. Avec un score *AUC* de 0.62, l'enrichissement sémantique (*Offre enrichie*) ne semble pas améliorer la performance générale du système.

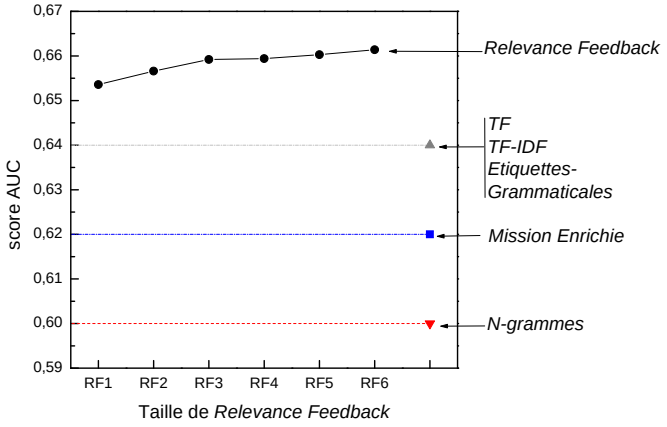


FIG. 5.7 – *Comparaison entre les scores AUC pour chaque méthode.*

Nous observons cependant une amélioration de la moyenne obtenue avec le retour de pertinence. RF1 obtient une moyenne de 0.65 et RF6 de 0.66. Il est malheureusement impossible de tester RFn avec $n > 6$ puisque le nombre de candidature **pertinentes** est trop petit dans certaines offres d'emplois (voir 5.22). La figure 5.7 illustre graphiquement ces résultats.

5.12 Conclusion

Le traitement des candidatures de façon automatisé est une tâche extrêmement difficile car l'information est en format libre malgré une structure conventionnelle. Ce chapitre a mis en avant le module de traitement des réponses à des offres d'emplois. L'évaluation des candidatures restant fortement subjective, celui-ci a pour but la mise en place d'un système d'aide à la décision pour le recruteur. Ce dernier effectue une évaluation des premières candidatures afin de guider le système par la suite. Après différentes étapes de filtrage et de racinisation, nous produisons une représentation vectorielle afin d'effectuer un classement des candidatures à l'aide de mesures de similarité et diverses représentations des documents. Les résultats à l'aide du retour de pertinence montrent une amélioration des AUC obtenues. Nous envisageons quelques tests complémentaires (recherche de critères discriminants sur les candidatures identifiées comme négatives, pondération en fonction de l'importance de chacune des parties de la mission, etc.) pouvant apporter de nouvelles améliorations. Nous souhaitions par ailleurs inclure d'autres paramètres tels que la richesse du vocabulaire et l'orthographe afin d'évaluer la lettre de motivation, ceux-ci étant à l'heure actuelle faiblement exploités lors de la prise de décision par les recruteurs mais les premiers tests sont restés peu concluants pour l'instant. Nous envisageons par ailleurs la mise en place d'un système d'évaluation de CV afin d'effectuer l'opération inverse (le candidat dépose son CV et le système lui propose les missions les plus adaptées à son profil). Ces travaux ont été publiés dans (Kessler et al., 2008), (Kessler et al., 2009a) et (Kessler et al., 2009b).

Chapitre 6

Conclusion

La technologie moderne ne peut ignorer un domaine d'activité comme le recrutement en ligne où le traitement de l'information est extrêmement présent que ce soit pour identifier l'information et/ou son destinataire ou identifier les informations qui peuvent permettre d'effectuer un choix stratégique. La quantité d'information disponible augmentant chaque jour, il est essentiel de pouvoir mettre en place des outils permettant de les traiter rapidement et intelligemment, sous peine d'être submergés d'informations. Nous présentons dans cette thèse les travaux que nous avons effectués sur le routage de courriel et le traitement automatique d'informations appliqués aux ressources humaines.

Le chapitre 3 est consacré au routage de courriel. Le manque de corpora disponibles en français nous a amené à construire un corpus réaliste à partir de plusieurs listes de diffusions. Un certain nombre de pré-traitements spécifiques ont été nécessaires afin de réaliser un modèle vectoriel, en raison des particularités de cette forme de communication (fautes d'orthographes, manque de ponctuation, manque d'accentuation, phonécrit, etc.). À partir de cette représentation des données, nous avons réalisé une étude de différentes méthodes d'apprentissage automatique non supervisé (*Fuzzy k-means*) et supervisé (*Machine à Vecteur Support* ; MVS), chacune ayant leurs propres avantages et inconvénients. Nous avons proposé une méthode hybride permettant de combiner les deux types d'apprentissage en effectuant une initialisation semi-supervisé afin d'éviter les problèmes de minima locaux d'une initialisation aléatoire pour l'apprentissage non-supervisé et la nécessité d'un étiquetage préalable pour l'apprentissage supervisé. Les résultats obtenus sont intéressants puisque la méthode hybride est proche de celle des MVS : en fournissant 0,05% d'exemples appris en initialisation (une vingtaine de courriels), le taux d'erreur est proche des performances des MVS.

Le chapitre 4 met en avant le module de catégorisation, premier composant d'E-Gen, système pour le traitement automatiquement des offres d'emploi. Le temps étant un facteur déterminant dans le domaine du recrutement en ligne, il était important de trouver un processus capable d'intégrer de façon automatique une offre d'emploi afin de pouvoir la diffuser rapidement après validation. Le traitement de ce type d'information est difficile car il s'agit de texte libre (non structuré) et de sources diverses.

L'analyse du corpus d'offres d'emploi a permis de détecter des blocs d'informations possédant des caractéristiques communes. Chaque offre se décompose en 4 parties distinctes, la description de l'entreprise qui recrute, un intitulé, la mission qui sera confiée au futur collaborateur et le profil auquel celui-ci devra correspondre. Nous avons donc utilisé la représentation vectorielle de textes afin d'effectuer une classification avec des machines à support vectoriel (MVS) ainsi qu'avec des bi-grammes de mots. Cette classification est par la suite transmise à un post-processus piloté par un automate de Markov et une méthode *Branch and Bound* qui améliorent sensiblement les résultats (plus de 80% d'offres d'emplois correctement integrées). Ce module d'E-Gen est actuellement en ligne sur le serveur d'Aktor et permet un gain de temps considérable dans le traitement quotidien des offres d'emploi avec un coût minimal en termes d'intervention humaine.

Le chapitre 5 présente les deuxième et troisième modules du système E-Gen, consacré au traitement des réponses à des offres d'emplois. Une séparation des *curiculum vitae* et des lettres de motivation a été effectuée. Les caractéristiques de chaque document, une fois transformé en version numérique, ont permis d'effectuer une classification extrêmement fiable avec un système de règles et un classifieur de type MVS (Fscore, Rappel et précision de 0,98). Le troisième module est consacré à la mise en place d'un système d'aide à la décision pour le recruteur. La quantité de réponses reçues par ce dernier entraîne un long travail de lecture, qui peut conduire à des oublis voire des erreurs. Il est donc extrêmement important pour un recruteur de pouvoir repêcher une candidature pertinente qu'il aurait évaluée de façon trop rapide. L'objectif de notre système n'est donc pas de trouver la candidature idéale mais de reproduire le jugement du consultant en recrutement. Nous avons donc utilisé la représentation vectorielle de textes afin d'effectuer un classement des candidatures à l'aide de mesures de similarité et de diverses représentations des documents. Nous avons introduit un processus de retour de pertinence, permettant d'inclure l'opinion du recruteur lors de l'évaluation de la candidature par notre système. Les résultats à l'aide du retour de pertinence montrent une nette amélioration des AUC obtenues. Cependant le manque de candidatures **pertinentes** dans certaines offres d'emploi du *Corpus Mission* limite actuellement les tests de ce processus : afin d'effectuer l'évaluation nous retirons les documents qui ont permis d'effectuer le retour de pertinence. Diverses pistes restent à explorer par ailleurs concernant l'ordonnancement des candidatures telles que la recherche de critères discriminants sur les candidatures identifiées comme négatives, permettant d'apprendre des exemples négatifs et l'utilisation d'algorithmes tels que celui de (Rocchio, 1971) ou encore une exploitation plus approfondie de la lettre de motivation. À l'heure où le Pôle emploi[1] se dit débordé face à la quantité de demandeurs d'emploi, une perspective intéressante aussi serait la mise en place d'un système chargé d'effectuer l'opération inverse (le candidat dépose son CV et le système lui propose les missions les plus adaptées à son profil).

Les méthodes numériques abordées dans cette thèse ont permis le développement d'outils sur deux exemples d'applications intéressantes, le routage de courriels et le traitement automatique des offres d'emploi et de leurs réponses. Il s'agit d'applications non triviales car on y travaille avec des événements très rares (les courriels), avec une

[1] www.pole-emploi.fr

information fortement non structurée (le traitement des offres d'emploi) ou encore subjective (pour l'ordonnancement des réponses). Le choix que nous avons fait de développer des méthodes statistiques est motivé par les nombreux avantages qui leur sont associés. Elles sont robustes, rapides et peu dépendantes de la langue des documents sources. Par ailleurs, celles-ci ont permis d'obtenir des résultats globalement intéressant et montre que pour traiter le texte une analyse très fine n'est pas forcément nécessaire. Les méthodes numériques agissent à grande échelle. Mais elles ont aussi leurs limites. Le modèle du sac de mots est une simplification exagérée qui néglige la structure du document, ce qui implique une perte importante d'information que celles-ci proviennent de CV, d'offres d'emplois ou de courriels. Par ailleurs, les méthodes numériques sont fortement dépendantes des corpora annotés, ceux-ci étant généralement peu nombreux. La génération des corpus afin d'entraîner nos modèles ont nécessité beaucoup de temps, de corrections manuelles et ceux-ci comportent pourtant encore beaucoup de données bruitées ou incomplètes. Améliorer les résultats obtenus reste cependant largement possible à mon sens mais nécessiterait un système hybride combinant l'approche numérique et une analyse à l'aide de ressources linguistiques, voire spécifiques au domaine/métier abordé.

