E-Gen: Traitement automatique d'informations de ressources humaines

Rémy Kessler* — **Juan-Manuel Torres-Moreno*,**** — **Marc El-Bèze***

* *LIA / Université d'Avignon, 339 chemin des Meinajariès, 84911 Avignon {remy.kessler, juan-manuel.torres, marc.elbeze}@univ-avignon.fr*
** *Ecole Polytechnique de Montréal, CP 6079 H3C3A7, Montréal (Québec) Canada*

RÉSUMÉ. *L'Internet est au cœur du marché du travail et son utilisation s'étend à mesure qu'augmente le nombre d'internautes. La recherche d'emploi au travers des « bourses à l'emploi électroniques » et l'e-recrutement se sont banalisés. Cette explosion d'informations pose divers problèmes pour leur traitement rapide et efficace. Nous présentons le projet E-Gen, qui permet d'analyser les offres d'emploi de manière automatique ou assistée. Basé sur des classifieurs pilotés par un automate de Markov, le système obtient de très bons résultats. Nous proposons également une stratégie afin d'assister les recruteurs dans la tâche –difficile et d'une grande subjectivité– de classement de candidatures. Nous évaluons différentes 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.*

ABSTRACT. *Internet is at the heart of the labor market and its use spreads as increase the number of Internet users in the population. Seeking employment through "electronic employment bursary" and e-recruitment has become something current. This information explosion poses various problems in their processing with the large amount of information difficult to manage quickly and effectively. We present in this work the E-Gen project, that analyses and integrates the ads. Based on 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 ads to assist recruiters. Relevance feedback approach allows to surpass our previous results on this task.*

MOTS-CLÉS : *Traitement Automatique du Langage Naturel Ecrit, Apprentissage Automatique, Recherche d'Information, Ressources humaines, modèles probabilistes, mesures de similarité.*

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

2e soumission à *Document Numérique numéro spécial*, le 18/05/10

1. Introduction

Que ce soit dans les réseaux d'entreprises ou directement sur le Web, la rapide augmentation de la quantité de données accessibles au 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. 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 est 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 centrée 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[1]) et HandiQuesta (`http://www.handiquesta.com`), un nouveau site d'emploi 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[2] 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 (CV-thèques). C'est notamment en ce sens qu'une étude de l'Agence pour l'emploi des cadres (APEC)[3] en 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 serait passée de plus de la moitié en 1996 à seulement un tiers, dix ans 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.

Cette explosion d'informations (août 2003 : 177 000 offres, mai 2008 : 500 000 offres comme le montre la figure 1)[4] pose divers problèmes dans leur traitement.

1. Devenu depuis Pôle Emploi (`http://www.pôle emploi.fr`).
2. 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/`).
3. Site web `http://www.apec.fr` ou `http://www.cadres.apec.fr`.
4. `http://www.Keljob.com`.

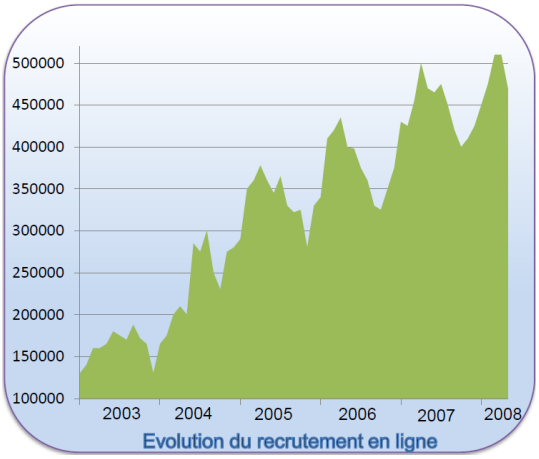

Figure 1 – Evolution du recrutement en ligne entre août 2003 et mai 2008

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) avance 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 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.*, 2006) pointent 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. État de l'art

Nous présentons dans cette partie les différentes approches qui ont été proposées dans la littérature afin d'aborder ces problématiques ainsi que différentes solutions innovantes sur le marché. La spécificité des informations contenues dans les documents

d'une candidature à une offre d'emploi conduit au 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 al.*, 2001). La méthode consiste à exploiter les caractéristiques dispositionnelles du document afin d'identifier chacune des parties et l'indexer en conséquence. (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. La seconde approche que l'on rencontre dans la littérature est celle d'ontologies spécifiques au domaine. Dans le domaine des ressources humaines, (Mocho *et al.*, 2006) décrivent l'importance 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 al.*, 2001; Dorn *et al.*, 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[5]. 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 interactions et du contexte. Plus proche des méthodes numériques, L'étude du document principal d'une candidature, le CV (curriculum vitae), a fait l'objet de différents travaux pour qu'il puisse être analysé automatiquement. (Clech *et al.*, 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 al.*, 2006; Roche *et al.*, 2008) décrivent une étude d'extraction de terminologie spécifique sur un corpus de CV[6]. 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.

Le second document d'une candidature est la lettre de motivation (abrégée en LM par la suite). 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.). Il y a peu de travaux consacrés

5. Par exemple `http://www.jobs.net`, `http://www.aftercollege.com`, `http://www..directjobs.com`, etc.
6. Corpus fourni par la société Vedior Bis (`http://www.vediorbis.com`).

au traitement des LM. On notera les travaux de (Audras *et al.*, 2006) sur les erreurs usuelles dans le passage à l'écrit d'une population apprenant le français. 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 porte en partie, sur la rédaction de LM. On notera aussi l'étude de (Amadieu, 2007) sur un petit échantillon de CV/LM. Il conclut qu'il n'existe pas de critère discriminant lors d'un recrutement dû à la faible différence de traitement entre les candidats dans les entreprises testées.

Parmi les solutions innovantes sur le marché, on notera les travaux de Lingway avec LINGWAY e-RH Applications[7] sur la base de thésaurus spécifiques au domaine, Twitter[8] qui lance le site de recherche d'emploi `www.twitterjobsearch.com` basé le concept de message court (moins de 140 caractères) et ZaPoint[9] qui propose une solution originale et intéressante dans l'intégration de CV, appelé Lifechart, qui permet de mettre le CV sous forme de graphique au travers de diverses courbes.

