Université de Montréal

Génération automatique de lettres de recrutement

par
Philippe Grand'Maison

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Décembre, 2016

RÉSUMÉ

Ce mémoire de maîtrise présente le développement d'un système de génération de la langue naturelle pour automatiser les lettres de contact envoyées par les chasseurs de tête. Les travaux de Ehud Reiter ont inspiré la portion de génération de texte. La génération du contenu est basée sur des règles d'associations obtenues par l'analyse statistique d'une base de données de profils LinkedIn. Le système écrit des lettres en anglais mais peut être facilement étendu à la langue française.

Ce projet s'inscrit dans le cadre du Butterfly Predictive Project, une collaboration entre l'Université de Montréal et LittleBIGJob.

Mots clés: Génération automatique de texte, forage de données, ressources humaines, recrutement.

ABSTRACT

This master's thesis presents the development of a Natural Language Generation system designed to automate the writing of first-contact letters by professional headhunters. A top-down approach modelled on Ehud Reiter's work handles the Natural Language portion of the system. Content generation is based on association rules obtained by statistical analysis of a large database of LinkedIn profiles. The system writes English letters but can easily be extended to French.

This project is part of the Butterfly Predictive Project, a collaboration between Université de Montréal and LittleBIGJob.

Keywords: Natural Language Generation, data mining, human resources.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

REMERCIEMENTS

Je tiens à remercier :

Guy Lapalme, pour ses nécessaires conseils et corrections, et pour son soutien continu ;

Fabrizio Gotti, pour ces discussions techniques, aussi productives que fascinantes ;

Isabelle Archer-Vézina, amie, complice et bientôt épouse ;

et toute l'équipe du RALI.

CHAPITRE 1

INTRODUCTION

1.1 *Butterfly Predictive Project*

Le problème auquel ce projet veut répondre est issu d'une collaboration, avec un partenaire commercial du monde des ressources humaines. Le projet a été baptisé *Butterfly Predictive Project* (*BPP*). *LittleBigJob* est une firme de recrutement dont les activités chevauchent l'Atlantique : elle offre ses services aux citoyens et entreprises tant canadiens que français. La société a approché le RALI pour mettre à profit les techniques de traitement de données textuelles.

Le *Butterfly Predictive Project* vise à développer des outils pour gagner des intuitions statistiques, notamment par la génération de tableaux de bord et autres visualisations ; pour classifier automatiquement les candidats en catégories intéressantes pour les recruteurs ; pour apparier des candidats aux offres d'emploi ; et pour générer des lettres de recrutement. Le projet dispose de millions de profils glanés des réseaux sociaux professionnels, plus particulièrement de *LinkedIn*, de centaines de milliers d'offres d'emplois et d'une base de données d'entreprises.

Ce mémoire décrit un logiciel de génération automatique de texte dans le domaine du recrutement. Plus particulièrement, nous souhaitons générer automatiquement des lettres de motivation envoyées par les chasseurs de tête aux candidats.

1.2 Lettres de motivation

Le travail du chasseur de tête est d'identifier des candidats pour une offre d'emploi donnée, de sélectionner les profils qui sont à la fois qualifiés et susceptibles de poser leur candidature, et de les contacter.

En ce moment, c'est le personnel de l'agence de recrutement qui accomplit chacune de ces étapes, ce qui entraîne des coûts importants que l'automatisation peut réduire.

Le processus est enclenché lorsque l'agence reçoit une description du poste à combler d'un client. L'agence développe des processus pour la recherche manuelle de candidats potentiels. Le personnel consulte chacun des candidats et pose un jugement sur son adéquation à la description du poste, et sur la probabilité du candidat de s'engager dans la démarche de recrutement. Les candidats retenus sont contactés par téléphone ou courriel. Le courriel de contact est une lettre standard adaptée manuellement par le recruteur. Cette lettre, nous l'appelons la *lettre de motivation*. Le processus est similaire sur *LinkedIn* : au lieu des courriels, on utilise le système de messagerie interne.

Un membre de l'équipe de *Butterfly Predictive Project* a conçu un algorithme qui sélectionne une liste de profils de réseaux sociaux pour une offre donnée. L'algorithme ordonne ensuite les candidats. Le mémoire de Dieng Mamadou Alimou [11] contient les détails de cet appariement. La tâche que nous nous proposons de résoudre fait suite à cet appariement : nous désirons automatiser le contact avec le candidat passif, c'est-à-dire celui qui n'est pas activement à la recherche d'un emploi mais qui pourrait répondre à une offre. Rappelons que ce contact est déjà partiellement automatisé chez LBJ : une lettre avec un patron très simple est utilisée.

La solution que nous proposons, nommée BPPGen, adapte le contenu des lettres générées au profil et à l'offre d'une manière non-triviale. Le but de cette personnalisation est de maximiser le nombre de candidats potentiels qui répondent à la lettre. BPPGen réduit aussi le temps de rédaction des lettres. Ce système s'appuie sur les données mises à la disposition du *Butterfly Predictive Project*.

1.3 Exemple

Voici un exemple du type de lettres générées. Les entrées sont :

- une offre pour un poste de MARKETING MANAGER pour ACME INC. Restaurant Group ;
- le profil LinkedIn d'un candidat potentiel qualifié.

Exemple 1.1: Un exemple de lettre

Dear Mr. John Doe,

I write this email after having read your LinkedIn profile. My name is Joan, I am recruiter with LittleBIGJob. Our company uses artificial intelligence to make data-driven decisions in human resources. I only contact the best candidates based on fancy statistical techniques, and you've made the short list.

The position you are being considered for is Marketing Manager with ACME INC. Restaurant Group. Let me tell you why I would recommend you.

You are an expert in Marketing Management, Management and Marketing, which is exactly what my client is looking for. You seem pretty qualified because you have held the identical occupation of Marketing Manager once and you are proficient in Public Relations, Marketing Strategy, Advertising and Sponsorship. You are a wonderful fit. Your experience as Executive Director suggests that you know a thing or two about Event Management and I suppose that your experience as General Manager has taught you a lot about Sales. I am not a specialist but I figure that your previous work as Communications Consultant has taught you all there is to know about Corporate Communications.

Would you be open to discuss this over the phone ? When would you be available ?

Thank you for your time and your reply,

1.4 Moyens persuasifs

L'objectif poursuivi par BPPGen est de persuader les candidats pressentis à s'engager dans le processus de recrutement. Nous jugerons que cet objectif est satisfait lorsque le destinataire cliquera sur le lien fourni au bas de la lettre.

BPPGen emploie les moyens suivants pour persuader l'individu :

1. utiliser un vocabulaire varié ;
2. présenter l'offre d'une manière qui tienne compte du parcours professionnel de l'individu ;
3. simuler un intérêt porté envers sa vie professionnelle ;
4. imiter le style auquel est habitué l'humain.

2 et 3 peuvent paraître similaires, et sont certainement co-dépendants, mais doivent être distingués : 3 nous propose de tirer profit des informations disponibles sur le candidat et sur l'ensemble des informations contenues dans la base de profil ; 3 est un moyen rhétorique qui utilise 2. Le moyen 3 impose une restriction importante : il ne faut pas que l'origine purement automatisée de la lettre soit trop apparente.

