E-Gen : traitement automatique des offres d'emploi

Rémy Kessler[1, 2], Marc El-Bèze[1]

[1] Laboratoire Informatique d'Avignon, BP 1228 F-84911 Avignon Cedex 9 FRANCE

[2] AKTOR Interactive Parc Technologique – 12, allée Irène Joliot Curie Bâtiment B3
69 800 Saint Priest

Abstract

The exponential growth of the Internet has made the development of a market of on-line job search sites possible. This paper aims at presenting the E-Gen system (Automatic Job Offer Processing system for Human Resources). E-Gen 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's 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 then transmitted to a "corrective" post-process (with the Markov model and a Branch&Bound algorithm for pruning the tree) which improves the quality of the solution.

Résumé

La croissance exponentielle de l'Internet a permis le développement d'un grand nombre de sites d'offres d'emploi et d'un marché de recrutement en ligne. Nous proposons le système E-Gen (Traitement automatique d'offres d'emploi) qui a pour but l'analyse et la catégorisation d'offres d'emploi et des réponses des candidats (lettre de motivation et CV). Le traitement de ce type d'informations est difficile car il s'agit de texte libre (non structuré). Nous présentons dans ce papier la stratégie mise en place afin de résoudre la première tâche : après des processus de pré-traitement classique (filtrage et lemmatisation), nous utilisons la représentation vectorielle de textes avant d'effectuer une classification avec des machines à support vectoriel (SVM) 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&Bound qui améliorent sensiblement les résultats.

Mots-clés : classification de texte, machines à support vectoriel, ressources humaines, offres d'emploi.

1. Introduction

La croissance exponentielle de l'Internet a permis le développement de *jobboards*[1] (Bizer et al, 2005 ; Rafter et al, 2000). Cependant, les réponses des candidats représentent une grande quantité d'information difficile à gérer rapidement et efficacement pour les entreprises (Bourse et al, 2004 ; Morin et al, 2004 ; Rafter et al, 2001). En conséquence, il est nécessaire de traiter cette masse de documents d'une manière automatique ou assistée. Le LIA et Aktor Interactive[2], agence de communication française spécialisée dans l'*e-recruiting*, développent le système E-Gen pour résoudre ce problème.

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

[1] Sites d'offres d'emploi en ligne : Monster (www.monster.fr), Central Job (www.centraljob.fr) etc.

[2] http://www.aktor.fr

- Un module d'extraction de l'information à partir de corpus des courriels provenant d'offres d'emplois extraites de la base de données d'Aktor.
- Un module pour analyser et calculer un classement de pertinence du profil du candidat (lettre de motivation et curriculum vitae).

Afin d'extraire l'information utile, ce 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, 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'information disponible sur internet et aux nombres importants de jobboards (spécialisés[3], non spécialisés[4] ou locaux[5]), Aktor a besoin d'un système capable de traiter rapidement et efficacement ces offres d'emploi afin de pouvoir par la suite les diffuser. Pour cela, Aktor utilise un système automatique pour envoyer les offres au format XML (*Robopost Gateway*). Le processus complet est décrit dans (Kessler et al, 2007). 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 (contrat, salaire, localisation, etc.). Auparavant, cette première étape était une tâche manuelle : on demande aux utilisateurs de copier et coller les offres d'emploi dans le système d'information de l'entreprise. Cette communication présente seulement ce premier module du système E-Gen et sa performance sur la tâche d'extraction et de catégorisation. Nous aborderons en section 2 l'architecture globale du système E-Gen avant de détailler dans la section 3 la modélisation adoptée pour notre problème. Nous présentons en section 4 les différents algorithmes de classification mis en place, avant de détailler les différents résultats obtenus dans la section suivante.

2. Architecture du système

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. 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 les documents MS-Word et produit une version texte du document découpé en segments[6].

Après une étape de filtrage[7] et lemmatisation[8], nous utilisons la représentation vectorielle pour chaque segment afin de lui attribuer une étiquette correspondant à son rôle dans le texte (cf. section 3.1) à l'aide des machines à support vectoriel (cf. section 4).