Malgré de nombreuses conversations avec des spécialistes du domaine du recrutement, aucun n'a été capable à ce jour de nous donner la « recette magique » qui faisait qu'il préférait choisir un candidat plutôt qu'un autre à compétence égale. Cette décision étant généralement extrêmement subjective et pas forcément identique d'un recruteur à un autre. À l'heure où le recrutement se veut de plus en plus transparent, avec l'anonymisation des candidatures, la lutte contre la discrimination à l'embauche, l'équité homme/femme, etc. les ordinateurs ne pourraient-ils pas apporter des réponses objectives à ces questions ?

Annexes

Annexe A

Récupération des informations d'une offre d'emploi

Afin de retrouver les différentes informations contenues dans une offre d'emploi, nous avons mis en place plusieurs fonctions à base d'expressions régulières :

1. **Type de contrat** : Utilisation d'expression régulière permettant de retrouver les termes cdi/cdd/stage/intérim.
2. **Durée de contrat** : Si le contrat détecté est de type /cdd/stage/intérim, nous effectuons une recherche afin de repérer une durée (jour/mois/an).
3. **Salaire** : Afin de repérer les salaires et leurs montants nous avons effectué divers traitements en cascade. Nous recherchons dans un premier temps des expressions de type "rémunération", "salaire" ainsi qu'une devise (k€, €, $,£). Une fois un segment contenant ce type d'éléments détectés, nous effectuons une recherche d'un montant en fonction de certains motifs (entre X et Y, de A à B, rémunération de Z).
4. **Lieu de travail** : nous recherchons toujours avec des expressions régulières et d'un dictionnaire, les codes postaux, les régions ou les noms de villes contenus dans l'annonce.
5. **Société** : A l'aide d'un dictionnaire contenant la liste des différents clients de la société Aktor, nous recherchons la société émettrice de l'annonce.
6. **Email** : A l'aide de plusieurs patrons permettant de détecter ce type de motif, nous recherchons l'adresse électronique de réponse à l'offre d'emploi.

Le tableau A.1 présente un exemple d'offre d'emploi avec en gras les différents champs qui ont été détectés dans l'annonce.

Créé en 1999, **AKTOR Interactive**, est aujourd'hui la principale agence de communication de recrutement dédiée exclusivement aux média Internet. La société est active dans les 5 grands pays Européens et enregistre une progression rapide de son volume d'affaire. La société recrute un :

Infographiste / Web-développeur

Missions :

Création et réalisation de supports publicitaires infographiques.

Création et réalisation d'annonce HTML et/ou annonce presse

Développement de site Internet en HTML / XML utilisant les langages Perl / PHP.

Webmastering / entretien et évolution des sites.

Participation au contenu et au travail rédactionnel

Graphiste de formation, vous maîtrisez parfaitement l'ensemble de la chaîne graphique et les principaux logiciels de PAO. Vous justifiez d'une expérience en agence de communication ou de publicité en tant que créatif. Vous êtes passionné d'Internet et avez acquis des connaissances assez solides en développement pour être autonome dans la réalisation et la maintenance de sites Internet y compris des sites dynamiques simples. Vous maîtrisez parfaitement Flash action script que vous utilisez personnellement et professionnellement. Vous avez l'esprit d'équipe et vous êtes rigoureux et précis dans la mise en œuvre de vos projets.
Envoyez-nous votre dossier de candidature comprenant CV + LM + vos références web (les dossiers sans références en ligne ne seront pas retenus) à : Rémunération annuelle : **entre 30 et 35 K€** Poste en **CDI** basé à **Lyon**.

Merci d'adresser votre candidature complète comprenant CV + LM par mail à **web58_4@mail.gestcom.biz**

TAB. *A.1* – *Exemple d'offre d'emploi.*

Annexe B

Librairie Mailviewer

La librairie Mailviewer a été créée afin d'optimiser la lecture de courriers électroniques en temps d'exécution ainsi qu'en occupation disque. Nous présentons en B.1 les principales fonctions ainsi que les points d'entrées de la librairie. L'utilisation de cette librairie a permis un gain de 50% d'espace disque et de multiplier par 4 la vitesse d'éxecution des applications de gestion de candidature de la société Aktor.

Point d'entrées	**Description**
new ()	Constructeur de la librairie Mailviewer.
InitialiseFlux()	Décompose chaque partie du courriel (en-tête/corps/pièce jointe). Elle initialise le flux de traitement du courrier électronique. Celle-ci fait appelle à la fonction TraiteSousPartie(). Elle construit par ailleurs l'objet _MailInfo à la fin.
GetNbParts()	Permet de récupérer le nombre de parties du courrier électronique.
GetMailInfo()	Permet de récupérer l'objet *_MailInfo*. On peut par ailleurs récupérer chaque information individuellement avec : -*GetName()* récupère le nom, le prénom et l'adresse électronique dans un courriel. -*GetFrom()* récupère l'expéditeur du courriel. -*GetDate()* récupère la date d'envoi du courriel. -*GetBodyContentType()* récupère le content-type du corps du message. -*GetSubjet()* récupère le sujet du mail. -*GetTo()* récupère tous les destinataires d'un message. -*GetCc()* récupère tous les destinataires en copie d'un message.
GetAllPJ()/GetPJ()	Permet de récupérer l'ensemble des pièces jointes du courrier électronique ou une pièce jointe en particulier.
GetBody()	Permet de récupérer le nombre de parties du courrier électronique.
GetNbParts()	Permet de récupérer le nombre de partie du courrier électronique.
GetMail()	Permet de récupérer un courriel sous forme de *string* en fonction du chemin passé en paramètre.
Fonction	**Description**
Initialise()	Fonction qui décompose chaque partie du courriel (en-tête/corps/pièce jointe).
TraiteSousPartie()	Fonction récursive permettant de traiter les parties de courriels composés de plusieurs parties (multipart).
SetPJ()	Fonction qui ajoute une pièce jointe dans le tableau de pièces jointes *_PJTaB*.

TAB. *B.1* – *Principales fonctions de la librairie Mailviewer.*

Annexe C

Listes des offres d'emploi des différents corpus

Le tableau C.1 présente les intitulés des différentes offres d'emploi contenues dans les corpus.

Mission	Intitulé de la mission
33725	conseiller commercial urbain h/f
34865	assistant comptable h/f
34861	ingénieur commercial(e) h/f
32854	attaché commercial h/f
33867	assistant(e) commercial(e) export h/f
31273	responsable de mission h/f
31767	assistant comptable junior h/f
33553	un délégué commercial h/f
32559	consultant logiciel de gestion h/f
33746	3 chefs de cuisine h/f
31022	assistant(e) en recrutement en contrat de qualification
32560	consultant conseil et formation h/f
34692	chef de projet produits assurance dommages h/f
33633	ingenieur commercial h/f
00015	un chargé de développement commercial (h/f)
00010	chargé(e) d'étude et de communication internationale
31274	assistant comptable junior h/f
32461	développeur Web php/mysql h/f
31702	comptable, département fournisseurs
34783	assistant comptable h/f
00014	développeur Internet php
33553	un délégué commercial h/f
34119	assistant commercial h/f
34990	collaborateur comptable h/f
35044	directeur de clientèle h/f
31248	développeur commercial h/f

TAB. *C.1* – *Intitulés des offres d'emplois par mission.*

Annexe D

Exemples d'offres d'emploi et découpages en bloc

D.1 Exemple 1

Notre groupe industriel est spécialisé dans la fabrication de produits de grande consommation et recherche dans le cadre de son développement pour son site de Moulins (Allier) un : Responsable Maintenance Industriel Vous êtes chargé d'organiser et d'optimiser la politique maintenance du site (maintenance curative et préventive), en animant une équipe de techniciens. Vous établissez et suivez les plannings d'intervention, vous supervisez la qualité des travaux réalisés selon les règles et méthodes de travail que vous contribuez à faire progresser, en vous impliquant personnellement dans la mise à niveau des connaissances de votre équipe. Vous devez impérativement apporter une valeur ajoutée à votre action et ne pas intervenir uniquement de manière réactive. A ce titre, vous participez à la mise en place de procédures afin d'anticiper les problèmes éventuels. L'ensemble de vos actions se déroulent dans le respect des règles de qualité, d'hygiène et de sécurité liées aux procédures internes et à la législation. De formation Bac + 2/5 (Type BTS MAI ou jeunes ingénieurs), vous justifiez d'une première expérience confirmée de 5 ans dans un environnement industriel idéalement à process continus et automatisés. Pour ce poste à responsabilités, il importe que votre investissement soit basé sur le long terme. Merci d'adresser votre candidature sous la Ref AB 471/028/B à Neptune Ressources Humaines 5 place louis Gavon 79001 Paris ou 21 bd MalMan 34009 AVIGNON ou Pierre.hurbi@neptuneRH.fr

TAB. *D.1* – *Premier Exemple d'offre d'emploi.*

Pour l'exemple D.1, le découpage obtenu est :

Titre

Responsable Maintenance Industriel

Description

Notre groupe industriel est spécialisé dans la fabrication de produits de grande consommation et recherche dans le cadre de son développement pour son site de Moulins (Allier) un :

Mission

Vous êtes chargé d'organiser et d'optimiser la politique maintenance du site (maintenance curative et préventive), en animant une équipe de techniciens. Vous établissez et suivez les plannings d'intervention, vous supervisez la qualité des travaux réalisés selon les règles et méthodes de travail que vous contribuez à faire progresser, en vous impliquant personnellement dans la mise à niveau des connaissances de votre équipe. Vous devez impérativement apporter une valeur ajoutée à votre action et ne pas intervenir uniquement de manière réactive. A ce titre, vous participez à la mise en place de procédures afin d'anticiper les problèmes éventuels. L'ensemble de vos actions se déroulent dans le respect des règles de qualité, d'hygiène et de sécurité liées aux procédures internes et à la législation.