L'exemple 1.1 remplit ces objectifs. Au deuxième paragraphe, un vocabulaire différent est utilisé pour exprimer le fait que le candidat a certaines compétences pertinentes : les deux premières phrases disent la même chose de compétences différentes. Le moyen 2 est également respecté au second paragraphe : les expériences passées du candidat sont mentionnées. Le moyen 3 passe par la mention de plusieurs faits relatifs à la vie professionnelle du candidat, en discutant principalement des qualifications du candidat. Le moyen 4 est surtout apparent lorsque plusieurs lettres sont écrites : la même information est alors exprimée de différentes manières, en combinant des segments de phrases.

Le problème serait alors visible si la même personne recevait des lettres pour deux offres d'emploi différentes.

1.5 Organisation de ce mémoire

Le mémoire est divisé en sept chapitres. Le chapitre 2 discute des générateurs automatiques de la langue en général, en nous concentrant sur l'architecture popularisée par l'oeuvre de Ehud Reiter (p.12), et sans négliger de présenter d'autres systèmes. Au chapitre 3, nous présenterons l'architecture de BPPGen, en illustrant chaque étape avec l'exemple 1.1. Au chapitre 4, nous décrirons l'utilisation des données amassées pour le projet *BPP*. Après avoir présenté le fonctionnement global du système et le type de données utilisées, nous présenterons en plus de détail les aspects saillants de BPPGen au chapitre 5. Nous présenterons un exemple récapitulatif au chapitre 6. Nous concluons par une brève discussion des expériences menées dans le projet et des différentes améliorations qu'on peut apporter au système (chapitre 8).

Avant de nous lancer dans le vif du sujet, quelques précisions sur le vocabulaire sont de rigueur, pour alléger la lecture et établir les correspondances aux termes anglais plus généralement utilisés.

1.6 Vocabulaire

Nous écrirons *GAT* (anglais : *NLG*) plutôt que *Générateur automatique de texte*. Si le contexte est clair, nous utiliserons les mêmes initiales pour *Génération automatique de la langue* (anglais : *Natural language generation*).

Nous dirons *TAL* (anglais : *NLP*) au lieu de *Traitement automatique de la langue* (anglais : *Natural language processing*), et *CLN* (anglais : *NLU*) pour *Compréhension du langage naturel* (anglais : *Natural language understanding*).

La génération de texte sera toujours supposée automatique. Nous ne nous intéressons pas ici au processus cognitif humain. De même, *générer un texte* signifie *produire un texte avec l'aide d'un logiciel de génération automatique de texte* si le sujet est humain ; ce sera *produire un texte* si le sujet est un logiciel.

Nous dirons qu'un texte est *humain* s'il est écrit par un humain. Nous dirons qu'il est *automatique*, ou *généré* s'il a été généré par un GAT.

Par *profil*, nous entendons *profil de réseau social du candidat potentiel* ; et lorsque nous mentionnerons le *candidat*, il est entendu que l'individu visé n'est pas candidat, mais un candidat potentiel. Cette terminologie est issue des ressources humaines, qui distinguent les candidats actifs (activement à la recherche d'un emploi) des candidats passifs, qui ne sont pas à la recherche d'emploi et donc improprement appelés candidats.

Par *recruteur*, on doit lire *chasseur de tête*[1], un recruteur externe chargé de contacter des candidats pour les intéresser à poser leur candidature pour un poste.

Une OCCUPATION est un type d'activité pratiquée par un ensemble de personnes. Une OCCUPATION peut avoir plusieurs *titres*. Un *emploi* (on dira aussi *poste*) est l'activité pratiquée par un travailleur ; on suppose que les emplois sont des instances d'une occupation. Dans la pratique, nous confondrons *titre* et *occupation* lorsque le contexte suffira à les distinguer.

Du côté de la linguistique, nous devons aussi clarifier certains termes. En particulier, nous utiliserons telle quelle l'expression anglaise *token* pour désigner la plus petite unité de sens rencontrée dans un texte. Une traduction ne serait pas cohérente avec l'expression bien établie de *tokénisation*, l'action consistant à transformer le texte en séquence de tokens.

Finalement, l'emploi du masculin pour les noms d'emploi est préféré pour alléger la lecture. L'anglais est utilisé pour la majorité des exemples en raison des meilleurs

[1] On dit aussi *chasseur de talents*, *chasseur de cadres*.

résultats obtenus dans cette langue.

CHAPITRE 2

GÉNÉRATION AUTOMATIQUE DE TEXTE

2.1 Quoi dire ? Comment le dire ?

Les GAT accomplissent deux tâches principales : déterminer le contenu à communiquer, puis produire le texte qui le communique.

2.1.1 Pourquoi pas des patrons ?

S'il suffisait qu'un logiciel produise du texte pour qu'il soit un GAT, la plupart des logiciels mériteraient l'appellation : pratiquement tous les programmes décemment écrits enregistrent certains événements, comme les erreurs, dans des registres (logs). Ces enregistrements sont écrits à l'intention des humains, et contiennent souvent des informations variables indiquant une valeur d'intérêt lors de l'exécution du programme. Ils sont d'une certaine manière des générateurs de texte par patrons.

Par exemple, *NameError : name 'a' is not defined* indique au programmeur qu'il référence erronément une variable, *a*, sans l'avoir préalablement définie. Le message a un contenu variable, le nom de la variable non-définie. La plupart des pages Web sont générées de cette façon.

Générer des messages de ce type est extrêmement courant, rapide d'écriture et d'exécution. Quels avantages pourraient donc présenter les GAT ?

Un des buts des GAT est de pouvoir représenter des données d'un autre système de manière flexible : la nature de l'application devrait être maximalement découplée d'un problème particulier. Les systèmes de patrons peuvent aussi être rendus très flexibles par composition conditionnelle de patrons, mais demandent alors un effort de programmation explicite qui peut devenir fastidieux. Cette étape de sélection du contenu à présenter

est souvent une motivation principale des GAT [17] : les GAT ne sont pas seulement responsables de la présentation du contenu, mais sont souvent associés à un système de sélection de contenu.

Une autre avantage des GAT est de produire un texte plus naturel. Pour reprendre l'exemple des logs, il n'est pas rare de voir un grand nombre de fois le même message être généré à une variable près : Utilisateur espion007 échoue à se connecter au poste 4063 à 16 :04 :23.

...

Utilisateur espion007 échoue à se connecter au poste 6061 à 16 :06 :26.

Utilisateur espion007 échoue à se connecter au poste 6062 à 16 :06 :27.

Utilisateur espion007 échoue à se connecter au poste 6063 à 16 :06 :27.

Plutôt que de lire deux mille phrases identiques, il pourrait être préférable de lire une phrase contenant une forme d'agrégation, comme : Utilisateur espion007 échoue à se connecter aux postes 4043 à 6063 entre 16 :04 et 16 :07.

Ces agrégations qui permettent de réduire la redondance sont un aspect fréquent des GAT [17][1]. Il serait possible d'obtenir des résultats équivalents avec un système de patrons suffisamment puissants[2], mais développer un tel outil pourrait devenir aussi fastidieux que développer un GAT sur mesure.