---

[3] http://www.admincompta.fr (comptabilité), http://www.lesjeudis.com (informatique).

[4] http://www.monster.fr, http://www.cadremploi.fr , http://www.cadresonline.com.

[5] http://www.emploiregions.com http://www.regionsjob.com.

[6] http://wvware.sourceforge.net. La segmentation de textes MS-Word étant difficile, on a opté pour un outil existant. Dans la majorité des cas, il sectionne en paragraphes le document.

[7] Pour réduire la complexité du texte, différents filtrages du lexique sont effectués~: la suppression des verbes et 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 <$>, <#>,<*>, etc.

Par la suite, cette séquence d'étiquettes, représentant la séquence des différentes parties du texte d'annonce, est traitée par un processus correctif qui la valide ou qui propose une meilleure séquence (cf. section 4.3). À la fin du traitement, un fichier xml est généré et envoyé au système d'information d'Aktor. La chaîne de traitement complète est représentée dans la figure 1.

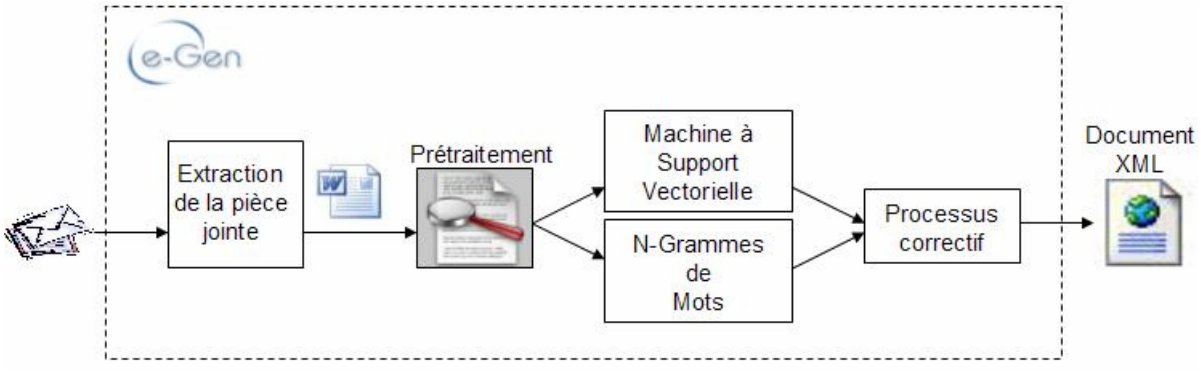

Figure 4 : Chaîne de traitement du système E-Gen

Lors de la publication d'une offre d'emploi, un certain nombre d'informations est requis par le jobboard. 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 écrite à la main afin de localiser des informations telles que salaires, lieu de travail, noms d'entreprises, contrat, référence, durée de la mission.

