SegCV : traitement efficace de CV avec analyse et correction d'erreurs

Luis Adrián Cabrera-Diego[1,4] Juan-Manuel Torres-Moreno[1,2,3] Marc El-Bèze[1,3]
(1) LIA, Université d'Avignon et des Pays de Vaucluse, France
(2) École Polytechnique de Montréal, Canada
(3) SFR Agorantic UAPV, France
(4) Flejay Group, France
adrian.cabrera@flejay.com ; {juan-manuel.torres, marc.elbeze}@univ-avignon.fr

RÉSUMÉ

Le marché d'offres d'emploi et des candidatures sur Internet a connu, ces derniers temps, une croissance exponentielle. Ceci implique des volumes d'information (majoritairement sous la forme de textes libres) intraitables manuellement. Les CV sont dans des formats très divers : .pdf, .doc, .dvi, .ps, etc., ce qui peut provoquer des erreurs lors de la conversion en texte plein. Nous proposons SegCV, un système qui a pour but l'analyse automatique des CV des candidats. Dans cet article, nous présentons des algorithmes reposant sur une analyse de surface, afin de segmenter les CV de manière précise. Nous avons évalué la segmentation automatique selon des corpus de référence que nous avons constitués. Les expériences préliminaires réalisées sur une grande collection de CV en français avec correction du bruit montrent de bons résultats en précision, rappel et F-Score.

ABSTRACT

SegCV : Eficient parsing of résumés with analysis and correction of errors

Over the last years, the online market of jobs and candidatures offers has reached an exponential growth. This has implied great amounts of information (mainly in a text free style) which cannot be processed manually. The résumés are in several formats : .pdf, .doc, .dvi, .ps, etc., that can provoque errors or noise during the conversion to plain text. We propose SegCV, a system that has as goal the automatic parsing of candidates' résumés. In this article we present the algoritms, which are based over a surface analysis, to segment the résumés in an accurate way. We evaluated the automatic segmentation using a reference corpus that we have created. The preliminary experiments, done over a large collection of résumés in French with noise correction, show good results in precision, recall and F-score.

MOTS-CLÉS : RI, Ressources humaines, traitement de CV, Modèle à base de règles.

KEYWORDS: Information Retrieval, Human Resources, CV Parsing, Rules Model.

1 Introduction

L'accès massif d'internet par les personnes, les institutions et les entreprises a changé radicalement la façon dont fonctionne le marché de l'emploi. De nos jours, des milliers de candidats mettent en ligne leur Curriculum Vitæ (CV), et les entreprises ou les institutions publient des profils de postes recherchés. Analyser automatiquement cette quantité d'informations est une tâche difficile

à accomplir. Ceci est dû, d'un côté à la masse grandissante de CV reçus par les départements de ressources humaines, et d'un autre à l'énorme diversité de la présentation des CV. En particulier, dans certains sections (identité, formation, expérience et compétences) et leur organisation. Si on ne peut pas parler vraiment de documents « non-structurés », on peut les qualifier de « trop librement structurés », répondant à une structure conceptuelle propre à chaque individu et difficile à modéliser. Nous nous situons dans la double perspective d'emplois académiques et commerciaux. L'employeur est ici une institution (université, grande école, centre de recherche) ou une entreprise, et les candidats présentant des dossiers adaptés pour correspondre au mieux aux profils recherchés. Donc, nous projetons de concevoir un système intégral d'analyse des candidatures académiques ou commerciales, dont la première étape consiste dans le découpage des CV des candidats.

La problématique qui aborde SegCV est plus générale que celle étudiée auparavant [6, 7, 3], car ces travaux analysent seulement des CV commerciaux. SegCV est composé des modules suivants : Extraction d'information à partir des CV en formats PDF, Word, Open Office, PS, DVI et RTF ; analyse des CV pour extraire les sections importantes. Cet article présente un système de découpage automatique des CV ainsi qu'une étude portant sur la correction d'erreurs lors de la transformation en format texte. Nous présentons en section 2 la stratégie mise en œuvre. En Section 3, sont décrits les corpus utilisés. Nous présentons, en Section 4, la méthode pour détecter et corriger les erreurs avec deux modèles basés sur des *n*-grammes. En section 6, sont détaillés les différents résultats obtenus avant de conclure.