Un autre avantage est que ces systèmes peuvent aisément être modifiés pour être multilingues. Si les langues visées sont similaires, il suffit de modifier l'engin lexical et le réalisateur.

[1] L'exemple de l'analyse et la génération automatiques de rapports d'incidents fondés sur les logs de logiciels informatiques n'est pas fortuit : des logiciels incontournables aujourd'hui dans l'administration de serveurs ne font que cela. Nommons *Logstash*, un projet Open Source entièrement dédié à l'analyse des logs.

[2] Comme MailMerge selon [18, p. 27], ou des systèmes plus récents comme *XSLT*, *Velocity*, *Tiles* ou *Thymeleaf*, qui sont des langages de programmation à part entière.

2.1.2 Pourquoi pas des générateurs par modèle de langue ?

Un modèle de langue est une distribution qui attribue une probabilité à toutes les séquences de tokens. Une distribution de probabilité est un objet mathématique qui peut être utilisé pour évaluer la probabilité d'un événement, mais aussi pour identifier l'événement le plus probable, en général ou sous certaines conditions. Ainsi, toute distribution de probabilité peut être utilisée pour générer les événements qu'elle décrit.

En général, les modèles de langues sont construits automatiquement sur un volume de données aussi grand que possible. Les modèles par n-grammes sont les plus simples et les plus populaires. Pour chaque token, ces modèles observent les $n-1$ mots qui précèdent l'observation. Dans un contexte de génération de texte, les modèles n-grammes répondent à la question : *Étant donné les* $n-1$ *mots précédents, quel est le mot le plus probable en fonction de ce que j'ai déjà observé ?*

Des modèles plus complexes ne tiennent pas simplement compte des derniers mots, mais de la structure des phrases comme les grammaires hors-contexte probabilistes[3], ou utilisant les réseaux de neurones [4].

Si ces générateurs de texte ont l'avantage d'être rapides à développer sans requérir le coûteux labeur d'informaticiens-linguistes. La qualité des textes qu'ils produisent est insuffisante pour une utilisation sérieuse, mais elle est idéale pour la parodie !

L'utilisation de modèles statistiques dans un contexte parodique est expliquée dans [21] et anticipée dans un des premiers systèmes parodiques facilement disponibles, *Postmodernism Generator*, mentionné dans [4]. Ce système visait à ridiculiser certains écrits dits *postmodernes* au milieu de la *Guerre des sciences* lancée par la supercherie d'Alan Sokal et par la publication d'un ouvrage devenu célèbre [22].

[3] Une description de ces objets peut être trouvée dans [20]

[4] *Long-Short Term Memory*, tels que présentés dans [23]. Andrej Karpathy rend disponible un *modèle de charactères* par réseau de neurones récurrents. Plutôt que d'apprendre une distribution sur les séquences de mots, le modèle apprend une distribution sur les séquences de caractères. L'architecture utilisée a été publiée dans [24].

Ceci ne signifie pas que les modèles de langue ne sont pas utiles. Ils ne peuvent tout simplement pas être utilisés tels quels dans un *GAT* sérieux. Mais ils peuvent y jouer un rôle, dont on discutera en conclusion.

2.1.3 Pourquoi pas des graphiques ?

Les *GAT* ne veulent pas remplacer les graphiques. Les données qui peuvent être comprises en un coup d'oeil par un graphique devraient être représentés graphiquement. Or, toutes les données ne sont pas adaptées à des présentations graphiques, ou encore demanderaient la construction de visualisations sur mesure. De plus, les graphiques doivent souvent être accompagnées de texte pour expliquer l'origine des données, leur signification ou la manière de les agréger.

Les graphiques ont une grande valeur descriptive, mais ils sont peu adaptés aux textes persuasifs que nous construisons. Si la stratégie de persuasion poursuivie met l'accent sur la présentation de données, ce que nous faisons partiellement, il peut être judicieux d'avoir un élément graphique.

Certaines données sont pourtant très difficiles à représenter graphiquement. Pensons simplement à des vecteurs de haute dimension. Les relations logiques que nous désirons exprimer dans la partie descriptive de BPPGen pourraient être décrites par un graphe, mais il est douteux que ce graphe contribue à la persuasion. Des graphiques générés automatiquement peuvent même nuire au moyen 3 (*simuler un intérêt porté envers sa vie professionnelle*), puisqu'il est peu plausible qu'un humain produise des graphiques pour chaque lettre.

2.2 Architecture des générateurs de texte selon Dale et Reiter

Building natural language generation systems [18] est reconnu comme l'ouvrage de référence en GAT.

2.2.1 Sélection du contenu

La première étape consiste à déterminer le contenu qui remplit l'objectif de communication que souhaite remplir le texte. Considérons deux types de documents et leurs générateurs : la lettre de motivation du chasseur de têtes ; et un document d'orientation professionnelle produit par un orienteur. Ces deux générateurs utilisent la même base de donnée, mais n'utiliseront pas les mêmes faits.

Le document d'orientation sera plus intéressé par des faits statistiques sur l'ensemble des emplois, et particulièrement sur les parcours professionnels. La lettre de motivation se concentre sur une offre et un profil, et utilise la base de données pour simuler la connaissance du domaine : le but n'est pas d'informer le lecteur.

Le contenu est finalement une fonction des entrées. Il n'y a pas de règle générale qui guide la création de cette fonction : elle sera propre à chaque système. La sortie de la sélection de contenu est une représentation abstraite de l'information à communiquer.

2.2.2 Structuration du document

La structuration du document comporte trois sous-étapes : le formatage de haut niveau, l'organisation en sections et l'ordonnancement des phrases.

2.2.2.1 Formatage de haut niveau

Le formatage de haut niveau désigne l'ensemble des politiques de formatage de la page avant que le contenu linguistique ne soit produit. Par exemple, c'est à cette étape

que serait décidé si une image est insérée ou non ; ou dans quel élément HTML pourrait s'insérer le contenu du texte à proprement parler.

JSRealB, le système que nous utilisons pour l'étape finale de la génération de texte, possède des fonctionnalités de formatage dans le texte : en effet, il est possible d'ajouter aux informations grammaticales (comme la personne ou le nombre) des informations de formatage HTML. Ce formatage permet d'appliquer des styles différents aux éléments du texte.

2.2.2.2 Division en sections

Les textes portant sur un sujet sont divisés en passages constitués de plusieurs paragraphes : un changement de sous-sujet correspond à un changement de paragraphe : mais tout paragraphe ne correspond pas à un changement de sous-sujet. Dans [7], on réussit à reproduire la compétence humaine à diviser un texte en détectant les changements de sujets, et confirme cette hypothèse. Il est donc raisonnable d'organiser le texte de cette façon.

La division du texte en unités sémantiquement distinctes est similaire à la question de l'ordonnancement des phrases : mais il est plus facile d'appliquer des règles manuelles au niveau de l'organisation des sections du document. Par exemple, les lettres générées doivent toujours commencer par la formule d'appel : c'est une règle codée en dur.