3. Modélisation du problème

3.1. Corpus

Un sous-ensemble de données[9] a été sélectionné à partir de la base de données d'Aktor. Ce corpus regroupe plusieurs types d'offres d'emploi en différentes langues, mais notre étude porte sur les offres en français (le marché français représente l'activité principale d'Aktor). Ce sous-ensemble a été nommé *Corpus de référence*. Un exemple d'offre d'emploi est présenté en figure 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).
```

Figure 5 : Exemple d'offre d'emploi

[8] La lemmatisation simple consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et/ou féminins au masculin singulier. Ainsi on pourra ramener à la même forme **chanter** les mots *chante, chantaient, chanté, chanteront* et éventuellement *chanteur*.

[9] Basé sur l'historique de la base de données de la société nettoyé des différents enregistrements erronées/vides ou de test, et filtré sur la langue française.

L'extraction à partir de la base de données d'Aktor a permis d'avoir un corpus de taille importante, sans catégorisation manuelle. Une première analyse a montré que les offres d'emploi se composent souvent de blocs d'information semblable qui demeurent, cependant, fortement non structurées. Une offre d'emploi est composée de quatre blocs :

- **Titre** : titre probable de l'emploi ***;***
- **Description** : bref résumé de l'entreprise qui recrute ;
- **Mission** : courte description de l'emploi ;
- **Profil et contacts** : qualifications et connaissances exigées pour le poste. Les contacts sont généralement inclus dans cette partie lors de l'insertion de l'annonce dans le système d'information d'Aktor ainsi que sur les sites d'offres d'emplois.

Pour l'exemple précédent, le découpage serait donc :

- **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 integrent l'ensemble des paramètres de la démarche (social, logistique, infrastructures et matériels, informatique) et dessinent le fonctionnement du futur ensemble (Production, methodes 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.

Quelques statistiques du corpus de référence sont rapportées dans le tableau 1.

Nombre d'offres d'emploi	D=1 000	
Nombre total de segments	P=15 621	
Nombre de segments « Titre »	1 000	6,34 %
Nombre de segments « Description »	3 966	25,38 %
Nombre de segments « Mission »	4 401	28,17 %
Nombre de segments « Profil et contacts »	6 263	40,09 %

Tableau 6 : Statistiques du corpus

3.2. Automate de Markov

Les résultats précédents (Kessler et al, 2007) ont montré que la catégorisation de segments sans utiliser leur position dans l'offre d'emploi peut être une source d'erreurs. 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 comme le montre la figure 5 (section 5).

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. Nous avons donc opté pour un automate de Markov à six états : « Début » (S), « Titre » (1), « Description » (2), « Mission » (3), « Profil et contacts » (4), « Fin » (E). On représente une offre d'emploi comme une succession d'états dans cette machine. Nous avons donc parcouru l'ensemble du corpus de référence afin de déterminer les probabilités de transition entre les états. La matrice ci dessous montre ces probabilités.

M =

	Début	Titre	Description	Mission	Profil	Fin
Début	0	0,01	0,99	0	0	0
Titre	0	0,05	0,02	0,94	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	0,00	0,82	0,18
Fin	0	0	0	0	0	0

Matrice de Markov

L'observation de cette matrice M 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 » (Profil et contacts). 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 montré sur la figure 1.