3. Architecture du système E-gen

Le LIA, en partenariat avec le LIRMM[10] et Aktor Interactive, une agence de communication française spécialisée dans l'*e-recruiting*, ont développé le système E-Gen pour résoudre ce problème. Celui-ci se compose de trois modules principaux :

– Extraction d'information à partir de corpus des courriels (d'offres d'emploi).

– Analyse de réponses des candidats (séparation automatique de LM et CV).

– Analyse et calcul d'un classement de pertinence du profil des candidats.

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. Après l'identification de la langue, E-Gen analyse le message et extrait le texte de l'offre d'emploi du message ou du fichier attaché à l'aide de modules externes en fonction du format (*wvWare*[11] et *pdftotext* afin de traiter les documents de source Ms Word et pdf). 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 contenant l'annonce est généré. Lors de la publication d'une offre d'emploi, une segmentation particulière des annonces est requise par les *job board*.

7. `http://www.lingway.com`.
8. `http://twitter.com`
9. `http://www.zapoint.com`
10. `http://www.lirmm.fr`
11. `http://wvware.sourceforge.net`.

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 dans la candidature est alors produite. Différents processus de filtrage et lemmatisation permettent au système d'identifier à l'aide de *Support Vector Machines* (SVM), le type du document (CV et/ou LM présents dans le corps du courriel ou dans les pièces jointes (module 2, section 5)). Une fois la LM et le CV 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, section 6). La figure 2 présente une vue d'ensemble du système.

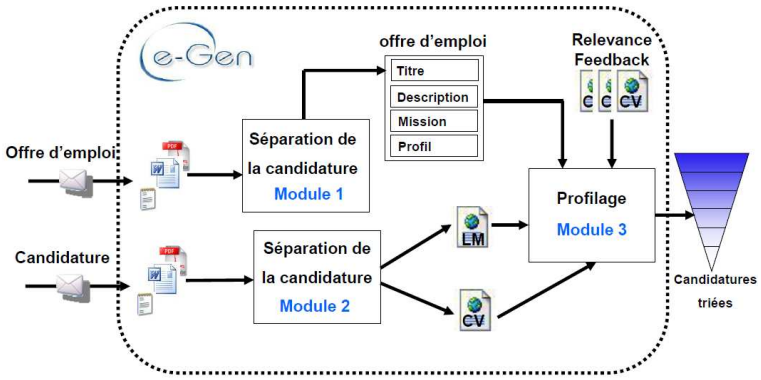

Figure 2 – Vue d'ensemble du système E-gen.

4. Prétraitement linguistiques

Dans un premier temps, nous effectuons un prétraitement des données textuelles (Offre d'emploi, CV et LM) permettant d'anonymiser les candidatures et de supprimer des informations non pertinentes telles que les noms des candidats, les adresses, les courriers électroniques, les noms de villes. La suppression des accents et des majuscules est également effectuée. 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 [12] :

12. *The curse of dimensionality.*

– **Uniformiser la casse** : transformation des majuscules en minuscules ;

– **Filtrage** : suppression des mots fonctionnels et des verbes (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.).

– **Lemmatisation** : Ce traitement peut entraîner une réduction importante du lexique. La lemmatisation simple consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et/ou féminins au masculin singulier[13] avant de leur associer un nombre d'occurrences. La lemmatisation permet donc de diminuer le nombre de termes qui définiront les dimensions de l'espace de représentation ou espace vectoriel ;

Même si une perte d'information difficilement quantifiable existe lors de ces opérations (perte de structures comme les énumérations, le rapprochement délicat de certains termes (« 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.

5. Traitement d'une offre d'emploi

5.1. *Analyse d'une offre*

L'extraction des 1 000 offres d'emploi les plus récentes à partir de la base d'informations d'Aktor Interative a permis d'avoir un corpus de taille relativement importante nommé par la suite *Corpus d'Offres d'Emploi*. Chacune de ces offres ayant été segmentée manuellement en fonction des besoins des sites d'emplois. Quelques statistiques du corpus sont rapportées dans le tableau 1. Une analyse rapide a montré que les offres d'emploi se composent souvent de blocs d'information thématiquement proches qui gardent une structure logique mais restent 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.

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

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

Nombre d'offres d'emploi	D=1 000	
Nombre total de segments	P=15 621	
Nombre de segments étiquetés **DESCRIPTION**	3 966	25,38 %
Nombre de segments étiquetés **TITRE**	1 000	6,34 %
Nombre de segments étiquetés **MISSION**	4 401	28,17 %
Nombre de segments étiquetés **PROFIL**	6 263	40,09 %

Tableau 1 – Statistiques du Corpus d'Offres d'Emploi.

5.2. *Classification par SVM, champs conditionnels aléatoires et méthode probabiliste*

Nous avons choisi les SVM pour cette tâche suite aux résultats lors de travaux précédents sur la classification de courriels (Kessler *et al.*, 2006). Nous nous sommes servi de l'implémentation LibSVM (Fan *et al.*, 2005), plus appropriée à cette tâche. Après une première étape de filtrage et de racinisation (voir section 4), nous utilisons une représentation vectorielle pour chaque segment de texte afin de lui attribuer une étiquette à l'aide des SVM. Nous avons effectué une série de tests afin de déterminer le noyau SVM le plus intéressant pour notre tâche. On observe que les noyaux polynomial, sigmoïde et radial obtiennent toujours des résultats inférieurs (linéaire 85 %, polynomial 75 %, sigmoide 55 %, radial 49 %). Au vu de ces résultats préliminaires, nous avons décidé d'utiliser un modèle SVM avec un noyau linéaire. Nous avons choisi par la suite les CRF (*Conditional Random Fields*) (Lafferty *et al.*, 2001) qui ont été utilisés avec succès dans de nombreuses tâches d'étiquetage telles que l'étiquetage morphosyntaxique ou la détection d'entités nommées. L'avantage principal des CRF par rapport aux modèles génératifs tels que les *Hidden Markov Model* (HMM) est la possibilité d'utiliser l'ensemble des observations d'une séquence pour prédire une étiquette. Dans notre cas, le corpus d'apprentissage est formaté de manière à associer à chaque mot une étiquette pour chaque partie de l'annonce. Nous nous sommes servi de l'implémentation CRF++[14]. Pour la méthode probabiliste, nous avons construit les uni-grammes et les bi-grammes de mots pour chaque partie de l'annonce avec leur probabilité P puis nous calculons pour obtenir le score $\tilde{t}$ des n-grammes pour un document D :

$$\tilde{t} = \underset{t}{\text{ArgMax}}\, P(t|W) = \underset{t}{\text{ArgMax}}\, \frac{P(W|t)P(t)}{P(W)} = \underset{t}{\text{ArgMax}}\, P(W|t)P(t) \quad [1]$$