2.2.2.3 Ordonnancement des phrases, transition et relations rhétoriques

Un texte n'est pas un ensemble d'énoncés dont le contenu informatif est entièrement décrit par leur contenu logique : l'ordre des phrases, des mots, les liens entre eux contiennent également une grande quantité d'information. Ces non-dits n'ont pas pour seul but de transmettre de l'information, mais aussi d'assurer au texte une cohérence dont le défaut compliquerait la lecture.

Ces liens, appelés *transitions*, peuvent être établis par des mots, ou simplement implicites. Ces enchaînements ne peuvent souvent être compris qu'à la lumière d'importantes suppositions du sens commun. Par exemple, *Jean lance la balle. Paul l'attrape au bond, pivote, jette un regard noir à André et la lui jette au visage. Jean accourt auprès du but.* Aucun marqueur textuel n'est nécessaire pour indiquer l'ordre chronologique des événements décrits ; et l'auditeur qui ne comprend pas que ces événements se succèdent n'a tout simplement pas compris le texte. L'écriture et la lecture exigent la compréhension des transitions entre les phrases, la faculté de lire entre les lignes. Le contenu sémantique du texte ne peut pas être réduit à une interprétation logique phrase-à-phrase.

PORTET et al. [16] présente une architecture d'écriture de textes narratifs en contexte hospitalier. Dans REITER et al. [19], les auteurs demandent à des linguistes d'évaluer la qualité des textes produits et concluent que leur plus grande faiblesse est la structure narrative et les transitions chronologiques.

Les lettres de recrutement ne contiennent pas de transitions chronologiques, bien que nous ayons envisagé d'inclure une courte biographie professionnelle à la lettre, basée sur les expériences décrites dans le profil *LinkedIn* du candidat.

Les transitions entre les phrases sont importantes pour comprendre et écrire un texte. L'étude de cette cohérence est l'objet de la pragmatique linguistique et de la philosophie du langage[5]. Nous nous contenterons de détailler les moyens linguistiques de réaliser l'illusion de la cohérence.

Une théorie a été développée spécifiquement pour rendre compte de la cohérence textuelle en génération de texte : la théorie de la structure rhétorique (RST, pour *Rheto-*

[5]Il ne faut pas exagérer la distance entre la philosophie du langage et le traitement des langues naturelles. La philosophie dite *analytique*, dominante dans le monde anglo-saxon, doit son existence au projet de traduire la langue naturelle en celle de la logique mathématique, pour faire correspondre notre notion de vérité à la *valeur de vérité* en logique. Cette langue de la logique pourrait être manipulée automatiquement et permettrait, disait-on, de dégager l'esprit humain des erreurs dues aux accidents du langage. L'opinion consensuelle des philosophes est que ce mouvement, appelé le positivisme logique, a perdu le souffle entre les années 50 et les années 60 (voir [5] p.1)

rical Structure Theory, voir [12]). Cette théorie propose une structure du texte artificiel : les énoncés sont soit noyaux, soit secondaires. Les énoncés secondaires n'ont pas de cohérence sans l'énoncé noyau auquel ils se rapportent. Ces relations entre les énoncés forment une arborescence, dont les arcs sont qualifiés par le type de la relation.

Plusieurs algorithmes de génération de texte utilisent implicitement la RST. Notamment, PORTET et al. [16] l'utilise pour la génération de rapports dans les unités néonatales.

L'algorithme qui construit les phrases dans BPPGen n'est pas fondée sur la RST. Celle-ci a été considérée dans l'élaboration de notre algorithme, décrit au chapitre suivant. La notion de *noyau* utilisée dans BPPGen est une simplification de la notion de noyau dans la RST.

2.2.3 Agrégation

L'agrégation a pour but de regrouper des informations similaires avant de produire les phrases. Cette étape permet de créer des phrases plus complexes, et de réduire la redondance.

La réduction de la redondance et de la répétition est une motivation importante de l'utilisation des GAT. L'agrégation peut servir à combiner des phrases semblables, mais pourrait être plus complexe. La notion d'agrégation dans le contexte des GAT n'est pas clairement définie : un système dont le contenu est produit par une base de données peut recevoir comme données des enregistrement déjà agrégés (par exemple en utilisant *GROUP BY* en *SQL*. L'agrégation dans BPPGen ne fait aucune supposition sur l'origine des faits à agréger : c'est simplement une étape qui condense les phrases avant la génération des phrases.

2.2.4 Choix lexical

On doit d'abord distinguer un lexique d'une *lexicalisation* (ou choix lexical). Le lexique est l'ensemble des mots qui peuvent être utilisés. La lexicalisation établit le lien entre des concepts et les syntagmes qui les réalisent. Autrement dit, la lexicalisation établit le lien entre la sémantique et la syntaxe.

Cette étape implique une forme de dictionnaire qui établit les traductions entre les données et les mots. Cette ressource sera parfois appelée la *lexicalisation*, auquel cas on utilisera l'expression *choix lexical* pour désigner l'utilisation de cette ressource.

Une note sur le lien entre l'organisation textuelle et la lexicalisation. Deux idées peuvent être identiques, à cela près qu'ils ne surviennent pas au même endroit du texte. Si ces événements n'ont qu'une seule façon d'être exprimés, la répétition sera nécessaire, ce qui peut sembler artificiel et nuire au moyen *imiter le style auquel est habitué l'humain*. La répétition trop fréquente de certains mots peut nuire à l'objectif de similitude au texte humain. Il est donc préférable de posséder plus d'une lexicalisation par concept, et d'établir une stratégie de choix lexical.

Une des façons de réduire la répétition est d'utiliser les expressions référentielles.

2.2.5 Expressions référentielles

Cette tâche consiste à créer des expressions référentielles pour identifier les objets discutés par le texte. Ces expressions incluent le nom de l'objet, mais aussi les pronoms, les anaphores et les paraphrases.

Cette étape est semblable à la lexicalisation, et y est intimement liée, mais présente des défis uniques qui ont été grandement étudiés dans la littérature [9].

Dale et Reiter [3] proposent une série de critères que doivent satisfaire les expressions référentielles générées par un GAT :

1. le référent ne doit pas être ambigu ;

2. le référent doit être facile à trouver ;

3. l'expression référentielle est générée en un temps raisonnable ;

4. aucune fausse inférence ne doit être suggérée au lecteur.

L'ambiguïté référentielle est au coeur du problème des expressions référentielles. Pour illustrer ce problème, considérez cet exemple

Exemple 2.1

Référence par démonstratif

Des compétences de programmation et de SQL sont exigées. Votre profil n'indique pas que vous avez cette compétence.

De quelle compétence parle-t-on ? SQL, ou la programmation ? Cette ambiguïté n'est pas due à une erreur de lexicalisation : si nous n'avions mentionné qu'une seule de ces compétences à la première phrase, le texte aurait été clair. Le choix d'expressions référentielles ne doit donc pas uniquement tenir compte de l'objet à lexicaliser, mais également des autres objets mentionnés dans le texte.

À cette difficulté s'ajoute celle des fausses inférences suggérées au lecteur. Considérons :

Exemple 2.2

Adressez-vous à l'homme qui a trois enfants.