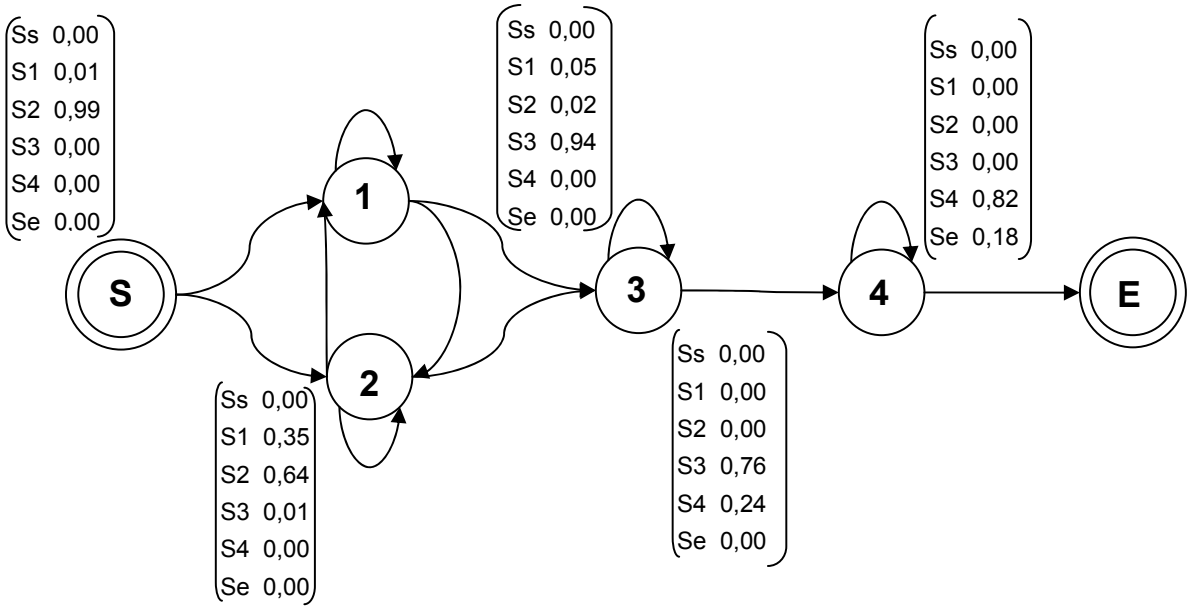

Figure 1 : Automate de Markov

4. Classification

Nous avons choisi deux techniques de classification, SVM et une classification par le produit des probabilités des N-grammes de mots, couplés à un post-processus correctif permettant d'améliorer les résultats obtenus par chacun d'eux.

4.1. Classification par SVM

Nous avons choisi les SVM pour cette tâche car nous avions déjà obtenu de bons résultats lors de travaux précédents sur la classification de courriels (Kessler et al, 2006) ainsi que sur d'autres travaux concernant la classification d'opinions (Torres et al, 2007). Les machines a support vectoriel (SVM) proposées par Vapnik (Vapnik, 1995) permettent de construire un classifieur à valeurs réelles qui découpe le problème de classification en deux sous problèmes : transformation non-linéaire des entrées et choix d'une séparation linéaire optimale. Les données sont d'abord projetées dans un espace de grande dimension H muni d'un produit scalaire où elles sont linéairement séparables selon une transformation basée sur un noyau linéaire, polynomial ou gaussien. Puis dans cet espace transformé, les classes sont séparées par des classifieurs linéaires qui déterminent un hyperplan séparant correctement toutes les données et maximisant la marge.

Elles offrent, en particulier, une bonne approximation du principe de minimisation du risque structurel (c'est-à-dire, trouver une hypothèse h pour laquelle la probabilité que h soit fausse sur un exemple non-vu et extrait aléatoirement du corpus de test soit minimale).

4.2. Classification par N-gramme de mots

Un n-gramme de mots est une séquence de n mots consécutifs. Pour un document donné, on peut générer l'ensemble des n-grammes (n = 1, 2, 3, ...) en déplaçant une fenêtre glissante de n cases sur le corpus. À chaque n-gramme, on associe une fréquence. De nombreux travaux (Damashek, 1995) ont montré l'efficacité de l'approche des n-grammes comme méthode de représentation des textes pour des tâches de classification. L'analyse du corpus nous a permis de déterminer que les offres d'emplois étaient composées de blocs d'informations (voir section 3.1), mais aussi de regroupements d'expressions plus ou moins figées (par exemple, management-équipe pour la classe mission ou encore prétention-salariales pour la classe profil). Afin de pouvoir recenser celles-ci, nous avons construit de façon automatique les collocations afin de nous permettre de calculer la probabilité P associés à chaque n-gramme de mots de notre corpus pour chaque classe. Nous effectuons par la suite le produit des probabilités des N-grammes pour chaque classe *t* afin de déterminer la classe la plus probable.

4.3. Processus correctif

Les résultats obtenus par SVM montrent une classification performante des segments. Pourtant, pendant la classification des offres d'emploi, quelques segments ont été 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, on a appliqué un post-traitement basé sur l'algorithme de Viterbi (Manning et Schütze, 2002 ; Viterbi, 1967). La classification par SVM donne à chaque segment une classe afin de caractériser une offre en entier. Par exemple, pour la

séquence (S)→ (2)→(2)→(1)→(3)→(3)→(4)→(E)[10], l'algorithme classique de Viterbi calculera la probabilité de la séquence. Si la séquence est improbable, Viterbi renvoie 0. Si la séquence a une probabilité nulle le processus correctif renvoie la séquence avec un nombre d'erreur minimal (comparé à la séquence originale produite par SVM) et une probabilité maximale.