2 Méthodologie

Nous présentons la première étape d'un analyseur automatique d'offres et de demandes d'emploi : un analyseur des CV basé sur le contexte. En fonction des sections définies comme étant importantes par le recruteur, le système extrait l'information pertinente du CV, puis génère un fichier avec le contexte et la granularité voulue. L'analyseur est essentiellement basé sur un nombre restreint de règles dépendantes de chaque langue. Il transforme l'information des CV en blocs d'information selon des modèles définis par l'utilisateur, faciles à comprendre par les humains et exploitables par les machines.

Les CV originaux sont déclinés en formats divers : .doc, .odt, .pdf, .ps, .txt, etc. Afin de pouvoir les traiter convenablement, les CV sont transformés en texte utf-8. Cependant, cette transformation n'est pas libre d'erreurs, surtout dans les fichiers issus de PDF. Nous considérons le bruit comme la différence entre la forme superficielle d'une représentation textuelle et le texte prévu, correct ou originel [8]. Si la source est PostScript ou PDF du LaTeX, le texte extrait peut comporter un certain nombre d'erreurs. Les caractères accentués, la police utilisée et les petites majuscules sont des sources d'erreurs récurrentes et difficiles à modéliser. Or, les fichiers générés par LaTeX risquent d'être très fréquents dans les CV issus du milieu académique. Cette étape du pré-traitement est souvent négligée alors qu'elle a un fort impact dans des étapes ultérieures. En effet, le découpage des CV (tâche déjà difficile du fait de la variabilité évoquée) peut être un vrai casse-tête si l'on tient compte du bruit introduit par les convertisseurs PDF à texte.

3 Corpus

Nous avons constitué un corpus de 100 CV en français issus du domaine commercial. Ce corpus a été découpé à la main par 2 annotateurs. Les annotateurs ont reçu des consignes strictes quant

au découpage des sections, selon un manuel fourni :
– Identité (coordonnées du candidat) ; Résumé ; Poste demandé (information qui décrit le poste demandé) ; Situation actuelle du candidat ; Autres (loisirs, les références, etc.).
– Formation (formation universitaire) ; formation additionnelle (diplômes ou certifications).
– Expérience (expérience professionnelle).
– Compétences (compétences ou aptitudes personnelles, les langues étrangères, les outils maîtrisés, etc).

Nous appelons ce corpus étalon CD. Pour les tests de découpage, nous avons constitué le corpus CN, composé des mêmes 100 CV, mais sans le découpage manuel.

Pour étudier le bruit, nous disposons d'un corpus de 750 CV commerciaux, provenant de fichiers .doc, .odt et .rtf, pour lesquels la conversion, en théorie, n'a généré aucune erreur [1]. Ce corpus sera nommé CVcomm. En ce qui concerne les CV académiques, la question est plus délicate : La plupart de CV sont bruités, et les dé-bruiter manuellement serait une tâche pénible et pas exempte d'erreurs. Cependant, nous avons détecté 8 CV sans bruit, qui seront utilisés lors de tests. Ce corpus sera nommé CVac.

4 Détection et correction du bruit

La transformation des CV en texte peut générer plusieurs erreurs : l'introduction de caractères composés, de caractères superposés, la séparation des caractères ou l'ajout des espaces entre caractères. En général, tous les cas, à exception du dernier, peuvent être corrigés en utilisant des expressions régulières car ces erreurs suivent des patrons réguliers. Cependant le problème d'ajout d'espaces entre les caractères semble être de nature aléatoire. Parfois, ce type d'erreur est occasionné par l'utilisation de caractères accentués, de majuscules ou par l'utilisation d'un format particulier des documents. Les blancs peuvent être présents plusieurs fois dans le même mot ou dans la même ligne. Ces espaces placés au milieu de mots peuvent empêcher le découpage correct des sections.