Les deux dernières égalités proviennent de l'application du théorème de Bayes. En prenant comme hypothèse afin de donner tout son poids au contenu :

$$P(t) = \frac{1}{|T|} \text{ avec } t \in T, \text{ si } T \text{ dénote l'ensemble des classes} \quad [2]$$

14. Toolkit CRF++ : `http://www/chasen.org/taku/software/CRF++/`

on obtient :

$$\tilde{t} \approx \underset{t}{\mathrm{ArgMax}}\, P(W|t) = \underset{t}{\mathrm{ArgMax}} \prod_{i=1}^{|D|} P_t(W_i|W_1^{i-1}) \quad [3]$$

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

$$P_t(W_i|W_1^{i-1}) \approx \lambda P_t(W_i|W_{i-1}) + (1-\lambda)P_t(W_i) \quad [4]$$

Le tableau 2 présente des exemples de bi-grammes de mots discriminants pour chacune des classes dans le *Corpus d'Offres d'Emploi*. Les résultats obtenus montrent une classification performante des segments individuellement pour chacun des classifieurs (voir Figure 3), 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 3 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 % d'erreurs en moyenne), le classifieur à base de méthode probabiliste reste inférieur aux SVM ($\approx$ 13 % d'erreurs en moyenne) sur l'ensemble des tests, eux même inférieur au CRF ($\approx$ 5 % d'erreurs en moyenne).

DESCRIPTION	agence-français,acteur-majeur,nous-recruter,grand-distribuer,
TITRE	homme-femme, ingénieur-commerce, gestion-comptable
MISSION	mission-consister, faire-preuve, participer-réunion
PROFIL	adresser-candidature, adresser-dossier, vous-justifier, formation-supérieur

Tableau 2 – Bi-grammes de mots discriminants pour chaque partie de l'annonce

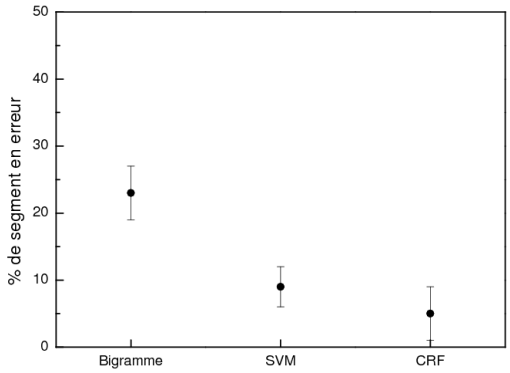

Figure 3 – Segments en erreurs pour les SVM, les CRF et la méthode probabiliste.