```
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.
Si Nombre d'erreur de la séquence en cours > Maximum d'erreur trouvé Alors
   retourne  la séquence en cours
Fin de Si
Si  le symbole courant est le dernier de la séquence Alors
    Si le nombre d'erreurs de la séquence en cours < Maximum d'erreur trouvé Alors
        Maximum d'erreur trouvé = Nombre d'erreur de la séquence en cours;
     Retourne  la séquence en cours;
 Fin de Si
 Sinon
        Pour Chaque symbole de la séquence Faire
            Séquence en cours = symbole suivant ;
            Si  la séquence en cours est la meilleure séquence  Alors
                Meilleure séquence = séquence en cours
                Si le nombre d'erreur de la séquence en cours < Maximum d'erreur trouvé  Alors
                    Maximum d'erreur trouvé = Nombre d'erreur de la séquence en cours ;
                Fin de Si
            Fin de Si
        Fin de Pour Chaque
  Fin de Sinon
```

Algorithme du processus correctif avec la méthode Branch&Bound

Les premiers résultats étaient intéressants, mais avec des temps de traitement assez grands. Nous avons introduit une amélioration en utilisant un algorithme Branch&Bound (Land et Doig, 1960) pour élaguer l'arbre : dès qu'une première solution est trouvée, son erreur et sa probabilité sont retenues et comparées chaque fois qu'une nouvelle séquence est traitée. Si la solution n'est pas meilleure, le reste de la séquence n'est pas calculée. L'utilisation de cet algorithme nous permet d'obtenir la solution optimale, mais peut-être pas le meilleur temps. Notons néanmoins qu'avec cette stratégie, le traitement de séquences contenant 50 symboles avoisine les 2 secondes alors que le parcours complet de l'arbre prenait un temps considérable (plusieurs heures en général).

5. Résultats et discussion

Un corpus de D=1 000 offres d'emploi avec P=15 621 segments a été utilisé. Chaque test a été effectué 20 fois avec une distribution aléatoire entre les corpus de test et d'apprentissage.

La figure 4 montre une comparaison entre les résultats obtenus par les *Support Vector Machines* et le processus correctif. Les courbes présentent le nombre de segments non reconnus en fonction de la taille du corpus d'apprentissage. Elle présente les résultats des SVM seules (ligne pointillée) appliquées sur la tâche de classification des segments. Les résultats sont bons et prouvent que même avec une petite fraction de patrons d'apprentissage

[10] *« Début »→ « Description » → « Description » → « Titre » → « Mission » → « Mission » → « Profil » → « Fin»*

(20% du total), le classifieur SVM obtient un faible taux de patrons mal classés (< 10% d'erreur).

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

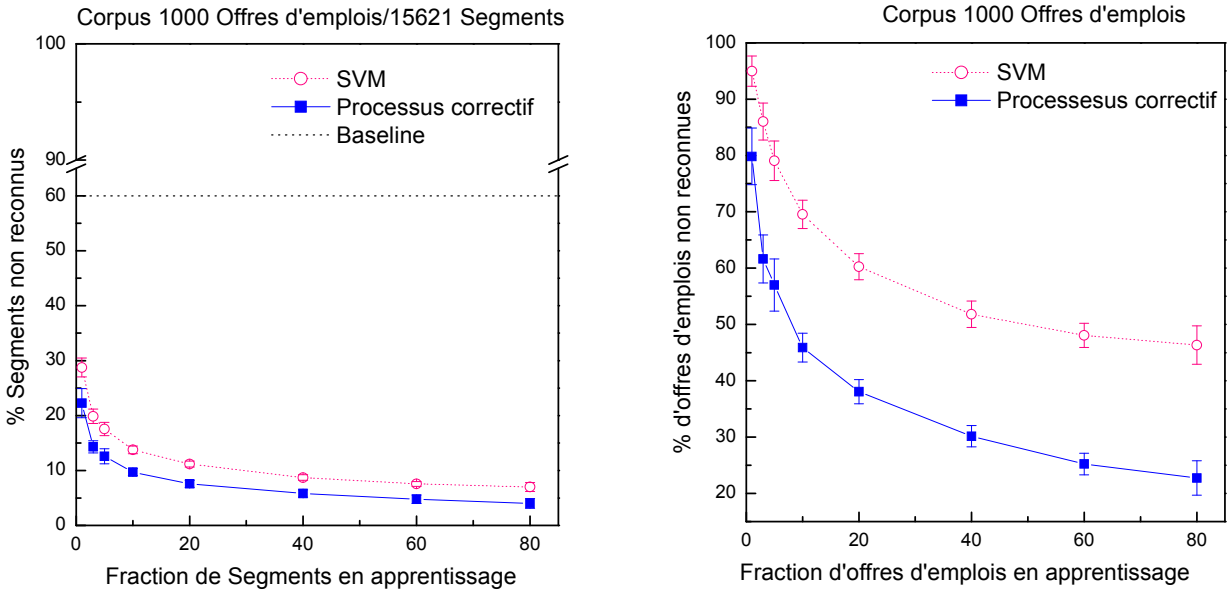

Figure 4 et 5 : À gauche, les résultats des SVM et de l'algorithme correctif par rapport aux nombre de segments mal reconnus. À droite, les résultats des SVM et de l'algorithme correctif par rapport aux offres d'emploi reconnues de façon erronées.