Pour bien mener nos tests, à partir du corpus CVcomm, nous construisons à tour de rôle 5 sous-corpus qui seront utilisés comme suit : 4/5 sous-corpus seront employés pour le calcul des n-grammes et 1/5 pour la phase de tests. Il faut dire que la génération des n-grammes est enrichie d'un ensemble T de textes sans bruit : romans, livres scientifiques et discours composé de 784k mots. Pour les tests, nous avons bruité le 1/5 du corpus avec des espaces en blanc introduits de façon aléatoire. Afin d'injecter chaque espace, nous avons généré 3 numéros aléatoires : le premier fixe la ligne du fichier à bruiter, le deuxième le mot et le troisième la position à l'intérieur du mot (en évitant les extrêmes). Le bruit injecté est donc à pourcentage variable [2]. Nous appellerons ces corpus $CB_{(i=0,5,10,15,...,100)}$. Pour les CV académiques, nous utiliserons le corpus CVac comme référence afin de tester le correcteur. Ainsi nous avons ajouté du bruit à l'ensemble CVac suivant la même procédure qu'auparavant. Les correcteurs utilisent tous les n-grammes générés avec les CV commerciaux plus les documents de l'ensemble T afin de debruiter les CV de CVac.

La correction d'erreurs est une tâche généralement abordée dans la reconnaissance optique de caractères (OCR) ou dans le traitement d'information informelle, comme les blogs, les forums, les SMS ou les tchats. Les travaux concernant la correction de bruit [2, 9, 1, 4] traitent la correction

1. Grâce à la codification homogène des éditeurs (Word, Libre/OpenOffice)
2. Nous considérons un mot comme l'ensemble de caractères entre deux espaces

de fautes d'orthographe et grammaticales, la mauvais ponctuation ou l'utilisation d'abréviations. Mais le problème spécifique des blancs a été peu traité à notre connaissance. Pour résoudre ce problème, nous proposons deux stratégies à base de n-grammes de caractères : un correcteur binaire et un autre probabiliste.

4.1 Correcteur binaire

L'algorithme utilise des n-grammes de caractères avec $n = 4, .., 7$. Ces n-grammes ont comme caractéristique principale la présence, d'au moins, un espace entre deux caractères ([a-zA-Z], caractères accentués ou l'apostrophe). Pour chaque ligne avec au moins un espace, on génère le n-gramme le plus grand possible avec un espace en son centre. Le n-gramme original et ses voisins à gauche et à droite du centre, sont recherchés.

Le n-gramme père est considéré comme correct (l'espace central doit être conservé), si lui ou ses fils, remplissent au moins une des conditions suivantes : i/ le 7-gramme existe ; ii/ deux 6-grammes existent ; iii/ au moins deux 5-grammes existent ; iv/ Deux 4-grammes existent (zone encadrée en pontillé de la figure 1). Ces conditions se basent sur l'idée qu'un n-gramme père avec un espace central engendre deux 6-grammes, trois 5-grammes et deux 4-grammes. Si la majorité de ses fils existent, il est probable que le n-gramme père soit correct. Si le père est considéré comme incorrect, il faut analyser la classe à laquelle il appartient. Les cas et les corrections dépendent du nombre d'espaces après ou avant l'espace central, du nombre de caractères à droite et à gauche ou si le n-gramme est au début ou à la fin d'une ligne. Les corrections sont des règles permettant l'élimination de blancs gênants. L'algorithme de correction peut être exécuté itératif afin de corriger des erreurs non trouvées lors des corrections précédentes.

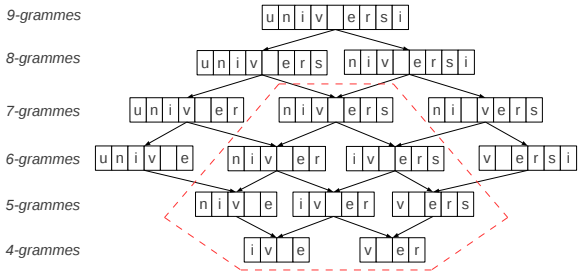

FIGURE 1: Exemple de n-grammes pour les correcteurs.