Profil et contacts

De formation Bac + 2/5 (Type BTS MAI ou jeunes ingénieurs), vous justifiez d'une première expérience confirmée de 5 ans dans un environnement industriel idéalement à process continus et automatisés. Pour ce poste à responsabilités, il importe que votre investissement soit basé sur le long terme.

Merci d'adresser votre candidature sous la Ref AB 471/028/B à Neptune Ressources Humaines 5 place louis Gavon 79001 Paris ou 21 bd MalMan 34009 AVIGNON ou Pierre.hurbi@neptuneRH.fr

D.2 Exemple 2

Ce groupe français spécialisé dans la prestation d'analyses chimiques, recherche un :

RESPONSABLE DE TRANSFERT LABORATOIRE. Sud Est.

En charge du regroupement et du transfert d'activités de différents laboratoires d'analyses, vous étudiez, conduisez et mettez en oeuvre le sequencement de toutes les phases nécessaires à la réalisation de ce projet, dans le respect du budget prévisionnel et des délais fixes.

Vos solutions intègrent l'ensemble des paramètres de la démarche (social, logistique, infrastructures et matériels, informatique) et dessinent le fonctionnement du futur ensemble (Production, méthodes et accréditations, développement produit, commercial)

De formation supérieure Ecole d'ingénieur Chimiste (CPE) option chimie analytique environnementale, vous avez deja conduit un projet de transfert d'activité. La pratique de la langue anglaise est souhaitée. Merci d'adresser votre candidature sous la référence VA 11/06 par e-mail beatrice.lardon@atalan.fr

TAB. *D.2* – *Second Exemple d'offre d'emploi.*

Pour l'exemple D.2, le découpage obtenu est :

Titre

RESPONSABLE DE TRANSFERT LABORATOIRE. Sud Est.

Description

Ce groupe français spécialisé dans la prestation d'analyses chimiques, recherche un :

Mission

En charge du regroupement et du transfert d'activités de différents laboratoires d'analyses, vous étudiez, conduisez et mettez en oeuvre le sequencement de toutes les phases nécessaires à la réalisation de ce projet, dans le respect du budget prévisionnel et des délais fixes. Vos solutions intègrent l'ensemble des paramètres de la démarche (social, logistique, infrastructures et matériels, informatique) et dessinent le fonctionnement du futur ensemble (Production, méthodes et accréditations, développement produit, commercial).

Profil et contacts

De formation supérieure Ecole d'ingénieur Chimiste (CPE) option chimie analytique environnementale, vous avez deja conduit un projet de transfert d'activité. La pratique de la langue anglaise est souhaitée. Merci d'adresser votre candidature sous la référence VA 11/06 par e-mail beatrice.lardon@atalan.fr

Annexe E

DEFT 2008

La campagne DEFT 2008 (DÉfi de Fouilles de Texte) a eu pour sujet la classification en genre et en thème de textes. Le défi propose deux tâches distinctes, l'une combine classification en genre et en thème tandis que l'autre se focalise uniquement sur la classification en thème.

E.1 Corpus Tâche 1

Ce corpus, noté *CORPUS T1*, est étiqueté par thèmes selon quatre classes : économie (ECO), art (ART), télévision (TEL) et sport (SPO). Chaque document est également annoté selon qu'il provient du journal Le Monde (LM) ou de Wikipedia (W). La répartition des volumes de documents pour chaque classe dans le corpus d'apprentissage (APP1) et d'évaluation (EVAL1) est indiquée dans le tableau E.1.

CORPUS T1	ECO	TEL	ART	SPO	LM	W	Echantillons
APP1	30,41%	8,88%	37,88%	22,82%	57,97%	42,02%	15223
EVAL1	29,11%	12,57%	36,27%	21,84%	53,63%	46,36%	10696

TAB. *E.1* – *Répartition des volumes de documents du CORPUS T1 pour chaque classe.*

On observe sur le corpus d'apprentissage de cette tâche un déséquilibre important dans la répartition entre chacune des classes. Ce déséquilibre est conjugué à des imbrications inter-classes importantes[1]. L'imbrication est particulièrement marquée dans le cas des classes TEL et ART, ceci étant accentué par une faible représentation de TEL par rapport à ART (respectivement 8.88% et 37.88% des documents).

[1] Les distances entre classes ont été mesurées par similarité cosinus

E.2 Corpus Tâche 2

Le corpus de la tâche 2, noté *CORPUS T2*, est composé de documents également issus des deux sources Le Monde et Wikipédia mais est annoté uniquement en thèmes selon cinq classes : société (SOC), actualité ou information française (FRA), ou internationale (INT), sciences (SCI) et littérature (classe LIV). La répartition des classes dans CORPUS T2 est donnée dans le tableau E.2 pour les corpus d'apprentissage (APP2) et d'évaluation (EVAL2).

CORPUS T2	SOC	FRA	INT	LIV	SCI	Echantillons
APP2	16,04%	14,12%	22,52%	19,43%	27,87%	23550
EVAL2	16,03%	14,12%	22,53%	19,42%	27,87%	15693

TAB. *E.2* – *Répartition des volumes de documents du CORPUS T2 pour chaque classe.*

La répartition entre les classes de ce corpus est beaucoup plus homogène dans *CORPUS T2* que dans *CORPUS T1*.On note cependant une légère sous-représentation de la classe FRA associée à une sur-représentation de la classe SCI. Les intersections entre les classes FRA, SOC, SCI semblent assez importantes.

E.3 Pré-traitements

Nous avons appliqué les divers pré-traitements classiques tels que la suppression des mots fonctionnels (*être, avoir, pouvoir, falloir ...*), des expressions courantes (*par exemple, c'est-à-dire, chacun de ...*), de chiffres (numériques et/ou textuelles) et des symboles comme <$>, <#>,<*>, ainsi que la *stop-list* de Jean Veronis[2] et ramené a leur racine les verbes fléchis ainsi que les mots pluriels et/ou féminins au masculin singulier. Chaque document est ainsi transformé en un vecteur de mots préalablement filtrés auquel on associe une classe (genre ou catégorie) pour la phase d'apprentissage.

E.4 Classification

Les classifieurs choisis pour cette tâche ont été les SVM de LibSVM(Fan et al., 2005), ainsi que les n-grammes de mots. Nous avons donc récupéré l'ensemble des n-grammes (uni-gramme, bi-grammes et tri-grammes) pour chaque classe contenue dans nos corpus et associé une fréquence. Nous calculons par la suite pour obtenir le score $\tilde{t}$ des n-grammes pour un document D :

$$\tilde{t} = ArgMax_t(\lambda_1 P_t(W_i|W_{i-2}W_{i-1}) + \lambda_2 P_t(W_i|W_{i-1}) + \lambda_3 P_t(W_i)) \quad \text{(E.1)}$$

[2]http ://www.up.univ-mrs.fr/veronis/data/antidico.txt

Les premiers résultats obtenus par chacun des systèmes nous ont fait opter pour une solution uniquement a base de SVM, appelés par la suite **SVM_R**, celles ci s'avérant plus efficace sur cette tâche que les *n*-grammes ou une combinaison des deux. Le noyau choisi a été le noyau linéaire, qui s'est avéré le plus performant, ceci s'expliquant par la taille importante de notre lexique.

E.5 Combinaison des systèmes

Plusieurs systèmes ont été mis en place par l'équipe jeune chercheur en vue d'une fusion dans le cadre de ce défi : un classifieur Bayésien Naïf, une implémentation de Boostexter [3], Enertex(Fernandez et al., 2007), une implémentation de icsiboost[4], un système à base de Cosinus, deux système à base de SVM, **SVM_R** et un second système avec une distribution de mots préalablement filtrés, **SVM_E**. L'ensemble des Fscores obtenus par chaque système sur les trois tâches, lors de la validation croisée de la phase d'apprentissage, est présenté dans le tableau E.7.

E.6 Protocole expérimental

Pour chaque tâche, la méthode de validation des classifieurs est une validation croisée classique sur N partitions du corpus d'apprentissage. Chacun des corpus d'apprentissage de APP1 et APP2 a été divisé en N parties égales qui constituent N sous-corpus. Nous utilisons $N-1$ sous-corpus en tant que données d'apprentissage et le Nième corpus est réservé au test. N jeux de tests tournants sont ainsi créés, le score final d'une classification étant ensuite calculé par la moyenne de tous les scores obtenus sur chacun des N corpus de test.

E.7 Résultats

Les tableaux E.4 et E.6 présente le détail des résultats obtenus pour la tache 1 et 2 pour le système **SVM_R** tandis que les tableaux E.3 et E.5 présentent les différentes matrices de confusion obtenues sur chacune des tâches.