La figure 5, est 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. 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 du 50% du score de SVM.

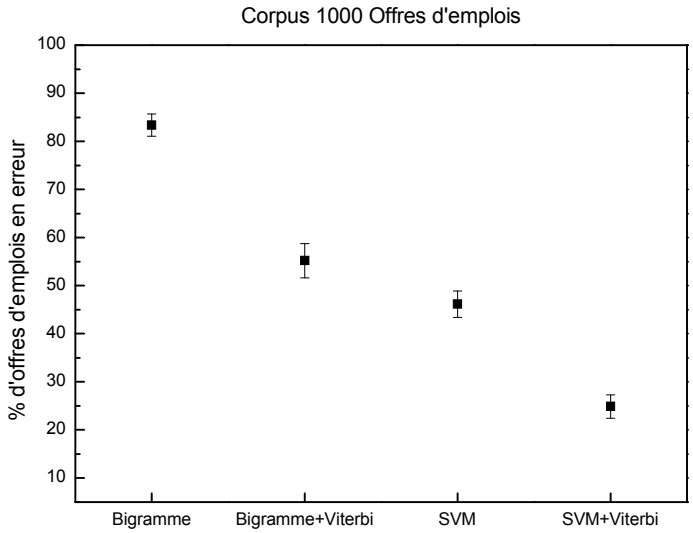

Figure 6 : comparaison des résultats obtenus entre les différentes méthodes de classification avec ou sans processus correctif.

La figure 6 est une comparaison entre les résultats obtenus pour chaque méthode, avec ou sans processus correctif, mais selon les offres d'emploi mal reconnues ou en partie seulement. On observe que le processus correctif améliore les résultats quelques soit l'algorithme de classification (amélioration d'environ 30% pour les bigrammes et d'environ 20% pour les SVM). L'ensemble des tests montre également que la classification par SVM obtient des résultats de meilleure qualité que la classification par bigrammes.

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 7.

Figure 7 : Erreur de bloc frontière

Ainsi, la séquence obtenue pour l'exemple en début d'article (voir section 3.1) est S→2→1→3→3 →3→ 4→4→E et la séquence correcte est S→2→1→3→3→4→4→4→E.

Le segment dont l'étiquetage est faux, est reproduit ci bas :

De formation supérieure Ecole d'ingénieur Chimiste (CPE) option chimie analytique environnementale, vous avez déjà conduit un projet de transfert d'activité.

On voit 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. L'amélioration de la détection du bloc frontière est une des pistes que nous explorons actuellement pour améliorer le système.

6. Conclusion

Le traitement des offres d'emploi est une tâche difficile car l'information y est toujours fortement non structurée. Ces travaux ont 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 SVM étaient très intéressants environ 10% de segments mal étiquetées pour un corpus d'apprentissage de 80%. Le processus correctif améliore ces résultats d'environ 50 % pour chaque méthode de classification (SVM et bigramme) et diminue considérablement les erreurs du type segments isolés incorrectement classés, tout en restant dans des temps de calcul très raisonnables. L'extraction d'informations telle 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 en test 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.

Ces résultats prometteurs nous permettent de continuer le projet E-Gen avec le module d'analyse de pertinence des réponses des candidats. Différentes approches (récupération de l'information et apprentissage) sont envisagées pour résoudre le problème de correspondance entre une candidature et une offre d'emploi. De même, une combinaison (Grilheres et al, 2004) de plusieurs classifieurs (SVM, arbres, modèles probabilistes...) pourrait améliorer les résultats des deux tâchées abordées.

Remerciements

Nous tenons à remercier Juan Manuel Torres-Moreno pour son implication et sa participation active dans le projet ainsi que l'Agence Nationale de la Recherche Technologique (ANRT[11]) et Aktor Interactive, qui ont financé ces travaux (contrat CIFRE numéro 172/2005).