5.3. *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 SVM produisent globalement une bonne classification des segments individuels, mais les segments d'une 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 donné au tableau 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>
```

Tableau 3 – Exemple d'offre d'emploi.

Le dernier segment est catégorisé **Titre**, mais cela est impossible. Afin de corriger ce problème, nous avons opté 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 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. Celui-ci possède 4 états plus les états initial et final. Chaque état pouvant boucler sur

	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

Tableau 4 – Matrice de Markov.

lui-même ou passer à l'état suivant en fonction d'une certaine probabilité déterminée par la matrice 4. Le processus correctif décrit en 5.4 est piloté par cet automate.

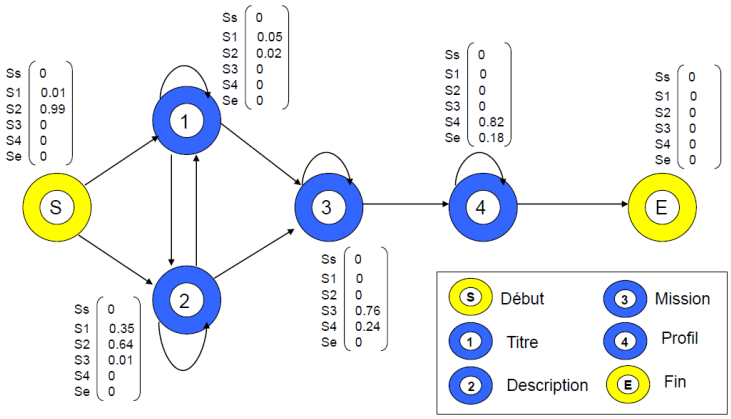

Figure 4 – Automate de Markov utilisé pour corriger les étiquettes des segments mal classés.

5.4. *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 al.*, 1999; Viterbi, 1967). La classification par SVM donne à chaque segment individuellement une classe. Une offre complète est une succession de segments 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. Notre algorithme inspiré 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, il renvoie 0, la séquence est rejetée et le processus correctif est interrogé afin de transmettre la séquence avec un nombre d'erreurs minimales (comparé à la séquence produite par les SVM) et une probabilité maximale. Ce processus parcourt l'arbre des solutions possibles en calculant la probabilité de la séquence ainsi que le nombre de différences par rapport à la séquence produite par les SVM. Le schéma 5 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, il est impossible de passer de l'état 4 (**Profil**) à l'état 3 (**Mission**)). Le post-traitement propose donc différentes solutions : les trois premières ayant deux différences par rapport à la séquence SVM, 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.

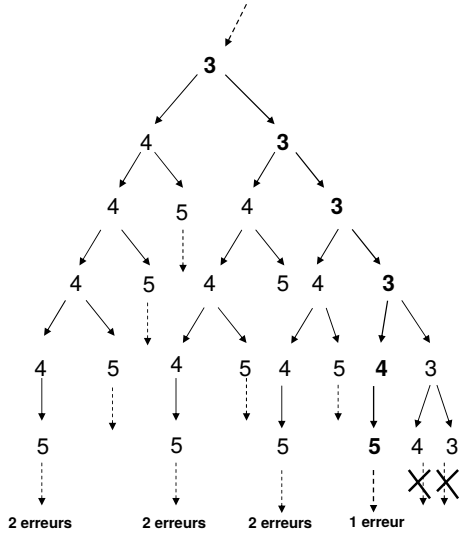

Figure 5 – Déroulement du post-processus correctif.

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 une annonce d'une cinquantaine de segments). Nous avons introduit une amélioration en utilisant un algorithme *Branch and Bound* (Land *et al.*, 1960) pour élaguer l'arbre : dès qu'une première solution est trouvée, le nombre de différences par rapport à la séquence du classifieur 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 inférieure), le reste de la séquence n'est pas calculé. L'utilisation de cet algorithme permet d'obtenir une solution avec des temps très ac-

ceptables (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, le processus correctif n'apporte aucune amélioration.

5.5. *Résultats*

Nous avons effectué une évaluation selon deux niveaux de finesse : par segment et par offre d'emploi entièrement reconnue. La figure 6 à gauche montre une comparaison entre les résultats obtenus par les SVM, les CRF avec et sans 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 SVM seules, des CRF seules 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), les deux classifieurs obtiennent un fort taux de patrons bien classés (taux d'erreurs < 10 %).

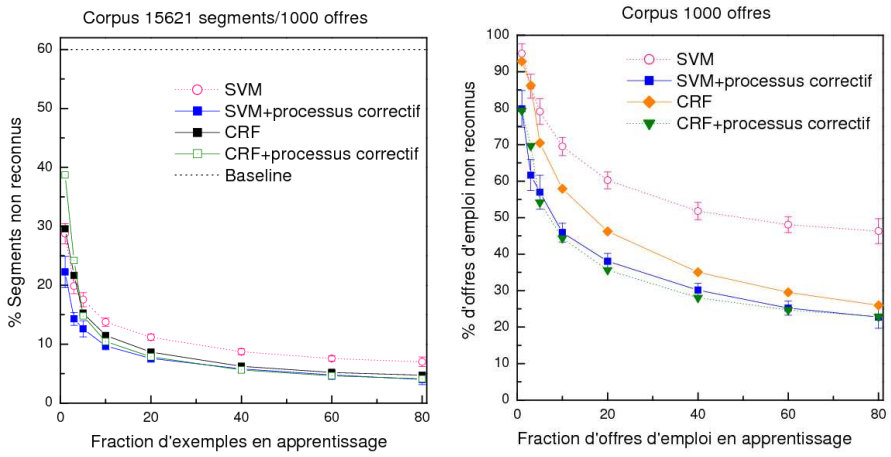

Figure 6 – À gauche, les taux d'erreurs en pourcentage des SVM et des CRF avec et sans l'algorithme correctif par rapport au nombre de segments mal étiquetés. À droite, les taux d'erreurs en pourcentage des SVM et des CRF avec et sans l'algorithme correctif par rapport aux offres d'emploi reconnues de façon erronée.

Le processus correctif (ligne continue) donne toujours de meilleurs résultats que les classifieurs 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 6 présente une comparaison entre les résultats obtenus par chaque méthode en fonction de chaque niveau de finesse (par segment et par offre d'emploi entièrement reconnue). On observe une considérable amélioration du nombre d'offres d'emploi identifiées avec le processus correctif. Avec un apprentissage de 80% SVM

obtient un minimum d'environ 50 % des offres d'emploi mal étiquetées, et le processus correctif en obtient 25 %, donc une amélioration de plus de 50 % du score des SVM. Les CRF obtiennent d'environ 30 % des offres d'emploi mal étiquetées, et le processus correctif en obtient 25 %, donc une amélioration de 5 %.

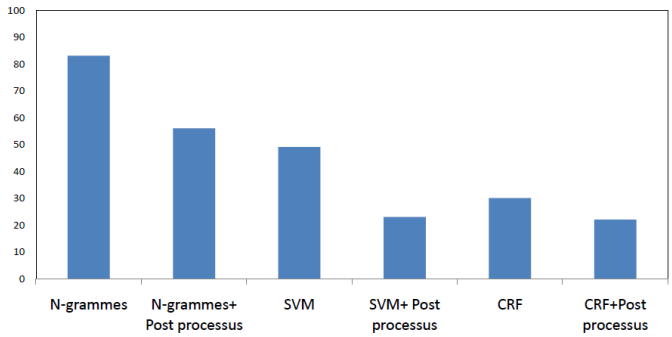

Figure 7 – Comparaison des résultats obtenus avec un apprentissage fixé à 80 % pour les différentes méthodes de classification avec ou sans processus correctif.

La figure 7 montre une comparaison entre les résultats obtenus pour chaque méthode, avec ou sans processus correctif, selon les offres d'emploi partiellement 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 la méthode probabiliste, environ 20 % pour les SVM et d'environ 5% pour les CRF). L'ensemble des tests montre également que la classification par CRF obtient des résultats de meilleure qualité que les deux autres classifieurs. Par ailleurs, les résultats des CRF et des SVM avec le post processus sont très proches.

6. Évaluation des réponses

Nous souhaitons mettre en place un système capable de fournir une première évaluation automatisée des candidatures selon divers critères. Nous présentons dans cette section les travaux concernant les problématiques de séparation de CV/LM ainsi que ceux correspondant au classement de candidatures.