On observe de très bons résultats pour la tâche 1, malgré une baisse significative du à la classe _TEL_, celle ci se confondant avec la classe _ART_ (356 documents _TEL_ classés en _ART_). Concernant la tâche 2, on observe un manque de précision marqué sur la classe _SCO_, qui se confond avec les classes _FRA_ et _SCI_. Les principales difficultés rencontrées ont été des temps d'apprentissage longs (environ 10 heures pour

[3] http ://www.research.att.com/sw/download/

[4] icsiboost est une version *open-source* de BoosTexter développée par le laboratoire ICSI http ://www.icsi.berkeley.edu/

	SPO	_TEL_	_ART_	_ECO_
SPO	3286	38	48	47
TEL	12	830	130	57
ART	106	356	5209	375
ECO	75	128	379	4147

TAB. *E.3* – *Matrice de confusion du système **SVM_R** pour la tâche 1.*

	SPO	_TEL_	_ART_	_ECO_	(toutes classes)
Précision	0.963	0.807	0.862	0.877	0.8770
Rappel	0.946	0.614	0.903	0.896	0.8400
Fscore	0.955	0.697	0.882	0.887	0.8581

TAB. *E.4* – *Précision, Rappel Fscore du système **SVM_R** pour la tâche 1.*

	FRA	_INT_	_LIV_	_SCI_	_SOC_
FRA	2386	31	85	164	419
INT	301	4576	66	164	385
LIV	34	37	3997	221	154
SCI	96	123	260	5406	479
SOC	509	357	219	684	2339

TAB. *E.5* – *Matrice de confusion du système **SVM_R** pour la tâche 2.*

	FRA	_INT_	_LIV_	_SCI_	SOC	(toutes classes)
Précision	0.762	0.833	0.900	0.849	0.569	0.7827
Rappel	0.717	0.863	0.874	0.824	0.824	0.7795
Fscore	0.739	0.848	0.887	0.837	0.593	0.78

TAB. *E.6* – *Précision, Rappel Fscore du système **SVM_R** pour la tâche 2.*

l'apprentissage et 1h30 pour les tests) ainsi que des problèmes de mémoire dans la longueur des vecteurs pour les SVMs ceci étant du de la taille importante des corpus DEFT par rapport aux corpus traités préalablement avec ces méthodes. Le tableau E.7 résume l'ensemble des Fscores obtenus par chacun des systèmes en phase apprentissage.

Système	tâche1-catégorie	tâche1-genre	tâche2-catégorie
SVM_R	0.8581	0.93	0.78
SVM_E	**0.9150**	0.9594	**0.8445**
N_Bayes_extended	0.8469	0.9353	0.8271
BoosTexter	0.8958	**0.9869**	0.8316
icsiboost	0.9150	0.9858	0.8409
Cosine_Discriminant	0.8508	0.9222	0.8244
Enertex	0.8328	0.8390	0.7561

TAB. *E.7* – *Performances des différents systèmes lors de la phase d'apprentissage.*

Annexe F

Résultats avec le logiciel Weka

Nous présentons au tableau F.1 les résultats obtenus avec le logiciel Weka[1]. Nous avons utilisé une sous-partie du *Corpus OUI/NON* composé des missions **34861**, **00014**, **34783**, **32461** et **31767** principalement pour éviter des problèmes de mémoire avec Weka lorsque les matrices étaient trop importantes. Nous avons utilisé un système de validation croisé pour tester chaque système. On observe de très bonne performance général de chacun des systèmes sur la classe **NON** (Fscore de 0,946 ; 0,947 ; 0,931 ; 0,945), cependant les performances chutent pour la classe **OUI** (Fscore de 0,45 ; 0,318 ; 0,343 ; 0,591). Ce qui s'explique par l'observation de la matrice de confusion (tableau F.2). En effet, les 4 algorithmes classent plus de 90% des candidatures en négatif, principalement dû au déséquilibre entre les classes (24 candidatures OUI et 205 candidatures NON).

	J48		BayesNet		IB1		KStar	
	OUI	**NON**	**OUI**	**NON**	**OUI**	**NON**	**OUI**	**NON**
Précision	0,5	0,932	0,714	0,915	0,462	0,913	0,591	0,945
Rappel	0,409	0,996	0,025	0,981	0,273	0,95	0,591	0,945
Fscore	0,45	0,946	0,318	0,947	0,343	0,931	0,591	0,945

TAB. *F.1* – *Précision, Rappel, Fscore obtenus par J48, BayesNet, IB1 et KStar.*

Candidature	Matrice de confusion							
	J48		BayesNet		IB1		KStar	
	OUI	**NON**	**OUI**	**NON**	**OUI**	**NON**	**OUI**	**NON**
NON	8	191	5	194	10	189	11	188
OUI	10	14	7	17	18	6	13	9

TAB. *F.2* – *Matrice de confusion J48, BayesNet, IB1 et KStar.*

Nous avons effectué différentes séries de tests avec les algorithmes de *clustering* de

[1]Weka est un logiciel libre qui propose un ensemble d'algorithmes d'apprentissage automatique. Il permet de faire de la classification, de la régression, du *clustering* et des règles d'associations. http://www.cs.waikato.ac.nz/~ml/weka/index.html

Weka, Cobweb et SimpleKmeans avec des résultats assez similaires puisque l'on obtient entre 10% (FarthestFirst) et 40% d'erreurs (SimpleKmeans). Les matrices de confusion montrent des résultats assez similaires aux précédents avec une forte majorité classée dans la classe **NON** comme le montre le tableau F.3.

Candidature	Matrice de confusion			
	SimpleKmeans		FarthestFirst	
	OUI	**NON**	**OUI**	**NON**
NON	386	184	2	568
OUI	24	45	0	83

TAB. ***F.3 –*** *Matrice de confusion pour SimpleKmeans et FarthestFirst.*

Annexe G

Algorithme fuzzy k-means

```
Dist (*a, *b, n)
{
Calcul de distance euclidienne entre a et b (dimension n)
    return Dist
}
fonction fuzzykmeans (
X(m x n ) : Matrice d'élements
m : nb de lignes
n : nb de colonnes dimensions
means( k x n ) : centroïdes
k : nb de centroïdes
p : paramètre fuzzy
Dist(*a, *b, n) : fonction de distance
epsilon : tolérance
w(m x k) : centroïdes
)
{
it = 0 // Itérations ;
// normaliser w par ligne

for (i=0 ; i ≤ m ; i++)
    {
    sum = 0

    for (j=0 ; j ≤ k ; j++) sum += w( i , j)
    for (j=0 ; j ≤ k ; j++) w( i , j) /= sum
```

```
running = 1

while running do
    // normaliser w par colonne (par cluster)


    for (j=0 ; j ≤ k ; j++) do
        sum = 0
    for (i=0 ; i ≤ m ; i++)  sum += w( i , j)
    for (i=0 ; i ≤ m ; i++)  w( i , j) /= sum

    end

    for (l=0 ; l ≤ k ; l++) do
        for(j=0 ; j ≤ n ; j++)  sum = 0
    for(i=0 ; i ≤ m ; i++)  sum += X[i][j] * w(i, l)
    means(l, j) = sum
    oldmeans(l, j) = means(l, j)

    end

    calculer les poids w
for (i=0 ; i ≤ m ; i++) do
        sum = 0
    For(i=0 ; i ≤ k ; i++)  dtemp = Dist (X[i], means[l], n)
    w(i, l) = 1 / dtemp(p-1)
    sum += w(i, l)
    For(i=0 ; i ≤ k ; i++)  w(i, l) = w(i, l) / sum

    end

    it += 1

    running = 0
for (i=0 ; i ≤ k ; i++) do
        if Dist(oldmeans[i], means[i], n) >= epsilon  running = 1   break

    end

end
```

Liste des illustrations

Liste des tableaux

Revues internationales avec comité de lecture

(Kessler et al., 2006)

Kessler, Rémy et Torres-Moreno, Juan Manuel et El-Bèze, Marc

Classification automatique de courriers électroniques par des méthodes mixtes d'apprentissage.

Publié dans *RSTI-ISI Vol 11* , 2006

Abstract : New forms of written communication (electronic mail, forum, chat, SMS, etc.) are new challenges for Natural Language Processing methods. These data present very particular linguistic phenomena : too short and very noised messages... This paper focuses on the development of generic tools and resources for e-mails classification. The capabilities, for a company, to manage these flows of information effectively and at low cost, are becoming a major stake for customers satisfaction. This study deals with the problems of the precise routing of e-mails. After a filtering and lemmatization step, vectorial representation of texts is used for classification purpose by means of supervised, semi-supervised and unsupervised learning techniques. This study shows that semi-supervised initialization optimizes learning. Very good results are presented on realistic corpora.

Communications internationales avec actes

(Kessler et al., 2009a)

Job Offer Management : How Improve the Ranking of Candidates

Kessler, Rémy et Nicolas, Béchet et Torres-Moreno, Juan Manuel et Mathieu, Roche et El-Bèze, Marc

Publié dans *International Symposium on Methodologies for Intelligent Systems* (ISMIS) 2009, Prague, à paraître

Abstract : *The market of online job search sites grows exponentially. This implies volumes of information (mostly in the form of free text) become manually impossible to process. An analysis and assisted categorization seems relevant to address this issue. We present E-Gen, a system which aims to perform assisted analysis and categorization of job offers and of the responses of candidates. This paper presents several strategies based on vectorial and probabilistic models to solve the problem of profiling applications according to a specific job offer. We have evaluated*

a range of measures of similarity to rank candidatures by using ROC curves. Relevance feedback approach allows to surpass our previous results on this task, difficult, diverse and higly subjective.