4.2 Correcteur probabiliste

Afin d'obtenir des performances plus robustes et de meilleurs résultats, nous avons développé un correcteur probabiliste. Le principe étant proche de celui binaire, sauf que le parcours des branches sera conditionné par la probabilité des n-grammes. L'algorithme construit le n-gramme le plus grand possible ($n = 4, ..., 9$) ayant un espace central entre deux caractères. Nous énumérons toutes les combinaisons de n-grammes en éliminant ou en maintenant les espaces qu'ils contenant. Des caractères à droite peuvent être ajoutés pour maintenir la taille et le contexte du n-gramme le plus grand possible. Puis on calcule les probabilités conditionnelles

de chaque combinaison en utilisant l'estimation du maximum de vraisemblance :

$$P\left(c_i|n_{i-1}\right) = \frac{C\left(n_{i-1}c_i\right)}{C\left(n_{i-1}\right)} = \frac{C\left(n_i\right)}{C\left(n_{i-1}\right)} \quad (1)$$

où c_i est le dernier caractère du n-gramme de taille i, $C\left(n_i\right)$ et $C\left(n_{i-1}\right)$ sont leurs fréquences. Si la probabilité de toutes les combinaisons du n-gramme sont nulles, la taille du n-gramme est diminuée en 1 caractère (figure 1) et le processus itère à nouveau. Autrement on considère comme une correction acceptable la combinaison ayant la probabilité conditionnelle la plus grande.

5 Découpage en sections

La tâche principale de SegCV consiste à repérer, découper et regrouper les sections pertinentes des CV. À cette fin, on peut être tenté d'utiliser des méthodes d'apprentissage automatique, car on sait qu'elles donnent de très bons résultats sur les tâches de TALN. Mais l'apprentissage automatique nécessite une grande quantité de documents préalablement étiquetés. Or, nous ne disposons pas d'un grand corpus annoté manuellement. En conséquence, nous avons deux possibilités pour faire face à ce problème. La première consiste à faire un découpage à de tailles fixes (1/3, 2/3, etc.), comme proposé par [5], mais cette approche nous semble trop grossière. L'autre possibilité consiste à établir des règles de découpage. Notre objectif étant de découper les CV de la manière la plus fine possible, nous avons décidé d'utiliser des règles.

À cette fin, nous avons suivi deux approches. La première est basée sur la structure du CV : les titres, les sous-titres ou les débuts des lignes avec un symbole délimitant une section. 94 expressions régulières composent ces règles. La deuxième approche essaie d'améliorer le découpage au moyen de mots-clés qui seront recherchés à l'intérieur des sections. Le découpage fait appel à un prétraitement (élimination ou normalisation de symboles et la normalisation d'espaces), puis, les règles de structure sont appliquées. Après ce premier découpage, nous vérifions la taille des sections trouvées. Si elle est anormalment grande (ou petite) par rapport à la taille du CV nous faisons appel aux mots-clés pour déclancher une procédure de déplacement de l'information. Par exemple, si un fragment de texte dans la section « Compétences » contient les mots *célibataire* ou *situation de famille* ce fragment sera déplacé à la section « Identité ».

6 Résultats

Nous avons effectué trois expériences pour évaluer le découpage automatique et la correction du bruit. Nous avons décidé d'utiliser des mesures de similarité et non pas des mesures basées sur les frontières du découpage car l'information dans les sections peut être éparpillée. Puisque les CV sont des fichiers trop librement structurés, les limites exactes des sections sont difficiles à repérer. Si l'on ajoute du bruit, ces frontières sont souvent perdues. Essayer de trouver les frontières exactes est alors un exercice très délicat et imprécis. Nous avons décidé donc de mesurer la pertinence du découpage par le contenu des sections, plutôt que par les frontières. A ce fin, nous avons utilisé deux mesures de similarité entre le découpage manuel et celui automatique : la similarité cosinus et une mesure de divergence de Kullback-Leiber modifiée (issue du domaine du résumé automatique). Une section sera considérée comme correctement découpée si sa similarité dépasse un certain seuil. Le seuil peut être strict (similarité $= 1$) ou relaxé ($0,95 <$ similarité $< 0,5$). Ensuite nous avons calculé la précision, le rappel et le F-score.

Pour évaluer la correction du bruit, nous avons comparé le nombre de mots corrects dans le fichier corrigé par rapport au nombre de mots dans le fichier d'origine. Formellement, la précision et le rappel ont été définis de façon classique comme suit :

$$\text{Précision} = \frac{C_C}{T_C} \qquad \text{Rappel} = \frac{C_C}{T_O} \tag{2}$$

où, C_C est le nombre de mots corrects dans le fichier corrigé, T_C le nombre de mots dans le fichier corrigé et T_O le nombre de mots dans le fichier d'origine.