6.1. *Corpus et analyse de candidatures*

Nous avons effectué une seconde extraction du système d'information en regroupant plusieurs offres d'emplois ainsi que les diverses réponses à ces offres d'emplois étiquetées par un recruteur du cabinet lors de la présélection. Il regroupe un ensemble d'offres d'emploi avec des thématiques différentes (emplois en comptabilité, entreprise, informatique, etc.) ainsi qu'une étiquette associée à chaque candidature par le recruteur. Afin de simplifier le problème, nous avons réduit l'ensemble des étiquettes

à deux 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 le corpus, celui-ci étant majoritairement composé de candidatures étiquetées **non retenues** comme le montre le tableau 5. Les offres d'emplois peuvent être rédigées en différentes langues, mais n'avons conservé pour cette étude que les offres et les réponses en français. Ce sous-ensemble, nommé *Corpus offres/réponses* a donc permis d'obtenir un corpus de réponses classées en fonction de l'offre d'emploi ainsi que du jugement d'un recruteur sur les candidatures.

Nombre total d'offres d'emplois	25
Nombre d'offres d'emplois avec moins de 10 réponses	2
Nombre d'offres d'emplois avec 11 à 50 réponses	8
Nombre d'offres d'emplois avec 51 à 100 réponses	6
Nombre d'offres d'emplois avec plus de 100 réponses	9
Nombre total de candidatures	1586
Nombre de candidatures **retenues**	160
Nombre de candidatures **non retenues**	1426

Tableau 5 – Statistiques du Corpus *offres d'emplois*.

Nous avons effectué un prétraitement 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 a également été effectuée. Nous avons, par la suite, utilisé les processus de filtrage et racinisation présentés en section 4.

6.2. *Séparation CV/LM*

Pour arriver à identifier automatiquement le type de document traité, nous avons tenté une classification simpliste en nous basant uniquement sur les noms des fichiers. Ceci s'est avéré insuffisant[15] en raison de la diversité des noms de fichiers[16]. L'observation des statistiques du tableau 6 nous a incité à créer deux classifieurs basiques 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 documents courts et les CV des documents de synthèse présentant peu de caractères délimitant les phrases (« . », « . : », etc.). Nous avons ainsi construit un premier classifieur basique *N1* qui décide en fonction de la longueur des phrases p ($p < 18 \Rightarrow$ CV, $p > 18 \Rightarrow$ LM) et un second *N2* en fonction du nombre de mots m ($m < 200 \Rightarrow$ LM, $m > 200 \Rightarrow$ CV). Cependant les résultats (tableau 7) montrent les limites de ce genre de méthode.

15. Le système constituait un corpus tronqué à 1725 CV et 910 LM.
16. Par exemple : PierreDurand.doc, Durand.pdf, sociétéX.doc, 13042007.doc, V3.doc, etc.

	CV	LM
Nombre de documents	2 165	2 165
Nombre total de phrases	45 655	20 658
Longueur moyenne des phrases	17,07	18,97
Nombre total de mots	922 103	412 008
Moyenne de mots par document	425,91	190,30

Tableau 6 – Statitistiques des classifieurs basiques.

Classifieur	Précision	Rappel	F-score
N1	0,66	1	0,75
N2	0,35	0,26	0,30

Tableau 7 – Précision, Rappel, F-score obtenus par les deux classifieurs basiques.

Le tableau 7 présente les résultats obtenus avec chacun des classifieurs basiques, *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 restent décevants. Nous expliquons cela par l'hétérogénéité des données (des CV parfois très courts et des LM parfois extrêmement longues) qui entraîne une variance importante perturbant la prise de décision.

Nous avons donc choisi les SVM, pour cette tâche compte-tenu des bons résultats obtenus dans les travaux précédents en catégorisation de texte. Après une étape de filtrage et de racinisation (voir section 4), nous utilisons une représentation vectorielle de chaque document et nous lui attribuons une étiquette (CV ou LM). Afin de régler les paramètres et tester nos méthodes, nous avons effectué une validation croisée en divisant le *Corpus offres/réponses* 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-Moreno *et al.*, 2007). Les algorithmes ont été évalués sur les corpus de test en utilisant la mesure F-score sur l'ensemble de documents bien classés, avec une macro-moyenne sur toutes les classes (avec $\beta = 1$ afin de ne privilégier ni la précision ni le rappel) (Goutte *et al.*, 2005).

Les résultats obtenus par les SVM sur la tâche de classification de CV/Lettre de motivation sont d'excellente qualité en Précision, Rappel et F-score sur chaque sous-ensemble (entre 0,95 et 0,98). Ceux-ci ont été publiés dans (Kessler *et al.*, 2007).

Une analyse des CV/Lettre de motivation mal étiquetés fait apparaître qu'il y a deux types de CV mal classés : certains sont de mauvais étiquetage dans le *Corpus de référence* et d'autres des CV, étiquetés LM, qui sont 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 comme le montre les exemples dans (Kessler *et al.*, 2008). Une fois la problématique de tri de CV/LM résolue, nous avons pu générer le *Corpus offres/réponses* présenté par la suite et qui a servi aux expériences pour l'ensemble du troisième module et de la tâche de classement de candidatures.

6.3. *Comparaison Candidature/offre d'emploi par mesure de similarité*

Nous avons opté pour une approche par mesure de similarité afin d'obtenir un classement des candidatures par rapport aux offres d'emploi proposées.