(Kessler et al., 2009b)

Profilage de candidatures assisté par Relevance Feedback

Kessler, Rémy et Nicolas, Béchet et Torres-Moreno, Juan Manuel et Mathieu, Roche et El-Bèze, Marc

Publié dans *Traitement Automatique des Langues Naturelles* (TALN) 2009

Abstract : *Le marché d'offres d'emploi et des candidatures sur Internet connaît une croissance exponentielle. Ceci implique des volumes d'information (majoritairement sous la forme de texte libre) qu'il n'est plus possible de traiter manuellement. Une analyse et catégorisation assistées nous semble pertinente en réponse à cette problématique. Nous proposons E-Gen, système qui a pour but l'analyse et catégorisation assistés d'offres d'emploi et des réponses des candidats. Dans cet article nous présentons plusieurs stratégies, reposant sur les modèles vectoriel et probabiliste, afin de résoudre la problématique du profilage des candidatures en fonction d'une offre précise. Nous avons évalué une palette de mesures de similarité afin d'effectuer un classement pertinent des candidatures au moyen des courbes ROC. L'utilisation d'une forme de relevance feedback a permis de surpasser nos résultats sur ce problème difficile et sujet à une grande subjectivité.*

(Kessler et al., 2008)

E-Gen : Profilage automatique de candidatures

Kessler, Rémy et Torres-Moreno, Juan Manuel et El-Bèze, Marc

Publié dans *Traitement Automatique des Langues Naturelles* (TALN) 2008

Abstract : *The exponential growth of the Internet has allowed the development of a market of on-line job search sites. This paper presents the E-Gen system (Automatic Job Offer Processing system for Human Resources). E-Gen will perform two complex tasks : an analysis and categorisation of job postings, which are unstructured text documents, an analysis and a relevance ranking of the candidate answers (cover letter and curriculum vitae). Here we present the work related to the second task : we use vectorial representation before generating a classification with SVM to determine the type of the attachment. In the next step, we try to classify the candidate answers with different classifiers (SVM and ngrams of words).*

(Kessler et El-Bèze, 2008)

Kessler, Rémy et El-Bèze, Marc

E-Gen : traitement automatique des offres d'emploi.

Publié dans Actes de Journées internationales d'Analyse statistique des Données Textuelles JADT 2008

Abstract : *The exponential growth of the Internet has allowed the development of a market of on-line job search sites. This paper aims at presenting the E-Gen system (Automatic Job Offer Processing system for Human Resources). EGen will implement two complex tasks : an analysis and categorisation of job postings, which are unstructured text documents (e-mails of job listings possibly with an attached document), an analysis and a relevance ranking of the candidate answers (cover letter and curriculum vitae). This paper aims to present a strategy to resolve the first task : after a process of filtering and lemmatisation, we use vectorial representation before generating a classification with Support Vector Machines and n-grams of words. This first classification is afterwards transmitted to a "corrective" post-process (with Markov model and a Branch&Bound algorithm for pruning the tree) which improves the quality of the solution.*

(Kessler et al., 2007)

E-Gen : Automatic Job Offer Processing system for Human Ressources

Kessler, Rémy et Torres-Moreno, Juan Manuel et El-Bèze, Marc

Publié dans *6th Mexican International Conference on Artificial Intelligence* (MICAI) 2007

Abstract : *The exponential growth of Internet allowed the development of a market of online job search sites. This paper aims at presenting the E-Gen system (Automatic Job Offer Processing system for Human Ressources). E-Gen will implement several complex tasks : an analysis and categorisation of jobs posting which are unstructured text documents (e-mails of job listings possibly with an attached document), an analysis and a relevance ranking of the candidate answers (cover letter and curriculum vitae). This paper aims to present a strategy to resolve the first task : after a process of filtering and lemmatisation, we use vectorial representation before generating a classification with Support Vector Machines. This first classification is afterwards transmitted to a "corrective" post-process which improves the quality of the solution.*

(Kessler et al., 2004b)

Classification thématique de courriels avec apprentissage supervisé, semi-supervisé et non supervisé

Kessler, Rémy et Torres-Moreno, Juan Manuel et El-Bèze, Marc

Publié dans *Veille Strategique Scientifique & Technologique* (VSST) 2004

Résumé : *Les nouvelles formes de communication écrite (courriers électroniques, forums, chats, SMS, etc.) présentent des défis considérables pour leur traitement automatique. Nous présentons des recherches destinées à créer des outils et des ressources génériques pour la classification de courriels. La capacité d'une entreprise de gérer efficacement, rapidement et à moindre coût, ces flux d'informations devient un enjeu majeur pour la satisfaction des clients. Ceci nécessite, en particulier, de disposer d'outils informatiques permettant notamment le routage pour acheminer*

les courriels vers le destinataire concerné et l'automatisation de réponses. Nous nous attachons à traiter dans cette étude des problèmes posés par le routage précis de courriels. Après un processus puissant de filtrage et de lemmatisation, nous utilisons la représentation vectorielle de textes avant d'effectuer la classification par des approches supervisées, semi supervisées et non supervisées. Nous avons trouvé, par ailleurs, une initialisation semi supervisée qui optimise l'apprentissage non supervisé. Lors des tests préliminaires, nous avons obtenu de très bonnes performances sur des corpus réalistes.

Workshop

(Kessler et al., 2008)

Kessler, Rémy et Nicolas, Béchet et Torres-Moreno, Juan Manuel et Mathieu, Roche et El-Bèze, Marc

Automatic Profiling System for Ranking Candidates Answers in Human Resources

Publié dans *On The Move* (OTM '08)

Abstract : *The exponential growth of Internet allowed the development of a market of online job search sites. This work aims at presenting the E-Gen system (Automatic Job Offer Processing system for Human Resources). E-Gen will implement several complex tasks : an analysis and categorization of jobs offers which are unstructured text documents (e-mails of job offers possibly with an attached document), an analysis and a relevance ranking of the candidate answers. We present a strategy to resolve the last task : After a process of filtering and lemmatisation, we use vectorial representation and different similarity measures. The quality of ranking obtained is evaluated using ROC curves.*

Ateliers

(Charton et al.)

Eric Charton, Nathalie Camelin, Rodrigo Acuna-Agost, Pierre Gotab, Remi Lavalley, Remy Kessler et Silvia Fernández

Pré-traitements classiques ou par analyse distributionnelle : application aux méthodes de classification automatique déployées pour DEFT08

Publié dans *les actes d'atelier DÉfi Fouille de Textes (DEFT)* 2008

Abstract : *In this paper we describe a set of automatic classification methods applied to the DEFT08 campaign. First, we evaluated and compared some of state-of-the-art classifiers like SVM, AdaBoost, probabilistic-based classifiers, and cosine-based classifiers. Subsequently, we developed a method to normalize classes using a distributional analysis of the text with the aim*

of improving the performance. Lastly, some additional results were obtained by two merging methods that showed to increase the scores of the individual classifiers.

Autres publications

(Kessler et al., 2004a)

Kessler, Rémy et Torres-Moreno, Juan Manuel et El-Bèze, Marc

Classification thématique de courriels

Soumis à *Journée ATALA sur les nouvelles formes de communication écrite*

Résumé : *Les nouvelles formes de communication écrite (courriers électroniques, forums, chats, SMS, etc.) présentent des défis considérables pour leur traitement automatique. Nous présentons des recherches destinées à créer des outils et des ressources génériques pour la classification de courriels. La capacité d'une entreprise de gérer efficacement, rapidement et à moindre coût, ces flux d'informations devient un enjeu majeur pour la satisfaction des clients. Ceci nécessite, en particulier, de disposer d'outils informatiques permettant notamment le routage pour acheminer les courriels vers le destinataire concerné et l'automatisation de réponses. Nous nous attachons à traiter dans cette étude des problèmes posés par le routage précis de courriels. Après un processus puissant de filtrage et de lemmatisation, nous utilisons la représentation vectorielle de textes avant d'effectuer la classification par des approches supervisées, semi- supervisées et non supervisées. Nous avons trouvé, par ailleurs, une initialisation semi-supervisée qui optimise l'apprentissage non supervisé. Lors des tests préliminaires, nous avons obtenu de très bonnes performances sur des corpus réalistes.*

Bibliographie

(Abney, 2008) S. P. Abney, 2008. *Semisupervised learning in computational linguistics*. Chapman & Hall/CRC.

(Allen et Pilot, 2001) C. Allen et L. Pilot, 2001. HR-XML : Enabling pervasive HR-e-Business. Dans les actes de *XML Europe 2001, Int. Congress Centrum (ICC)*.

(Amadieu, 2007) J.-F. Amadieu, 2007. Synthèse du test du recrutement réalisé à la demande de la HALDE. *Adia/Paris I, Observatoire des discriminations*, 67–78.

(Amini, 2001) M. R. Amini, 2001. *Modèles Dynamiques d'Apprentissage Numérique pour l'Accès à l'Information Textuelle*. Thèse de doctorat, Université de Paris 6.

(Amini, 2007) M. R. Amini, 2007. *Apprentissage de Fonctions de Classification et d'Ordonnancement avec des Données Partiellement Étiquetées*. Habilitation à diriger des recherches, Université de Paris 6.

(Audras et Ganascia, 2006) I. Audras et J.-G. Ganascia, 2006. Apprentissage du français langue étrangère et TALN : Analyses de corpus écrits à l'aide d'outils d'extraction automatique du langage. Dans les actes de *8èmes Journées d'Analyse de Données Textuelles (JADT 06)*, 67–78. J.-M. Viprey Ed., Univ. de Franche Compté, Besançon 2006.