Découpage automatique. La première expérience a consisté à découper automatiquement les fichiers du corpus CN. La figure 2 montre le F-score pour les deux mesures de similarité.

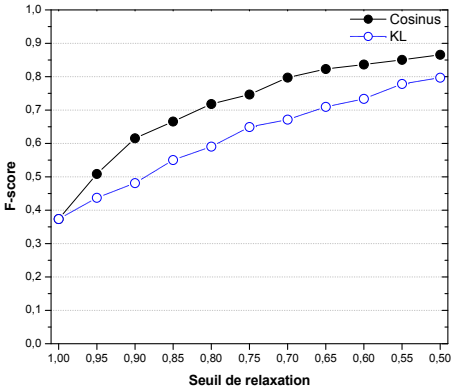

FIGURE 2: Découpage de CV : F-score

Le système ne découpe pas les sections avec une grande précision. Les raisons de ce problème sont variées. D'abord, les annotateurs ont évité les informations inutiles (numéros de page, en-têtes ou les pieds de page), ce qui le système ne fait pas encore. Ensuite, la perte de la structure du CV (comme les tables ou les colonnes) produit une mélange erronée de l'information. Et finalement, les règles de mots-clés peuvent déplacer incorrectement l'information d'une section.

Correction du bruit. Nous avons simulé le bruit par ajout aléatoire de blancs au milieu des mots. La quantité d'espaces a été déterminée par la taille du fichier d'origine et par un pourcentage variable (0 %, 5 %, 10 %...100 %). Les correcteurs binaire et probabiliste ont été appliqués itératif trois fois. Au delà de la troisième application, les résultats n'ont guère changé. Pour l'évaluation des corpus bruités CB_i, nous nous sommes servis des corpus de référence. La figure 3 montre le F-score pour cette expérience mesuré sur des CV commerciaux à gauche et académiques à droite. Les résultats montrent que le correcteur binaire fonctionne assez mal, même pour des quantités minimales de bruit. A 50 % de bruit, le correcteur probabiliste obtient un F-score de 0,82 (CV commerciaux) et de 0,75 (CV académiques). Pour un taux de bruit de 100 % le correcteur probabiliste obtient un F-score de 0,80 (commerciaux) et de 0,71 (académiques). Il faut dire que la quantité de bruit dans les cas réels n'est pas si élevée, mais nous voulions tester nos correcteurs dans les cas extrêmes.

Découpage automatique plus correction de bruit La dernière expérience a consisté à segmenter automatiquement le corpus CD. Mais cette fois, nous y avons ajouté du bruit de manière

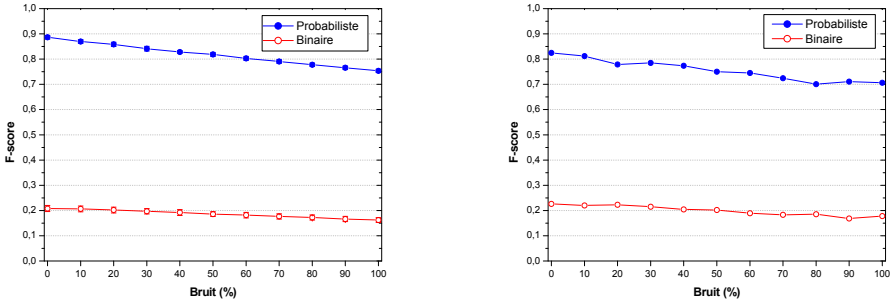

FIGURE 3: Correction de l'injection de bruit : à gauche CV commerciaux, à droite CV académiques

aléatoire (de la même façon que pour le CB), en utilisant le correcteur probabiliste, appliqué 3 fois, pour le diminuer. Nous avons évalué la qualité du découpage avec la mesure de cosinus. La figure 4 montre la surface de F-score en fonction du pourcentage du bruit et du seuil de relaxation. Les résultats obtenus indiquent que l'utilisation du correcteur impacte la qualité