Chaque document a été transformé en un vecteur avec des poids représentant la fréquence des termes (*Tf*). Une série de tests a été effectuée dans (Kessler *et al.*, 2009) avec le *Tf-idf* sans amélioration visible. Les mesures de similarité que nous avons utilisées sont décrites dans (Bernstein *et al.*, 2005) : le cosinus (5), qui permet de calculer l'angle entre l'offre d'emploi et la réponse de chaque candidat, les distances de Minkowski (6) ($p = 1$ pour Manhattan et $p = 2$ pour la distance euclidienne), et la distance de Recouvrement (7) :

$$sim_{\text{cosine}}(j, d) = \frac{\sum_{i=1}^{n} j_i \cdot d_i}{\sqrt{\sum_{i=1}^{n} j_i^2 \cdot \sum_{i=1}^{n} d_i^2}} \quad [5]$$

$$sim_{\text{Minkowski}}(j, d) = \frac{1}{1 + (\sum_{i=1}^{n} |j_i - d_i|^p)^{\frac{1}{p}}} \quad [6]$$

$$sim_{\text{Recouvrement}}(j, d) = \frac{\sum_{i=1}^{n} j_i \cdot d_i}{Min\left(\sum_{i=1}^{n} |j_i|^2, \sum_{i=1}^{n} |d_i|^2\right)} \quad [7]$$

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.
Afin de combiner ces mesures, nous avons effectué une normalisation de chaque mesure selon les minima/maxima puis nous avons utilisé un algorithme de décision (AD) (Boudin *et al.*, 2007; Torres Moreno *et al.*, 2009) qui fusionne les valeurs obtenues par chaque mesure de similarité.

6.4. *Relevance Feedback*

Afin d'améliorer les résultats obtenus, 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 en l'enrichissant de façon plus ou moins automatique au moyen de documents trouvés lors d'une première passe. 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 jugés pertinents pour améliorer le résultat de cette nouvelle requête. L'algorithme de (Rocchio, 1971) et ses différentes variations ont été largement utilisés dans le domaine de la recherche d'information (Frakes *et al.*, 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 [17]. 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 de pouvoir signaler à un recruteur 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 en repêchant ce qui a échappé à sa vigilance. 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 al.*, 1990). Dans notre contexte, nous effectuons un tirage aléatoire de quelques candidatures (de une à sept dans nos expérimentations) parmi l'ensemble des candidatures étiquetées par le recruteur comme **pertinentes**. Celles-ci sont ajoutées à l'offre d'emploi. Nous 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 Sim', pour chaque mesure de similarité entre la candidature que nous évaluons et l'offre d'emploi enrichie du nombre de candidatures **pertinentes** du processus de *Relevance Feedback* :

$$Sim'_{\text{mesure}}(j, d) = Sim_{\text{mesure}}(j, d \| p_1 \| \cdots \| p_n) \quad [8]$$

avec j une offre d'emploi, d la candidature, p_i une candidature **pertinente**, n le nombre de candidatures retenues pour le *Relevance Feedback* et $\|$ l'opérateur de concaténation.

6.5. *Protocole expérimental*

Nous souhaitons mesurer la similarité entre une offre d'emploi et ses candidatures. Le *Corpus offres/réponses* contient 25 offres d'emplois associées à au moins quatre candidatures. Nous transformons chaque document en une représentation vectorielle

17. JobFinder (`http://jobfinder.com`).

(Salton, 1991). Puis, nous mesurons leur similarité à l'aide des mesures présentées en section 6.3. Chacune de ces 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 d'incorrect (c'est à dire les candidatures **non pertinentes** mieux classées que les candidatures **pertinentes**) en abscisse et le taux de correct (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*). Le principal avantage des courbes ROC est leur résistance au déséquilibre dans le corpus (par exemple un déséquilibre entre les exemples **retenu** et **non retenu**). Le détail et l'intérêt de cette mesure sont développés dans (Roche *et al.*, 2006). Pour chaque offre d'emploi du corpus, nous évaluons la qualité du classement obtenu avec les courbes ROC. Nous avons écarté lors de l'évaluation les candidatures où une pièce était manquante (CV ou LM).

6.6. *Résultats*

Nous nous sommes intéressés à la structure de nos données. Comme déjà mentionné en 5.1, 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**) sans tenir compte du découpage ;
– L'offre d'emploi composée de son titre, sa mission et son profil (**TMP**).

	AUC	Cosine	Minkowski	Manhattan	Overlap	Décision
DTMP	LM	0,567	0,561	**0,591**	0,573	0,596
	CV	**0,604**	0,510	0,503	0,543	0,562
	LM+CV	**0,621**	0,539	0,532	0,522	0,571
TMP	LM	0,560	0,559	0,580	0,562	**0,591**
	CV	**0,622**	0,508	0,501	0,538	0,561
	LM+CV	**0,622**	0,538	0,528	0,531	0,592

Tableau 8 – AUC obtenu en fonction du découpage de l'offre d'emploi.

Le tableau 8 présente les résultats obtenus en fonction de la candidature globale : le CV, la LM avec une offre d'emploi DTMP ou TMP. Ceux-ci ont été publiés dans (Kessler *et al.*, 2009). 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 meilleurs 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. Nous observons que le CV contient plus d'informations pertinentes que la LM. Ceci vient confirmer notre intuition que le CV est le document principal de la candidature.

AUC	Cosine	Minkowski	Manhattan	Overlap	Décision
CV_1/3	**0,589**	0,497	0,505	0,539	0,579
CV_2/3	**0,600**	0,524	0,520	0,577	0,580
CV_3/3	**0,526**	0,497	0,503	0,479	0,501
LM_1/3	0,573	0,561	**0,588**	0,571	0,580
LM_2/3	0,565	0,570	**0,578**	0,578	0,570
LM_3/3	0,447	0,528	**0,538**	0,446	0,470

Tableau 9 – AUC obtenu en fonction du découpage des réponses.