Références

Bellman R (1961). *Adaptive Control Processes.* Princeton University Press.

Bizer C., Heese R., Mocho M., Oldakowski R., Tolksdorf R. and Eckstein R. (2005). The Impact of Semantic Web Technologies on Job Recruitment Processes. *International Conference Wirtschaftsinformatik (WI 2005)*. Bamberg, Germany.

Bourse M., Leclère M., Morin E. and Trichet F. (2004). Human Resource Management and Semantic Web Technologies. *1st International Conference on Information & Communication Technologies : from Theory to Applications (ICTTA)*.

Damashek M. (1995). A gauging similarity with n-grams : Language independent categorization of text. *Science 267*, p.843-848.

El-Bèze M., Torres-Moreno J. M. and Béchet F. (2007). Un duel probabiliste pour départager deux Présidents, *RNTI,* p.1889-1918. (à paraître).

[11] http://www.anrt.fr

Fan R.-E., Chen P.-H. and Lin C.-J. (2005). Towards a Hybrid Abstract Generation System, Working set selection using the second order information for training SVM, p. 1889-1918.

Ferret O., Grau B., Minel J.-L. and Porhiel S. (2001). Repérage de structures thématiques dans des texts, *TALN2001*, Tours.

Grilheres B., Brunessaux S. and Leray P. (2004). Combining classifiers for harmful document filtering, *RIAO*, p. 173-185.

Joachims T. (1999). Making large scale SVM learning practical. Advances in kernel methods : support vector learning, *The MIT Press*, p. 169-184.

Kessler R., Torres-Moreno J. M. and El-Bèze M. (2006). Classification automatique de courriers électroniques par des méthodes mixtes d'apprentissage, *RSTI-ISI,* Vol. 11, p. 93-112.

Kessler R., Torres-Moreno J. M. and El-Bèze M. (2007). E-Gen : Automatic Job Offer Processing system for Human Ressources, *MICAI 2007*, 12 pages. (à paraître).

Land A. H. and Doig A. G. (1960). An Automatic Method of Solving Discrete Programming Problems, *Econometrica* Vol. 28, p. 497-520.

Manning D. C. and Schütze H. (2002). Foundations of Statistical Natural Language Processing. *The MIT Press.*

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

Rafter R., Bradley K. and Smyth B. (2000). Automated Collaborative Filtering Applications for Online Recruitment Services. Lecture Notes in *Computer Science*, p. 363-368.

Rafter R., Bradley K. and Smyth B. (2000). Inferring Relevance Feedback from Server Logs : A Case Study in Online Recruitment, citeseer.ist.psu.edu/382534.html.

Rafter R. and Smyth B. (2002). Passive Profiling from Server Logs in an Online Recruitment Environment, citeseer.ist.psu.edu/rafter01passive.html.

Torres-Moreno J. M., El-Bèze M., Béchet F. and Camelin N. (2007). Comment faire pour que l'opinion forgée à la sortie des urnes soit la bonne ? Application au défi DEFT 2007, *DEFT07*, p. 119-133, Plate-forme AFIA 2007, Grenoble.

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

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