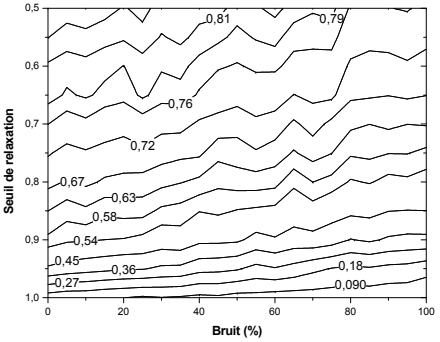

FIGURE 4: Découpage automatique avec correction de bruit : F-score

de découpage. Pour un pourcentage de zéro bruit ajouté et un seuil de relaxation égal à 1,00, c'est-à-dire une similarité exacte, le découpage automatique avec correction probabiliste obtient un F-score = 0,13, contre un F-score = 0,37 du découpage sans correctif. Nonobstant, pour un niveau de bruit nul et un seuil de relaxation égal à 0,50, le F-score pour le premier est de 0,84 et de 0,79 sans correctif. Pour un niveau de bruit égal à 50 % et un seuil de relaxation de 50 %, le découpage automatique avec correction obtient un F-score de 0,796.

7 Conclusions et perspectives

L'analyse automatique de CV est une tâche extrêmement difficile. Ceci s'explique par plusieurs raisons, dont la principale est la structure des CV : malgré une structure conventionnelle, l'information présente dans les CV est en format libre. En outre, ils sont produits en plusieurs formats électroniques. Leur transformation peut occasionner des erreurs ou perte d'information. Le vocabulaire utilisé peut varier énormément au niveau des CV ou des profils. Dans ce travail,

nous avons présenté la première étape d'un système d'analyse automatique des CV. Nous avons présenté un module pour découper des CV en français et un module pour corriger les erreurs générées à cause de la transformation du fichier d'origine. Les expériences réalisées montrent que le découpage automatique doit être amélioré pour se rapprocher plus du découpage manuel. Par contre, la correction de bruit a montré de très bons résultats. Nous avons vérifié que la méthode probabiliste corrective donne les meilleurs résultats. Cependant, il faut éviter la correction de fichiers non bruités, car, en effet, il semble que la correction de faux positifs génère une diminution de la qualité du découpage. À l'avenir, nous voulons augmenter la qualité de nos modules et les appliquer dans de corpus académiques de taille plus conséquente. Pour le découpage automatique, nous pensons ajouter un module de nettoyage afin d'éliminer les numéros de pages, les en-têtes et les pieds de page. De la même façon, il sera intéressant d'effectuer des expériences avec des CV dans des langues autres que le français.

Remerciements

Ce travail a été financé par la convention ANRT-CIFRE n° 2012/0293 entre Flejay et l'UAPV.

Références

[1] Sumeet Agarwal, Shantanu Godbole, Diwakar Punjani, and Shourya Roy. How much noise is too much : A study in automatic text classification. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 3–12. IEEE, 2007.

[2] Alexander Clark. Pre-processing very noisy text. In *Proc. of Workshop on Shallow Processing of Large Corpora*, pages 12–22, 2003.

[3] Jérémy Clech and Djamel A. Zighed. Data mining et analyse des cv : une expérience et des perspectives. In *Extraction et la Gestion des Connaissances, EGC'03*, pages 189–200, 2003.

[4] Lipika Dey and SK Mirajul Haque. Opinion mining from noisy text data. *International Journal on Document Analysis and Recognition (IJDAR)*, 12(3) :205–226, 2009.

[5] Rémy Kessler, Nicolas Béchet, Mathieu Roche, Marc El-Bèze, and Juan-Manuel Torres-Moreno. Automatic profiling system for ranking candidates answers in human resources. In *OTM '08 Monterrey, Mexico*, pages 625–634, 2008.

[6] Rémy Kessler, Juan-Manuel Torres-Moreno, and Marc El-Bèze. E-Gen : Automatic Job Offer Processing system for Human Ressources. In *MICAI*, pages 985–995, 2007.

[7] Rémy Kessler, Juan-Manuel Torres-Moreno, and Marc El-Bèze. E-Gen : Profilage automatique de candidatures. In *TALN'08 Avignon*, 2008.

[8] Craig Knoblock, Daniel Lopresti, Shourya Roy, and L.Venkata Subramaniam. Special issue on noisy text analytics. *IJDAR*, 10(3-4) :127–128, 2007.

[9] Benoît Sagot, Pierre Boullier, et al. Sxpipe 2 : architecture pour le traitement pré-syntaxique de corpus bruts. *Traitement Automatique des Langues*, 49(2) :155–188, 2008.