Dans le tableau 9, nous présentons 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, cela permet de conclure que les informations les plus importantes afin de déterminer si une candidature est pertinente ou pas se situent dans les deux premiers tiers de chaque document. Nul besoin d'effectuer une analyse très poussée pour découvrir ce à quoi on pouvait s'attendre : le dernier tiers 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.). La figure 8 présente, sous forme graphique, l'ensemble des résultats obtenus en fonction des découpages.

6.7. *Résultats Relevance Feedback*

Afin de pouvoir tester le principe de *Relevance Feedback*, il a été nécessaire de retirer certaines offres d'emploi du *Corpus offres/réponses*, celles-ci ne possédant pas un nombre de candidatures **retenues** suffisantes. Les offres avec moins de 6 candidatures **retenues** ont été retirées. Nous désignerons ce nouveau corpus comme *Corpus RF*. Nous utilisons un *residual ranking* (Billerbeck *et al.*, 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 un tirage aléatoire des candidatures **pertinentes** ajoutées au *Relevance Feedback*. Nous désignons par RF1 à RF7 le nombre de candidatures utilisées lors du processus de *Relevance Feedback* (RF1 pour 1 candidature, RF2 pour 2 etc.). On observe une progression positive du score AUC entre la référence et le RF7 pour 10 offres sur 12. L'étude détaillée des résultats montre que l'offre 33746 comporte quelques candidatures vides étiquetées **pertinentes**. Ce qui conduit le système à dégrader les résultats obtenus lorsque celles-ci sont les seules à être évaluées. L'offre 34783 obtient un bon score dès le départ mais

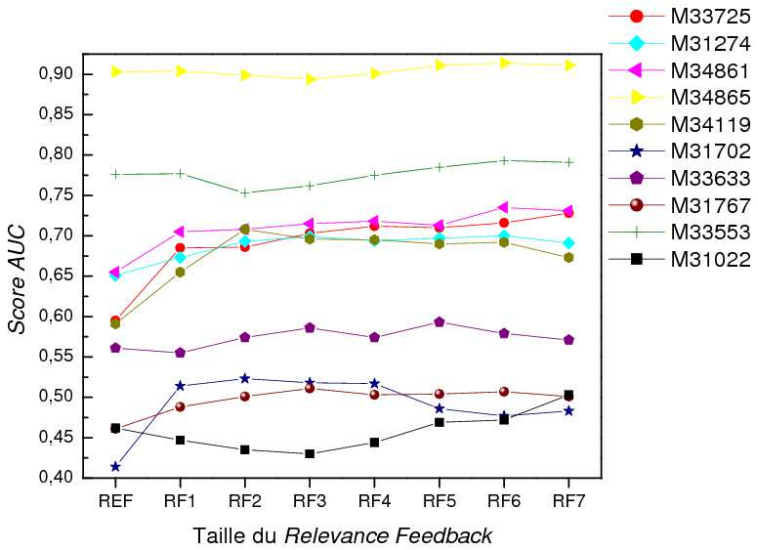

Figure 8 – Comparaison des résultats obtenus avec chaque mesure de similarité pour chaque tiers de document (CV et LM).

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 prises en compte dans le résultat. La figure 9 illustre graphiquement les résultats des offres d'emploi ayant eu une progression avec le retour de pertinence.

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

Tableau 10 – Comparaison entre les scores AUC obtenus en fonction de chaque taille de Relevance Feedback et la référence pour chaque offre d'emploi, triée en fonction du numéro de l'offre.

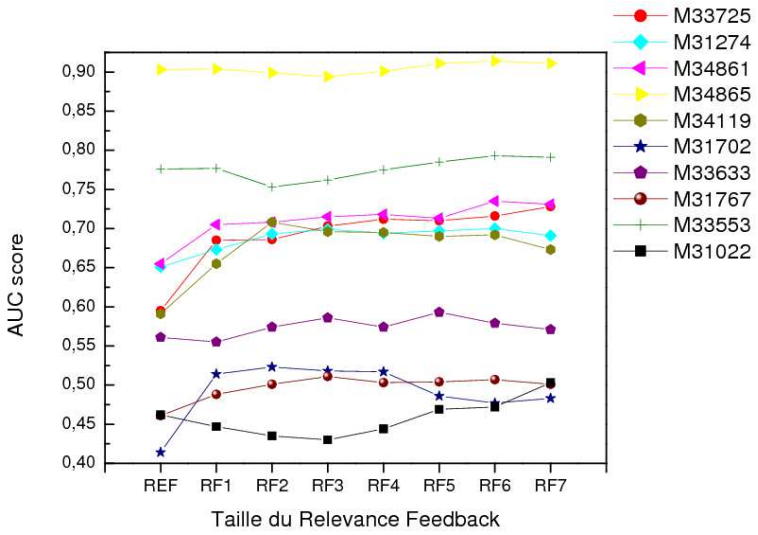

Figure 9 – Comparatif entre les tailles de RF et les résultats de référence (REF) pour chaque offre d'emploi

La progression n'est pas régulière mais le processus de Relevance Feedback améliore les résultats obtenus. 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 candidatures **pertinentes** est trop petit dans certaines offres d'emplois. Il serait intéressant de pouvoir tester le Relevance Feedback avec un nombre plus important de candidatures ajoutées afin de connaître ses limites, cependant cela nous obligerait à retirer à nouveau des offres d'emploi du *Corpus RF*.

7. Conclusion

Dans cet article, nous avons présenté E-gen, système pour le traitement automatiquement des offres d'emploi, découpé en 3 modules. Le premier module est dédié à la catégorisation des différentes sections composant une offre d'emploi. Le traitement de ce type d'information est difficile car il s'agit de textes libres (non structurés) provenant 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. Les premiers résultats obtenus par les SVM et les CRF étaient très intéressants avec environ 90 % et 95 % de segments bien étiquetés pour un corpus d'apprentissage de 80 %. Le processus correctif améliore ces résultats pour chaque

méthode de classification (SVM, CRF et méthode probabiliste) et diminue considérablement les erreurs de segments isolés incorrectement classés, tout en restant dans des temps de calcul très raisonnables.

La seconde partie met en avant les deux modules de traitement des réponses à des offres d'emploi. La tâche de séparation de chaque document de la candidature est effectuée par un SVM et obtient de très bons résultats. Le traitement des candidatures de façon automatisée est une tâche extrêmement difficile car l'information est en format libre malgré une structure conventionnelle. Le classement des candidatures restant fortement subjectif, nous avions 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. Nous avons observé une amélioration des AUC obtenue à l'aide du retour de pertinence. 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 l'offre, découpage de la candidature en fonction de termes particuliers 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 offres d'emploi les plus adaptées à son profil) ainsi qu'une simulation de l'évaluation sur une offre donnée pour permettre au candidat d'améliorer son dossier.

8. Bibliographie

Allen C., Pilot L., « HR-XML : Enabling Pervasive HR- e-Business », *XML Europe 2001, Int. Congress Centrum (ICC)*, 2001.

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

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

Autor D. H., « Wiring the Labor Market », *Journal of Economic Perspectives*, vol. 15, n° 1, p. 25-40, 2001.

Beauvallet G., Le Garff M.-C., Negri A.-L., Cara F., « L'usage d'Internet par les demandeurs d'emploi », *Revue de l'IRES - numéro spécial : Internet, recrutement et recherche d'emploi*, vol. 3, n° 52, p. 41-69, 2006.

Bernstein A., Kaufmann E., Kiefer C., Bürki C., SimPack : A Generic Java Library for Similarity Measures in Ontologies, Technical report, University of Zurich, August, 2005.

Billerbeck B., Zobel J., « Efficient query expansion with auxiliary data structures », *Inf. Syst.*, n° 7, p. 573-584, 2006.