(Autor, 2001) D. H. Autor, 2001. Wiring the Labor Market. *Journal of Economic Perspectives 15*(1), 25–40.

(Bachimont, 2000) B. Bachimont, 2000. Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances. *Ingénierie des connaissances, évolutions récentes et nouveaux défis, Paris : Eyrolles, Collection Technique et Scientifique des Télécommunications*, 305–323.

(Banter, 2001) I. Banter, 2001. Natural Language Engines for Advanced Customer Interaction.

(Baum et Egon, 1967) L. E. Baum et J. A. Egon, 1967. An inequality with Applications to Statistical Estimation for Probabilistic Functions of a Markov process and to a Model for Ecology. *Bull. Amer. Meterol. Soc. 73*, 360–363.

(Baum et Petrie, 1966) L. E. Baum et T. Petrie, 1966. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *Annual Mathematical Statistics 37*, 1554–1563.

(Béchet et al., 2000) F. Béchet, A. Nasr, et F. Genet, 2000. Tagging unknown proper names using decision trees. Dans les actes de *38th Annual Meeting of the Association for Computational Linguistics*, Hong-Kong, 77–84.

(Beauchene et al.,) D. Beauchene, C. Million-Rousseau, et C. Rieu. Détection automatique de l'insatisfation du client dans un contexte de commerce électronique. Dans les actes de *Colloque International sur la Fouille de Textes dans le cadre de la Conférence Fédérative sur le document (CFD'2002)*.

(Beauregard, 2001) S. Beauregard, 2001. Génération de texte dans le cadre d'un système de réponse automatique à des courriels.

(Beauvallet et al.,) G. Beauvallet, M.-C. Le Garff, A.-L. Negri, et F. Cara. L'usage d'Internet par les demandeurs d'emploi. *Revue de l'IRES - numéro spécial : Internet, recrutement et recherche d'emploi*.

(Bellot et El-Bèze, 2001) P. Bellot et M. El-Bèze, 2001. Classification et segmentation de textes par arbres de décision. Dans les actes de *Technique et Science Informatiques (TSI)*, Volume 20, 107–134. Hermès.

(Bernstein et al., 2005) A. Bernstein, E. Kaufmann, C. Kiefer, et C. Bürki, 2005. SimPack : A generic java library for similarity measures in ontologies. Rapport technique, University of Zurich.

(Bezdek, 1981) J. C. Bezdek, 1981. *Pattern Recognition With Fuzzy Objective Function Algorithms.* Norwell, MA, USA : Kluwer Academic Publishers.

(Billerbeck et Zobel, 2006) B. Billerbeck et J. Zobel, 2006. Efficient query expansion with auxiliary data structures. *Inf. Syst.* (7), 573–584.

(Biskri et Meunier, 2002) I. Biskri et J. Meunier, 2002. SATIM : Système d'Analyse et de Traitement de l'Information Multidimensionnelle. Dans les actes de *6èmes Journées d'Analyse de Données Textuelles (JADT 06)*, 67–78. A. Morin and P. Sébillot Eds., IRISA Université Rennes 1, St Malo 2006.

(Bizer et Rainer, 2005) R. H. Bizer et E. Rainer, 2005. Impact of Semantic Web on the job recruitment Process. *International Conference Wirtschaftsinformatik*.

(Bélanger et Lapalme, 2004) L. Bélanger et G. Lapalme, 2004. Identification de questions pour traiter les courriels par une méthode question-réponse. Dans les actes de *7èmes Journées d'Analyse de Données Textuelles (JADT 06)*, 128–135. Presses universitaires de Louvain Ed., Louvain La Neuve, Belgique 2004.

(Blum et Mitchell, 1998) A. Blum et T. Mitchell, 1998. Combining labeled and unlabeled data with co-training. Dans les actes de *COLT' 98 : Proceedings of the eleventh annual conference on Computational learning theory*, New York, NY, USA, 92–100. ACM.

(Bondu et al., 2007) A. Bondu, V. Lemaire, et B. Poulain, 2007. Apprentissage actif d'émotions dans les dialogues Homme-machine. Dans M. Noirhomme-Fraiture et G. Venturini (Eds.), *5èmes Journées Extraction et Gestion des Connaissances (EGC)*, 427–432. Cépaduès-Éditions, Namur Belgique 2007.

(Boudin et al., 2008) F. Boudin, J. Torres-Moreno, et P. Velazguez-Morales, 2008. An Efficient Statistical Approach for Automatic Organic Chemistry Summarization. *Lecture Notes in Computer Science 5221*, 89–99.

(Boudin et Torres Moreno, 2007) F. Boudin et J. M. Torres Moreno, 2007. NEOCORTEX : A performant user-oriented multi-document summarization system. Dans les actes de *CICLing*, 551–562.

(Bourigault, 2002) D. Bourigault, 2002. UPERY : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. Dans les actes de *9ème conférence annuelle sur le Traitement Automatique des Langages Naturelles (TALN 2002)*, Nancy, France, 75–84.

(Bourigault et Aussenac-Gilles, 2003) D. Bourigault et N. Aussenac-Gilles, 2003. Construction d'ontologies à partir de textes. Dans les actes de *10ème conférence annuelle sur le Traitement Automatique des Langages Naturelles (TALN 2003)*, Volume T2, Batz-sur-Mer, France, 27–50. Université de Nantes.

(Bourigault et al., 2005) D. Bourigault, C. Fabre, C. Frérot, M. Jacques, et S. Ozdowska, 2005. Syntex, analyseur syntaxique de corpus. Dans les actes de *TALN 2005*, Volume 2, 17–20.

(Bourse et al., 2004) M. Bourse, M. Leclère, E. Morin, et F. Trichet, 2004. Human resource management and semantic Web technologies. Dans les actes de *ICTTA*, 641–642.

(Brutlag et Meek, 2000) J. Brutlag et C. Meek, 2000. Challenges of the email domain for text classification. Dans les actes de *Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA, 103–110.

(Cazalens et Lamarre, 2001) S. Cazalens et P. Lamarre, 2001. An organization of Internet agents based on a hierarchy of information domains. Dans Y. D. . F. J. Garijo (Ed.), *10th European Workshop on Multi-Agent Systems, MAAMAW 2001*, Annency, France, 12–27.

(Charton et al.,) E. Charton, N. Camelin, R. Acuna-Agost, P. Gotab, R. Lavalley, K. R., et S. Fernández. Pré-traitements classiques ou par analyse distributionnelle : application aux méthodes de classification automatique déployées pour DEFT08.

(Chauché et al., 2003) J. Chauché, V. Prince, S. Jaillet, et M. Teisseire, 2003. « Classification automatique de textes à partir de leur analyse syntaxico-sémantique». Dans les actes de *Proceedings of TALN*, Volume 1, 45–55.

(Clark et Niblett, 1989) P. Clark et T. Niblett, 1989. The CN2 induction algorithm. *Machine Learning 3*(4), 261–283.

(Clech et Zighed, 2003) J. Clech et D. A. Zighed, 2003. Data mining et analyse des CV : une expérience et des perspectives. Dans les actes de *Extraction et la Gestion des Connaissances, EGC'03*, Lyon, France, 189–200.

(Cohen, 1996a) W. Cohen, 1996a. Learning to classify English text with ILP methods. Dans L. De Raedt (Ed.), *Advances in Inductive Logic Programming*, 124–143. IOS Press.

(Cohen, 1996b) W. W. Cohen, 1996b. Learning rules that classify email. Dans les actes de *AAAI Spring Simposium on Machine Learning in Information Access*, Minneapolis, Min., USA, 18–25.

(Collobert et Bengio, 2000) R. Collobert et S. Bengio, 2000. On the convergence of SVMTorch, an algorithm for large-scale regression problems. Technical Report IDIAP-RR 00-24, IDIAP, Martigny, Switzerland.

(Daille, 2003) B. Daille, 2003. Conceptual structuring through term variations. *Proceedings ACL 2003 Workshop on Multiword Expressions : Analysis, Acquisition and Treatment 18*, 9–16.

(Damashek, 1995) M. Damashek, 1995. Gauging similarity with n-grams : Language-independent categorization of text. *Science 267*(5199), 843–848.

(David et Plante, 1990) S. David et P. Plante, 1990. De la nécessité d'une approche morpho-syntaxique dans l'analyse de textes. *Intelligence Artificielle et Sciences Cognitives au Québec 3*, 140–154.

(deGruijter et McBratney, 1988) J. J. deGruijter et A. B. McBratney, 1988. A modified fuzzy k means for predictive classification. Dans H. Bock (Ed.), *Classification and Related Methods of Data Analysis*, Amsterdam, 97–104. Elsevier Science.

(Desmontils et al., 2002) E. Desmontils, C. Jacquin, et E. Morin, 2002. Indexation sémantique de documents sur le Web : application aux ressources humaines. Dans les actes de *Journées de l'AS-CNRS Web Sémantique*.

(Dorn et Naz, 2007) J. Dorn et T. Naz, 2007. Meta-search in human resource management. Dans les actes de *Proceedings of 4th International Conference on Knowledge Systems ICKS'07*, Bangkok,Thailand, 105 – 110.

(Dorn et al., 2007) J. Dorn, T. Naz, et M. Pichlmair, 2007. Ontology development for human resource management. Dans les actes de *4th International Conference on Knowledge Management*, Vienne, 109–120.