La phrase peut porter à confusion s'il n'y a qu'un homme dans la salle, et que ses enfants ne sont pas présents. L'auditeur pourrait en déduire qu'elle n'est pas dans la bonne pièce.

BPPGen parvient à éviter ce problème en raison de la nature particulière du problème de la lettre : le sujet de la lettre est invariablement le candidat, et aucune autre entité ne peut prétendre à une place aussi centrale que lui. Il serait impertinent de donner une série de faits sur une compétence particulière, cela nuirait indirectement au moyen persuasif 3 (*simuler un intérêt porté envers sa vie professionnelle*) en changeant de sujets.

2.2.6 Réalisation

La réalisation consiste à produire un texte grammaticalement correct à partir d'une certaine représentation. Les systèmes de réalisation prennent en entrée une représentation du texte permettant l'application de règles grammaticales, et produisent la séquence finale du texte en appliquant celles-ci. On *réalise* une structure grammaticale en texte.

Ces représentations sont soit des grammaires par constituants (GC), soit des grammaires par dépendance. Ces deux types de grammaire sont caractérisés par des structures arborescentes différentes. La librairie utilisée est de type GC.

Les arbres des GC ont des noeuds intermédiaires, qui ne sont pas directement associés à un token ; et les noeuds terminaux, ou feuilles, qui contiennent généralement un token. Ces structures seront familières au lecteur.

Voici un exemple de code JSRealB, sa représentation graphique et sa réalisation :

Exemple 2.3: Code JSRealB

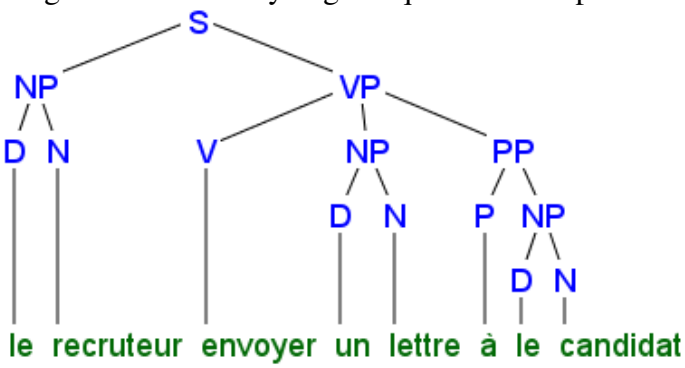
```
S(NP(D('le'), N('recruteur')),
  VP(V('envoyer'),
     NP(D('un'), N('lettre')),
     PP(P('à'), NP(D('le'), N('candidat')))))
```

Exemple 2.4: Réalisation de l'exemple 2.3

Le recruteur envoie une lettre au candidat.

Du point de vue du génie logiciel, les GC ont une propriété intéressante qui les rendent intuitives : elles respectent la règle de couverture. Celle-ci prévoit que si deux feuilles (donc les tokens) A et B ont un ancêtre commun P non partagé par la feuille C, alors il est impossible que nous obtenions $A...C...B$. Autrement dit, les dépendances grammaticales sont imbriquées comme dans des parenthèses, à la manière d'appels de fonction, ce qui donne un aspect déclaratif à l'API.

Figure 2.1 : Arbre syntagmatique de l'exemple 2.3



2.2.6.1 JSRealB

JSRealB transforme une représentation en GC et produit du texte. Il y a les syntagmes terminaux (*N*, *V*, *Adv*, *A*, *P*, *C*, *D*, *Pro*), les syntagmes abstraits de niveau plus élevé (XP, où X est un de *N*, *V*, *Adv*, *A*, *P* ou *C*), auquel on ajoute le syntagme de départ, *S*. En plus, ces syntagmes peuvent être modifiés par des attributs, comme le temps, le nombre, le genre, la personne et des informations de formatage.

JSRealB [14] est une librairie Javascript développée au RALI par Nicolas Daoust et Paul Molins sous la direction de Guy Lapalme. Elle reproduit la syntaxe de la librairie Java SimpleNLG [6]. JSRealB doit charger un lexique en mémoire avant de fonctionner. Le lexique définit, pour chaque mot, son appartenance à une catégorie grammaticale et à un type morphologique. Le type morphologique détermine quelles règles sont utilisées pour transformer le mot de base en une de ses formes. Par exemple, le type morphologique de *chien* indique qu'il fait son pluriel avec le *s*.

JSRealB accomplit deux tâches : modifier le mot en fonction des attributs ; propager les attributs aux segments de la phrase qui en dépendent. Par exemple, il suffit d'indiquer que le sujet est pluriel pour que cet attribut se propage à l'adjectif qui le complète, qui sera lui aussi au pluriel. Cette propagation est assurée par un ensemble de règles.

Maintenant que nous avons contextualisé brièvement les GAT en général, le prochain

chapitre introduit BPPGen spécifiquement. Nous commencerons par donner une vue d'ensemble de son fonctionnement. Nous décrirons ensuite la génération des ressources utilisées par BPPGen.

CHAPITRE 3

BPPGEN - LES ÉTAPES DE LA GÉNÉRATION

Ce chapitre présente les principales étapes de BPPGen et les illustre avec l'exemple 1.1 introduit à la page 3.

BPPGen est un générateur de texte de type données-vers-texte (*data-to-text*) qui prend en entrée les données d'un candidat et d'une offre d'emploi et produit un texte écrit en anglais ou en français. Son architecture est directement inspirée de celle décrite par Dale et Reiter (p. 2.2).

On peut décrire la séquence des opérations de BPPGen en quelques phrases. Les entrées déterminent la structure globale du document. La structure du document et les entrées déterminent quels faits seront exprimés. Les faits sélectionnés sont ensuite placés dans des structures prédéfinies. Les faits et la structure qui les supportent sont transformés en arbres de syntaxe, qui sont finalement réalisés en texte.

Le but de ce chapitre est de montrer comment l'exemple 1.1 a été généré. Nous montrerons donc le résultat de chaque étape.

3.1 Structuration du document

S'inspirant des leçons tirées de [7], le GAT décompose la tâche en passages composés de quelques paragraphes. Un planificateur crée les objets responsables de produire le texte d'un passage. Ces objets, les **générateurs de section**, sont responsables des agrégations et de l'ordonnancement des phrases. Le tableau en 3.I illustre la division en sections pour l'exemple 1.1.

Nous ne décrirons pas le contact et les salutations, car ce sont des segments invariables. La formule d'appel est créée à partir d'un gabarit décrit au chapitre 5.