Boudin F., Torres Moreno J. M., « NEO-CORTEX : A Performant User-Oriented Multi-Document Summarization System », *CICLing*, p. 551-562, 2007.

Bourse M., Leclère M., Morin E., Trichet F., « Human resource management and semantic Web technologies », *ICTTA*, p. 641-642, 2004.

Cazalens S., Lamarre P., « An organization of Internet agents based on a hierarchy of information domains », *in* Y. D. . F. J. Garijo (ed.), *10th European Workshop on Multi-Agent Systems, MAAMAW 2001*, Annency, France, p. 12-27, mai, 2001.

Clech J., Zighed D. A., « Data Mining et analyse des CV : une expérience et des perspectives », *Extraction et la Gestion des Connaissances, EGC'03*, Lyon, France, p. 189-200, 2003.

Desmontils E., Jacquin C., Morin E., « Indexation sémantique de documents sur le Web : application aux ressources humaines », *Journées de l'AS-CNRS Web Sémantique*, 2002.

Dorn J., Naz T., Pichlmair M., « Ontology Development for Human Resource Management », *4th International Conference on Knowledge Management*, Vienne, p. 109-120, 2007.

Fan R.-E., Chen P.-H., Lin C.-J., « Working set selection using the second order information for training SVM », , vol. 6, p. 1889-1918, 2005.

Ferri C., Flach P., Hernandez-Orallo J., « Learning decision trees using the area under the ROC curve », *19th International Conference on Machine Learning, ICML'02*, p. 139-146, 2002.

Fondeur Y., « Internet, recrutement et recherche d'emploi : une introduction », *Revue de l'IRES - numéro spécial : Internet, recrutement et recherche d'emploi*, vol. 3, n° 52, p. 3-10, 2006.

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

Goutte C., Gaussier E., « A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation », *ECIR*, p. 345-359, 2005.

Joachims T., « A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization », *ICML '97 : Proceedings of the Fourteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 143-151, 1997.

Kessler R., Béchet N., Roche M., El-Bèze M., Torres-Moreno J. M., « Job Offer Management : How Improve the Ranking of Candidates », *ISMIS 2009, Prague*p. 431-441, 2009.

Kessler R., Torres-Moreno J. M., El-Bèze M., « Classification automatique de courriers électroniques par des méthodes mixtes d'apprentissage. », p. 93-112, 2006.

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

Kessler R., Torres-Moreno J. M., El-Bèze M., « E-Gen : Profilage automatique de candidatures », *TALN 2008, Avignon, France*p. 370-379, 2008.

Knouse S. B., « Impression management in the resume and its cover letter », *Journal of Business and Psychology*p. 242-249, 1988.

Lafferty J. D., McCallum A., Pereira F., « Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data », *ICML '01 : Proceedings of the Eighteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 282-289, 2001.

Land A. H., Doig A. G., « An Automatic Method of Solving Discrete Programming Problems », *Econometrica*, vol. 28, p. 497-520, 1960.

Leuski A., « Relevance and reinforcement in interactive browsing », *In Proceedings of Ninth International Conference on Information and Knowledge Management*, p. 119-126, 2000.

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

Mellet K., « 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*, vol. 3, n° 52, p. 71-100, 2006.

Mocho M., Paslaru E., Simperl B., « Practical Guidelines for Building Semantic eRecruitment Applications », 2006.

Morin E., Leclère M., Trichet F., « The Semantic Web in e-recruitment », *First European Symposium of Semantic Web (ESWS'2004)*, 2004.

Quilan J., « C4.5 : Programs for Machine Learning. », *Kaufmann, San Mateo, CA*, 1993.

Rafter R., Bradley K., Smyth B., « Automated Collaborative Filtering Applications for Online Recruitment Services », , vol. 1892, p. 363-368, 2000a.

Rafter R., Smyth B., Bradley K., « Inferring Relevance Feedback from Server Logs : A Case Study in Online Recruitment », 2000b.

Rocchio J., *Relevance Feedback in Information Retrieval*, p. 313-323, 1971.

Roche M., Kodratoff Y., « Pruning Terminology Extracted from a Specialized Corpus for CV Ontology Acquisition », *OTM'06, Montpellier, France*, p. 1107-1116, 2006.

Roche M., Prince V., « Evaluation et détermination de la pertinence pour des syntagmes candidats à la collocation », *JADT'08 (Journées internationales d'Analyse statistique des Données Textuelles)*, vol. 2, Lyon, France, p. 1009-1020, 2008.

Salton G., « Developments in Automatic Text Retrieval », *Science*, vol. 253, n° 5023, p. 974-980, 1991.

Salton G., Buckley C., « Improving Retrieval Performance by Relevance Feedback », *Journal of the American Society for Information Science*, vol. 41, n° 4, p. 288-297, 1990.

Spärck J. K., « Some thoughts on classification for retrieval », *Journal of Documentation*, vol. 26, p. 89-101, 1970.

Torres-Moreno J.-M., El-Bèze M., Béchet F., Camelin N., « Comment faire pour que l'opinion forgée à la sortie des urnes soit la bonne ? », *DEFT 2007*, Grenoble, France, p. 119-133, 2007.

Torres Moreno J. M., St-Onge P.-L., Gagnon M., El-Bèze M., Bellot P., « Automatic Summarization System coupled with a Question-Answering System (QAAS) », *CoRR*, 2009.

Trog D., Christiaens S., Gang Z., de Laaf J., « Toward a community vision driven topical ontology in Human resource Management », *OTM Workshops*, vol. 5333, Monterrey, Mexique, p. 615-624, 2008.

Viterbi A. J., « Error bounds for convolutional codes and an asymptotically optimal decoding algorithm », *IEEE Transactions on Information Processing*, vol. 13, p. 260-269, 1967.