(El-Bèze et al., 2007) M. El-Bèze, J. Torres-Moreno, et F. Béchet, 2007. Un duel probabiliste pour départager deux Présidents. *Revue des Nouvelles Technologies de l'Information - RNTI-E10*, 1889–1918.

(Enguehard et Pantera, 1995) C. Enguehard et L. Pantera, 1995. Automatic natural acquisition of a terminology. *Journal of Quantitative Linguistics 2*(1), 27–32.

(Fan et al., 2005) R.-E. Fan, P.-H. Chen, et C.-J. Lin, 2005. Working set selection using the second order information for training SVM. *6*, 1889–1918.

(Fernandez et al., 2007) S. Fernandez, E. SanJuan, et J. M. Torres-Moreno, 2007. Textual Energy of Associative Memories : performants applications of ENERTEX algorithm

in text summarization and topic segmentation. Dans A. Gelbukh et A. K. Morales (Eds.), *MICAI 2007 : Advances in Artificial Intelligence*, Volume 4827, Aguascalientes, Mexico, 861–871. Springer Berlin / Heidelberg.

(Ferri et al., 2002) C. Ferri, P. Flach, et J. Hernandez-Orallo, 2002. Learning decision trees using the area under the ROC curve. Dans les actes de *19th International Conference on Machine Learning, ICML'02*, 139–146.

(Fisher, 1987) D. Fisher, 1987. Knowledge acquisition via incremental conceptual clustering. *Machine Learning 2*(2), 139–172.

(Fondeur, 2006) Y. Fondeur, 2006. Internet, recrutement et recherche d'emploi : une introduction. *Revue de l'IRES - numéro spécial : Internet, recrutement et recherche d'emploi 3*(52), 3–10.

(Frakes et Baeza-Yates, 1992) Frakes, W. B. et R. Baeza-Yates (Eds.), 1992. *Information retrieval : data structures and algorithms*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc.

(Fred et Jain, 2003) A. Fred et A. Jain, 2003. Robust data clustering. Dans les actes de *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Volume 2, 128–133.

(Freud et Schapire, 1997) Y. Freud et R. Schapire, 1997. A decision-theoretic generalization of on-line learning. *Computer and System Sience 55*, 119–139.

(Goutte et Gaussier, 2005) C. Goutte et E. Gaussier, 2005. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. Dans les actes de *ECIR*, 345–359.

(Grilheres et al., 2004) B. Grilheres, S. Brunessaux, et P. Leray, 2004. Combining classifiers for harmful document filtering. Dans les actes de *Recherche d'Information Assistée par Ordinateur*, Avignon, 173–185.

(Huang et al., 2005) J. Huang, M. Ng, H. Rong, et Z. Li, 2005. Automated variable weighting in k-means type clustering. *IEEE Trans. on PAMI 27*(5), 657–668.

(Hurault-Plantet et al.,) M. Hurault-Plantet, M. Jardino, et G. Illlouz. Modèles de langage n-grammes et segmentation thématique pour une tâche de filtrage de textes. Volume 2.

(Illouz, 2000) G. Illouz, 2000. Vers un apprentissage en TALN dépendant du type de Texte. Dans les actes de *Actes de la 7ème conférence annuelle sur le Traitement Automatique des Langues Naturelles TALN'2000*, 16–18.

(Jain et al., 1999) A. K. Jain, M. N. Murty, et P. J. Flynn, 1999. Data clustering : a review. *ACM Comput. Surv. 31*(3), 264–323.

(Jalam et Chauchat, 2002) R. Jalam et J.-H. Chauchat, 2002. Pourquoi les n-grammes permettent de classer des textes ? Recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques. Dans A. Morin et P. Sébillot (Eds.), *6èmes Journées internationales d'Analyse statistique des Données Textuelles, JADT 2002*, St-Malo, France, 381–390.

(Joachims, 1997) T. Joachims, 1997. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Dans les actes de *ICML '97 : Proceedings of the Fourteenth International Conference on Machine Learning*, San Francisco, CA, USA, 143–151. Morgan Kaufmann Publishers Inc.

(Joachims, 1998) T. Joachims, 1998. Text categorization with support vector machines : Learning with many relevant features. Dans les actes de *10th European Conference on Machine Learning*, Chemnitz, Germany, 137–142.

(Julio et Gil-Lafuente, 2009) R. M. Julio et J. Gil-Lafuente, 2009. The signing of a professional athlete reducing uncertainty with a weighted mean hemimetric for fuzzy subsets. *ICEIS 2009 - Proceedings of the Tenth International Conference on Enterprise Information Systems, à paraître*.

(Juola, 1998) P. Juola, 1998. Cross-entropy and linguistic typology. *Proceedings of New Methods in Language Processing 3*, 843–848.

(Jupiter, 2000) C. Jupiter, 2000. E-mail Customer Service : Taking Control of Rising Customer Demand.

(Kerloch et Gallinari, 2005) F. Kerloch et P. Gallinari, 2005. Extraction d'information à partir de modèles de Markov cachés. Dans les actes de *Dans les actes de la conférence" Traitement Automatique des Langues Naturelles"(TALN 2005)-Atelier DEFT'05*, Volume 2, 145–153.

(Kessler et al., 2008) R. Kessler, N. Béchet, M. Roche, M. El-Bèze, et J. M. Torres-Moreno, 2008. Automatic profiling system for ranking candidates answers in human resources. Dans les actes de *OTM '08 in Monterrey, Mexico*, 625–634.

(Kessler et al., 2009a) R. Kessler, N. Béchet, M. Roche, M. El-Bèze, et J. M. Torres-Moreno, 2009a. Job Offer Management : How Improve the Ranking of Candidates. *ISMIS 2009, Prague, à paraître*.

(Kessler et al., 2009b) R. Kessler, N. Béchet, M. Roche, M. El-Bèze, et J. M. Torres-Moreno, 2009b. Profilage de candidatures assisté par Relevance Feedback. *TALN 2009, Senlis, à paraître*.

(Kessler et El-Bèze, 2008) R. Kessler et M. El-Bèze, 2008. E-Gen : traitement automatique des offres d'emploi. *JADT2008*, 591–601.

(Kessler et al., 2004a) R. Kessler, J. M. Torres-Moreno, et M. El-Bèze, 2004a. Classification thématique de courriels. Dans les actes de *Journée ATALA sur les nouvelles formes de communication écrite*, Paris. ATALA, http ://www.up.univ-mrs.fr/veronis/jenfce/resumes.html.

(Kessler et al., 2004b) R. Kessler, J. M. Torres-Moreno, et M. El-Bèze, 2004b. Classification thématique de courriels avec apprentissage supervisé, semi-supervisé et non supervisé. Dans les actes de *VSST 2004*, Volume B, Toulouse, 493–504.

(Kessler et al., 2006) R. Kessler, J. M. Torres-Moreno, et M. El-Bèze, 2006. Classification automatique de courriers électroniques par des méthodes mixtes d'apprentissage. 93–112.

(Kessler et al., 2007) R. Kessler, J. M. Torres-Moreno, et M. El-Bèze, 2007. E-Gen : Automatic Job Offer Processing system for Human Ressources. *MICAI 2007, Aguscalientes, Mexique, pp 985-995*.

(Kessler et al., 2008) R. Kessler, J. M. Torres-Moreno, et M. El-Bèze, 2008. E-Gen : Profilage automatique de candidatures. *TALN 2008, Avignon, France*, 370–379.

(Khmelev et Tweedie, 2002) D. Khmelev et F. Tweedie, 2002. Using markov chains for identification of writers. *Literary and Linguistic Computing 16*(4), 299–307.

(Kiritchenko et Matwin, 2001a) S. Kiritchenko et S. Matwin, 2001a. Email classification with co-training. Dans les actes de *CASCON '01 : Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*, 8. IBM Press.

(Kiritchenko et Matwin, 2001b) S. Kiritchenko et S. Matwin, 2001b. Email classification with co-training. Dans les actes de *Conference of the Centre for Advanced Studies on Collaborative research*, Toronto, Ontario, Canada.

(Knouse, 1988) S. B. Knouse, 1988. Impression management in the resume and its cover letter. *Journal of Business and Psychology*, 242–249.

(Kosseim et al., 2001) L. Kosseim, S. Beauregard, et G. Lapalme, 2001. Using information extraction and natural language generation to answer E-mail. *Lecture Notes in Computer Science 1959*, 152–163.

(Kosseim et Lapalme, 2001) L. Kosseim et G. Lapalme, 2001. Critères de sélection d'une approche pour le suivi automatique du courriel. Dans les actes de *8ème conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN)*, Tours, France, 357–371.

(Lamontagne, 2004) L. Lamontagne, 2004. *Une Approche CBR textuel de réponse au courrier électronique*. Université de Montréal.

(Land et Doig, 1960) A. H. Land et A. G. Doig, 1960. An Automatic Method of Solving Discrete Programming Problems. *Econometrica 28*, 497–520.

(Laszlo et Mukherjee, 2006) M. Laszlo et S. Mukherjee, 2006. A genetic algorithm using hyper-quadtrees for low-dimensional k-means clustering. *IEEE Trans. PAMI, 28(4)*, 533–543.

(Lelu et al., 1998) A. Lelu, M. Halleb, et B. Delprat, 1998. Recherche d'information et cartographie dans des corpus textuels à partir des fréquences de n-grammes. Dans

S. Mellet (Ed.), *4èmes Journées internationales d'Analyse statistique des Données Textuelles (JADT 1998)*, Sophia Antipolis, France, 391–400.