Tableau 3.I : Exemple de lettre, segmenté en sections

Formule d'appel	Dear Mr. John Doe,
Présentation	I write this email after having read your LinkedIn profile. My name is Joan, I am recruiter with LittleBIGJob. Our company uses an artificial intelligence to make data-driven decisions in human resources. I only contact the best candidates based on fancy statistical techniques, and you've made the short list. The position you are being considered for is Marketing Manager with ACME INC. Restaurant Group. Let me tell you why I would recommend you.
Qualifications	You are an expert in Marketing Management, Management and Marketing, which is exactly what my client is looking for. You seem pretty qualified because you have held the identical occupation of Marketing Manager once and you are proficient in Public Relations, Marketing Strategy, Advertising and Sponsorship. You are a wonderful fit. Your experience as Executive Director suggests that you know a thing or two about Event Management and I suppose that your experience as General Manager has taught you a lot about Sales. I am not a specialist but I figure that your previous work as Communications Consultant has taught you all there is to know about Corporate Communications.
Contact	Would you be open to discuss this over the phone ? When would you be available ?
Salutations	Thank you for your time and your reply,

La section *Présentation* fonctionne de la même manière que la section *Qualifications*, mais est beaucoup plus simple. En effet, deux faits sont invariablement générés directement à partir des entrées brutes. Ce mémoire se concentre donc sur la section *Qualifications*, pour laquelle toutes les étapes du *GAT* sont exécutées.

3.2 Sélection de contenu

La sélection de contenu consiste à identifier les faits qui seront exprimés en fonction des entrées. Les faits sont des énoncés qui sont vrais d'une entité. Par exemple, *You are a specialist of Marketing Strategy* est l'expression d'un fait. De manière abstraite, un fait est un prédicat que vérifient des entités. Nous reprendrons cette notion au chapitre 5

Exemple 3.1: Extrait de la table 3.I, annotée pour les faits

You are an expert in Marketing Management, Management and Marketing[1], which is exactly what my client is looking for. You seem pretty qualified because ***you have held the identical occupation of Marketing Manager once[2]*** *and* ***you are proficient in Public Relations, Marketing Strategy, Advertising and Sponsorship[3]****. You are a wonderful fit.* ***Your experience as Executive Director suggests that you know a thing or two about Event Management[4]*** *and* ***I suppose that your experience as General Manager has taught you a lot about Sales[4]****. I am not a specialist but I figure that* ***your previous work as Communications Consultant has taught you all there is to know about Corporate Communications[5]****.*

L'exemple 3.1 illustre l'apport des faits au texte final : ils forment la substance du propos. Si les phrases correspondant aux faits sont faciles à interpréter pour un humain, les faits eux-mêmes ont un sens informatique beaucoup plus restreint.

Les nombres correspondent à des types de faits, qui seront expliqués plus loin.

3.3 Agrégation

La production du texte passe ensuite à la structure *micro* du document. Les détails de cette structure sont données en détail à la section 5.4. Il suffit ici de dire que des emplacements pour les faits sont prédéfinis dans la structure du document. Ces emplacements sont appelés NOYAUX. BPPGen décide de l'emplacement de chacun des faits dans cette structure. Le noyau joue le rôle d'ossature d'une section.

L'exemple 3.II présente la section *Qualifications* de l'exemple 1.1, séparé par noyaux simples. Les segments de texte propres au noyau sont en gras. Notez que ces segments sont exactement le complément des segments associés aux faits (voir exemple 5.1, p. 53).

3.4 Le choix lexical

Le choix lexical est l'étape consistant à transformer des structures abstraites en arbres de syntaxe. Chaque structure a au moins une manière d'être exprimé. Ces *manières d'être exprimé* sont appelées des *lexicalisations*. Le choix lexical consiste à choisir une lexicalisation pour exprimer un concept.

Par exemple, BPPGen peut écrire *I am not well-versed in these matters but* au lieu de *I am not a specialist but*, dont le sens est similaire.

3.5 La réalisation

Finalement, les arbres syntaxiques pleinement lexicalisés sont expédiés au réalisateur. BPPGen, ayant choisi l'arbre syntaxique associé à *I am not a specialist but*, produira le texte.

Un lexique doit être construit pour que JSRealB puisse réaliser ces structures : plusieurs mots sont inconnus à ce système. Ces mots inconnus sont les lexicalisations des

Tableau 3.II : Extrait *Qualifications* de 1.1 segmenté par noyau

A	You are an expert in Marketing Management, Management and Marketing, **which is exactly what my client is looking for**.
B	**You seem pretty qualified because** you have held the identical occupation of Marketing Manager once and you are proficient in Public Relations, Marketing Strategy, Advertising and Sponsorship.
C	**You are a wonderful fit**. Your experience as Executive Director suggests that you know a thing or two about Event Management and I suppose that your experience as General Manager has taught you a lot about Sales.
D	**I am not a specialist but I figure that** your previous work as Communications Consultant has taught you all there is to know about Corporate Communications.

milliers d'entités générées automatiquement par BPPGen. La prochaine partie explique en détail la génération des ressources utilisées par le GAT.

3.6 Mêmes faits, lettres différentes

BPPGen permet ainsi de produire plusieurs lettres différentes pour exprimer les mêmes faits. L'exemple 3.2 a été généré à partir des mêmes entrées que l'exemple principal. Le contenu informationnel est le même, mais la lettre est différente.

Exemple 3.2: Section *Qualifications* écrite différemment

You are an expert in Marketing Management, which is exactly what my client is looking for. You seem pretty qualified because you are proficient in Public Relations, Marketing Strategy, Advertising and Sponsorship, you are expert in Management and Marketing and you have held the identical job of Marketing Manager previously. You are an exceptional fit. Your professional experience as Executive Director suggests that you know a thing or two about Event Management and I figure that your professional experience as General Manager has taught you all there is to know about Sales. I am not well ver-

sed but your past experience as Communications Consultant tells me that you master Corporate Communications.

Cette section présentait la séquence des opérations de BPPGen. Les prochains chapitres expliquent son fonctionnement. Le chapitre 4 explique la génération et l'utilisation des ressources que nécessite BPPGen. Le chapitre 5 expose les détails d'implantation que le présent chapitre a omis.

CHAPITRE 4

BPPGEN - LES RESSOURCES

4.1 Formats des données et entrées

Les enregistrements de la base de données et les entrées ont le même format. Cette section décrit les champs les plus importants.

4.1.0.1 Offres

Les offres d'emploi ont un champ company_name, qui contient le nom de la compagnie qui a publié l'offre. Ce champ n'est pas toujours fiable : il peut contenir le nom de la firme de recrutement agissant au nom de son client, l'employeur véritable. Le champ title, dans lequel on peut trouver le titre de l'offre emploi proposé sur le site de recherche d'emplois. C'est une chaîne de caractères généralement composée du titre du poste, et d'autres informations (par exemple, la ville, le salaire, le nom de la compagnie).

Le champ description contient le texte de l'offre d'emploi. On y trouve en texte libre les responsabilités associées au poste ; les compétences et la scolarité exigées ; la description de l'entreprise ; les avantages sociaux ; peut-être même la rémunération. Ces informations ne sont pas structurées et sont difficiles à extraire.

L'exemple au tableau 4.I illustre bien que la description contient la plus grande partie de l'information de l'offre, et que cette information est difficile à extraire du texte. Plus de 3000 caractères du champ description ont été omis.

4.1.0.2 Profils

Les profils sont plus structurés que les offres. Plusieurs champs, dont les titres d'emplois présent et antérieurs ainsi que les compétences, sont des textes courts de moins

Tableau 4.I : Exemple d'offre bien renseignée

title	Corporate Marketing Manager
langid	en
place	Vancouver, BC
company_name	ACME INC. Restaurant Group
description	ACME INC. Restaurant Group is dedicated to developing & supporting restaurant Franchise brands that wow the guest & create business ownership opportunities for motivated industry leaders. We bring fun [. . .] Oversees enhancement, sales growth & logistical support of company-wide Gift Card program. Provides support & enrichment of Franchise Marketing including printed/digital communications, Franchise recruitment, and lead generation activity. Manages videography and photography projects for advertising, corporate collateral & digital platforms Desired Qualifications, Skills & Experience : Post-secondary education in Marketing, Communications, or related field. Minimum 5 years' Marketing experience in restaurant and/or Franchise business environment ; understanding of foodservice industry metrics strongly preferred. Proficiency using Microsoft Office (Excel, Word, Outlook) ; SharePoint experience a strong asset. Intermediate Adobe Creative Cloud skills ; knowledge & understanding of various video platforms. Solid verbal and written communication and presentation skills. Proven ability to think strategically by identifying needs, seeking solutions, and outlining areas of improvement with regards to Marketing & brand concepts. Results-oriented and process driven, with high expectations of self in managing change & tight deadlines. Compensation : Competitive full-time salary commensurate with experience, and includes comprehensive health & dental benefits,[. . .] Job Type : Full-time Required experience : Marketing in restaurant/hospitality industry : 5 years

de six mots. *LinkedIn* fournit un service d'autocomplétion pour les compétences : ceci amène une standardisation *de facto* des compétences. Le tableau 4.II présente un exemple d'un profil bien renseigné.

Les profils contiennent également deux champs de marque personnelle (*personal branding*), une courte description de la carrière d'un candidat. Elle peut être courte ou longue. *LinkedIn* permet les deux formes : le personalBranding_pitch, un texte plus long qui met en relief les compétences, aspirations et réalisations les plus importantes ; et la prétention (personalBranding_claim), une description de quelques mots.

Les profils listent également une liste d'expériences professionnelles des candidats, incluant l'emploi au moment de la collecte des données. Un exemple d'une expérience est fournie au tableau 6.IV. Une expérience est constituée du titre de l'emploi occupé (la function), d'une description des responsabilités associées (la mission), ainsi que les dates de fin (endDate) et de début (startDate). Nous utilisons les dates pour établir l'ordre chronologique des occupations.

Les profils ont aussi un champ éducation, qui a été laissé de côté parce que le contenu ne semblait pas immédiatement pertinent pour les candidats qui ont terminé leur formation il y a de longues années. Nous avons laissons l'exploration de champ à des travaux extérieurs.

4.2 Description de la base de données

Les données mises à la disposition du *Butterfly Predictive Project* étaient de deux natures : 267 903 offres d'emplois extraites de deux sites d'emploi[1] ; et 25 345 054 profils LinkedIn du Canada (11 210 042) et de France(14 135 012), de langues anglaise (14 716 563) et française (10 628 491). Les enregistrements, des documents JSON, ont

[1] *indeed.ca* est un site généraliste d'offres d'emploi en anglais et en français. *Cadremploi.fr* est un site plus spécialisé d'offres de postes de cadres en France.

Tableau 4.II : Exemple de profil bien renseigné

countryCode	CA
personalBranding_pitch	A driven CIM (Chartered Institute of Marketing) qualified marketing manager with proven success ; strong management and creative capability, leadership and management experience. A record of effectively managing agencies and working in teams to deliver on set targets and goals. Motivated by challenges, with excellent interpersonal and communications skills and proven ability to operate effectively in a variety of disciplines and in high-pressured environments.
administrativeArea	Nova Scotia
personalBranding_claim	Executive Director at Wayne Entreprises, Inc.
skills	• Project Management • Management • Strategic Planning • Public Speaking • Team Building • Team Leadership • Social Networking • Marketing Strategy • Strategy • Marketing • Public Relations • Event Management • Advertising • ...

Tableau 4.III : Exemple d'expérience passée du profil du tableau 4.II

function	Marketing Manager
startDate	2009-02
endDate	2012-06
missions	Seasoned marketing manager.
companyName	Skynet Robotics Inc.

Tableau 4.IV : Exemple d'expérience courante du profil du tableau 4.II

function	Executive Director
startDate	2013-07
endDate	
missions	Seasoned marketing executive.
companyName	Skynet Robotics Inc.

été insérés dans une base de données *document* pour en faciliter l'entretien, les pré-calculs et les requêtes exploratoires..

Les profils ne sont pas tous suffisamment renseignés pour être utiles. À peine 23% des profils contiennent plus de 500 caractères[2], ce qui est bien peu pour l'application des techniques du TAL. Aucune validation n'est opérée sur les champs par le réseau social : dans le champ compétence, on retrouve des noms de ville, de personnes physiques et morales. De plus, l'identifiant de langue n'est pas toujours exact : on peut trouver du chinois et du français dans les profils dits anglais. Ce problème est particulièrement saillant dans les profils de langue française. En effet, lorsque les compétences constituent une grande partie du contenu textuel de l'enregistrement, on lui attribue souvent incorrectement la langue anglaise en raison de l'autocomplétion anglaise. Cette *contamination* des profils français par l'anglais a un effet visible sur les lexicalisations françaises.

Ceci mène naturellement à cette conclusion, qui est déterminante dans le choix des

[2]Statistique compilée par Rémy Kessler. Le texte considéré est la concaténation des champs skills, personalBranding_pitch, personalBranding_claim et experiences. On peut consulter ces statistiques à `http://www-etud.iro.umontreal.ca/~gottif/lbj/visualization/stats.html`, consulté le 30 novembre 2016.

analyses de données : les profils ne donnent pas un portrait complet de la vie professionnelle du candidat. En particulier, si une information est absente d'un profil, on ne peut pas en conclure que sa négation est vraie. Par exemple, les programmeurs et développeurs écrivent très rarement PROGRAMMING comme une compétence dans leur profil, mais il ne faudrait pas en conclure que les programmeurs et développeurs ne savent pas programmer ! On en conclut que les seules probabilités de voir un titre et une compétence ensemble ne peuvent pas permettre de tirer beaucoup de conclusions. On préférera donc une analyse qui tient compte des cooccurrences les plus fréquentes à l'exclusion des autres.

Nous utilisons les données à trois fins :

- **Ontologique.** Découvrir les entités (OCCUPATIONS et COMPÉTENCES) manipulées par BPPGen.
- **Lexicographique.** Lexicaliser ces entités et produire une ressource linguistique.
- **Sémantique.** Construire, par inspection d'un grand volume de données, des faits qui mettent ces entités en relations qui constitueront les énoncés.

Les premier et deuxième objectifs sont réalisés simultanément. Chacun de ces motifs remplit les *moyens persuasifs* décrits plus hauts (1.4). Leur utilisation simule la connaissance du domaine en exprimant en langue naturelle des associations inférées des données. Le souci primordial est l'exactitude des informations transmises au candidat : nous voudrions éviter de parler de *HTTP/2* à un plaideur de droit commercial. Il était donc raisonnable de sacrifier la précision au profit du rappel. Ceci étant dit, les données ne se prêtaient pas à l'apprentissage machine supervisé : des jeux de données n'étant pas annotés. Le calcul du rappel et de la précision était impossible.