(Leuski, 2000) A. Leuski, 2000. Relevance and reinforcement in interactive browsing. Dans les actes de *In Proceedings of Ninth International Conference on Information and Knowledge Management (CIKM'00*, 119–126.

(Lewis et al., 1996) D. D. Lewis, R. E. Schapire, J. P. Callan, et R. Papka, 1996. SIGIR'96. Training algorithms for linear text classifiers.

(MacQueen, 1967) J. MacQueen, 1967. Some methods for classification and analysis of multivariate observations. Volume 1, 281–297. University of California Press.

(Maes, 1994) P. Maes, 1994. Agents that reduce work and information overload. *Communications ACM 37*(7), 30–40.

(Manning et Schütze, 1999) D. Manning et H. Schütze, 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

(Mayfield et McNamee, 1998) J. Mayfield et P. McNamee, 1998. Indexing using both n-grams and words. *NIST Special Publication 500-242 : TREC7*, 419–424.

(Mccallum, 1998) A. Mccallum, 1998. A comparison of event models for naive bayes text classification. Dans les actes de *In AAAI-98 Workshop on Learning for Text Categorization*, 41–48. AAAI Press.

(Mellet, 2006) K. Mellet, 2006. Sésame, ouvre-toi ! Analyse des données d'usage d'un moteur de recherche d'annonces d'offres d'emploi : www.keljob.com. *Revue de l'IRES - numéro spécial : Internet, recrutement et recherche d'emploi 3*(52), 71–100.

(Miller et al., 2000) E. Miller, D. Shen, J. Liu, et C. Nicholas, 2000. Performance and scalability of a large-scale n-gram based information retrieval system. *Journal of Digital Information 1*(5), 1–25.

(Mocho et al., 2006) M. Mocho, E. Paslaru, et B. Simperl, 2006. Practical Guidelines for Building Semantic eRecruitment Applications.

(Morin et al., 2004) E. Morin, M. Leclère, et F. Trichet, 2004. The semantic Web in e-recruitment. Dans les actes de *First European Symposium of Semantic Web (ESWS'2004)*.

(Needleman et Wunsch, 1970) S. Needleman et C. Wunsch, 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology 48*(1), 443–453.

(Payne et Edwards, 1997) T. R. Payne et P. Edwards, 1997. Interface agents that learn : An investigation of learning issues in a mail agent interface. *Applied Artificial Intelligence 11*(1), 1–32.

(Pincemin, 2002) B. Pincemin, 2002. « Similarités texte-textes. Expérience d'une application de diffusion ciblée et proposition». *Matemáticas y Tratamiento de Corpus, Séminaire interlatin de linguistique appliquée*, 35–52.

(Quilan, 1993) J. Quilan, 1993. C4.5 : Programs for machine learning. Dans les actes de *Kaufmann, San Mateo, CA*.

(Quinlan, 1993) R. J. Quinlan, 1993. *C4.5 : Programs for Machine Learning*. San Francisco, CA, USA : Morgan Kaufmann Publishers, Inc.

(Rafter et al., 2000a) R. Rafter, K. Bradley, et B. Smyth, 2000a. Automated Collaborative Filtering Applications for Online Recruitment Services. *1892*, 363–368.

(Rafter et Smyth, 2000) R. Rafter et B. Smyth, 2000. Passive Profiling from Server Logs in an Online Recruitment Environment.

(Rafter et al., 2000b) R. Rafter, B. Smyth, et K. Bradley, 2000b. Inferring Relevance Feedback from Server Logs : A Case Study in Online Recruitment.

(Rajman et Lebart, 1998) M. Rajman et L. Lebart, 1998. Similarités pour données textuelles. Dans les actes de *4th International Conference on Statistical Analysis of Textual Data (JADT'98)*, 545–555.

(Robertson et al., 1994) S. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, et M. Gatford, 1994. Okapi at TREC-3. *NIST Special Publication 500-225 : TREC3*, 109–126.

(Rocchio, 1971) J. Rocchio, 1971. *Relevance Feedback in Information Retrieval*, 313–323.

(Roche et Kodratoff, 2006) M. Roche et Y. Kodratoff, 2006. Pruning Terminology Extracted from a Specialized Corpus for CV Ontology Acquisition. Dans les actes de *OTM'06, Montpellier, France*, 1107–1116.

(Roche et Prince, 2008) M. Roche et V. Prince, 2008. Evaluation et détermination de la pertinence pour des syntagmes candidats à la collocation. Dans les actes de *JADT'08 (Journées internationales d'Analyse statistique des Données Textuelles)*, Volume 2, Lyon, France, 1009–1020.

(Rosenblatt, 1958) F. Rosenblatt, 1958. The Perceptron : A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review 55*, 386–408.

(Sahami, 1999) M. Sahami, 1999. *Using Machine Learning to Improve Information Access*. Thèse de doctorat, Computer Science Department, Stanford University.

(Salton, 1991) G. Salton, 1991. Developments in automatic text retrieval. *Science 253*(5023), 974–980.

(Salton et Buckley, 1990) G. Salton et C. Buckley, 1990. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science 41*(4), 288–297.

(Salton et McGill, 1986) G. Salton et M. J. McGill, 1986. *Introduction to Modern Information Retrieval*. New York, NY, USA : McGraw-Hill, Inc.

(Schapire et Singer, 2000) R. Schapire et Y. Singer, 2000. BoosTexter : A boosting-based system for text categorization. *Machine learning 39*(2), 135–168.

(Schmid, 1994) G. Schmid, 1994. Treetagger - a language independent part-of-speech tagger. Dans les actes de *EACL-SIGDAT 1995*, Dublin, Ireland, 44–49.

(Segal et Kephart, 1999) R. B. Segal et J. O. Kephart, 1999. MailCat : An intelligent assistant for organizing E-mail. Dans les actes de *Third International Conference on Autonomous Agents (AGENTS'99)*, New York, NY, USA, 276–282. ACM Press.

(Segal et Kephart, 2000) R. B. Segal et J. O. Kephart, 2000. Incremental learning in SwiftFile. Dans P. L. Ed. (Ed.), *Seventh International Conference on Machine Learning*, Standford, CA, USA, 863–870. Morgan Kaufmann.

(Sigaud, 2007) O. Sigaud, 2007. Les systèmes de classeurs : un état de l'art. *Revue d'Intelligence Artificielle 21*, 75–106.

(Slodzian, 2000) M. Slodzian, 2000. L'émergence d'une terminologie textuelle et le retour du sens. *Le sens en terminologie*.

(Smadja, 1993) F. Smadja, 1993. Retrieving collocations from text : Xtract. *Computational Linguistics 19*, 143–177.

(Spärck, 1970) J. K. Spärck, 1970. Some thoughts on classification for retrieval. *Journal of Documentation 26*, 89–101.

(Stanfill et Waltz, 1986) C. Stanfill et D. Waltz, 1986. Toward memory-based reasoning. *Commun. ACM 29*(12), 1213–1228.

(Teahan, 2000) W. Teahan, 2000. Text classification and segmentation using minimum cross-entropy. Dans J. Mariani et D. H. Eds. (Eds.), *RIAO 2000*, Paris, France, 943–961.

(Teytaud et Jalam,) O. Teytaud et R. Jalam. Kernel-based text categorization. Dans les actes de *IJCNN'01, 12th International Joint Conference on Neural Networks*, 1892–1897.

(Tolksdorf et al., 2006) R. Tolksdorf, M. Mochol, R. Heese, R. Eckstein, R. Oldakowski, et C. Bizer, 2006. Semantic-Web-Technologien im Arbeitsvermittlungsprozess. *Wirtschaftsinformatik 48*(1), 17–26.

(Torres et al., 2007) J. Torres, M. El-Bèze, F. Béchet, et C. N., 2007. Comment faire pour que l'opinion forgée à la sortie des urnes soit la bonne ? Dans les actes de *DEFT 2007, Plate-forme AFIA*, Grenoble, France, 119–133.

(Trog et al., 2008) D. Trog, S. Christiaens, Z. Gang, et de Laaf J., 2008. Toward a community vision driven topical ontology in Human resource Management. Dans les actes de *OTM Workshops*, Volume 5333, Monterrey, Mexique, 615–624.

(Vapnik, 1982) V. Vapnik, 1982. *Estimation of dependencies based on empirical data*. Springer series in statistics. Springer.

(Vapnik, 1995) V. Vapnik, 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag.

(Vinot et al., 2003) R. Vinot, N. Grabar, et M. Valette, 2003. Application d'algorithmes de classification automatique pour la détection des contenus racistes sur l'internet. Dans les actes de *TALN*, France, 275–284.

(Viterbi, 1967) A. J. Viterbi, 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. Dans les actes de *IEEE Transactions on Information Processing*, Volume 13, 260–269.

(Watanabe et al., 2004) Y. Watanabe, K. Sono, K. Yokomizo, et Y. Okada, 2004. A question answer system using mails posted to a mailing list. Dans les actes de *2004 ACM Symposium on Document Engineering*, Milwaukee, Wisconsin, USA, 67–73. ACM, NY.

(Widrow et Hoff, 1960) G. Widrow et M. Hoff, 1960. Adaptive switching circuits. *In Institute of Radio Engineers, Western Electronic Show and Convention 4*, 96–104.

(Witten et Frank, 1999) I. H. Witten et E. Frank, 1999. *Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

(Zaragoza, 1999) H. Zaragoza, 1999. *Modèles Dynamiques d'Apprentissage Numérique pour l'Accès à l'Information Textuelle*. Thèse de doctorat, Université de Paris 6.