Nous avons fait le choix de baser l'essentiel des associations sur la base de profils plutôt que sur la base d'offres d'emploi. Le plus grand volume de données, la plus

grande homogénéité des compétences due à l'autocomplétion, la structure plus riche et les champs plus courts sont autant de raisons qui motivent ce choix.

4.3 Génération des entités et de leurs lexicalisations

Le but de l'extraction d'information dans BPPGen n'est pas de couvrir l'ensemble des OCCUPATIONS et des COMPÉTENCES, mais bien de déduire suffisamment de faits pour avoir du contenu à écrire.

Après avoir déterminé la langue avec un outil externe [10], on sélectionne les 7000 titres d'occupation les plus fréquents ; pour les compétences, on se limite à 3500. Augmenter le nombre de compétences introduisait trop de compétences qui étaient des reformulations des précédentes, ou encore des compétences déjà vues incorrectement orthographiées. On impose également un seuil de 100 observations minimum, mais ce seuil n'a jamais été utile. La liste des OCCUPATIONS, la liste des COMPÉTENCES, et les relations entre les éléments de ces listes, constituent l'ontologie de BPPGen.

4.3.1 Entités

Le problème des bases de données textuelles en est un de qualité des données : comment réduire le bruit dans les données ? Il faut sélectionner une des différentes manières d'écrire un même mot ou groupe de mots (majuscules, pluriels, préposition omise, etc.).

Par définition, la normalisation entraîne une perte d'information : plusieurs chaînes de caractères sont ramenées à une seule. Cette perte d'information est souvent souhaitée pour identifier le lexème représenté par le morphème en éliminant les marques de flexion. L'information morphologique était superflue pour BPPGen : nous n'avions pas l'intention d'utiliser les formes féminines ou plurielles des entités.

BPPGen opère une normalisation agressive par racinisation, en utilisant une configuration de l'algorithme de racinisation appelé *Snowball* [15]. À proprement parler, *Snow-*

ball est un langage dédié à la construction de racinisateurs non-lexicalisés. Nous nous sommes contentés d'utiliser les configurations de *Snowball* pour l'anglais et le français qui étaient présentes dans le projet NLTK [2].

La dénormalisation est l'opération inverse de la normalisation : étant donné une chaîne normalisée, on veut connaître la distribution des entrées non-normalisées dont elle est le résultat. À la normalisation, on remplit une table de dénormalisation qui garde la trace de la distribution des entrées. Utiliser un dénombrement ralentissait le processus en raison de la grande quantité de clés. Il a donc fallu accorder une taille maximale à chaque entrée dans la table de dénormalisation. Lorsque nous rencontrons une chaîne dénormalisée et que la table de normalisation a plus de dix entrées, on suppose que l'entrée correspondante à une bonne dénormalisation a déjà été vue.

Exemple 4.1: Exemple de normalisation et distribution de dénormalisations
aim :aime(1), aimes(2), aimer(10), aimant(3), aim(4).

L'imposition de ce seuil est justifiée. Les entités extraites sont des groupes nominaux courts ou des verbes non-conjugués : ces syntagmes sont moins variables morphologiquement que les groupes verbaux, particulièrement en français.

Pour produire les lexicalisations entre une entité, identifiée par sa racine, et son entrée lexicale, la chaîne la plus fréquente est sélectionnée. Une liste noire filtre les erreurs les plus fréquentes. Par exemple, il a fallu exclure *Boston, New York, Paris, Marseille, Montreal, Toronto, Vancouver, Hamilton* et *Calgary*. Les noms de ville étaient souvent présents dans les compétences des profils.

Finalement, les chaîne normalisée forme l'identifiant de l'entité ; la dénormalisation préférée devient sa lexicalisation ; et un fichier de lexicalisation des entités est produit.

Un type d'erreurs imprévu est survenu : beaucoup de profils contiennent en fait des libellés d'occupations dans skills, et ces occupations ont la même racinisation qu'une compétence correspondante (ACCOUNTANT, *accounting*). La dénormalisation donnait

donc le nom de l'occupation plutôt que celle de la compétence. Un filtre subséquent compare les deux dénormalisations, détecte les suffixes associés aux emplois et corrige la lexicalisation produite.

Prenons par exemple la racine *account*. La compétence ACCOUNT était dénormalisée par $[Accountant, Accounting, Accounts, ACCOUNTING]$ et l'occupation ACCOUNT par $[Accountant, Accounting]$. Le système détecte qu'il y a un conflit dans la normalisation entre les compétences et les occupations. Si la dénormalisation de la compétence contient une expression dont le suffixe est associé à une fonction et un autre candidat pour le remplacer, alors on choisira l'expression avec le bon suffixe pour l'occupation, et la compétence aura la dénormalisation la plus fréquente. Pour l'anglais, les suffixes de fonction choisis étaient : *er, ant, tress, ian, ist, or, yst*.

4.3.2 Réutilisation d'une ontologie préexistante ?

Il aurait été possible d'utiliser l'ontologie créée par d'autres membres du *Butterfly Predictive Project* [8] sans modification majeure. Plusieurs raisons motivent la création d'une ontologie indépendante.

Les compétences dans [8] ont été extraites automatiquement dans les description des offres d'emploi. Rappelons que ce champ contient le texte des offres d'emploi. C'est une information non-structurée. Le résultat final est un bon rappel, mais une précision plus faible. Suite à une discussion avec l'auteur, il a été jugé plus convenable pour nos fins d'utiliser les profils. Nous voulons éviter de mentionner des compétences qui n'en sont pas. Les priorités des deux systèmes étaient irréconciliables.

Par ailleurs, nous traitons 7000 OCCUPATIONS par langue, alors que l'autre ontologie se limite à moins de trois cent. La couverture des occupations est moins grande.

4.4 Génération des faits

La troisième utilisation des données suppose qu'est disponible une liste des compétences et des occupations autorisées. C'est-à-dire, elle suppose que les étapes décrites à la section précédente ont été accomplies.

Dans un premier temps, on filtre les profils pour ne retenir que ceux qui contiennent au moins deux expériences dont le titre est dans la liste, et deux compétences de la même manière. Ce filtrage nous permet de nous concentrer non seulement sur les profils qui sont bien renseignés, mais également sur ceux qui sont suffisamment bien décrits par notre ontologie. Cette restriction nous permet d'utiliser le même échantillon pour extraire les différentes statistiques pertinentes, ce qui simplifie l'analyse.

Ce filtre réduit considérablement la taille de l'ensemble considéré : on passe à un échantillon de 1 501 873 profils, soit moins de 6% de la base d'origine. Exiger plus de deux expériences ou de deux compétences aurait réduit la taille de notre échantillon et la fiabilité des associations extraites de celui-ci.

La dernière étape du prétraitement est l'extraction de statistiques pertinentes. Logiquement, cette étape pourrait être faite sur demande lors de la sélection de contenu, mais la taille des bases de données est trop importante pour que cette option soit retenue. Les statistiques sont donc précalculées.

4.4.1 Règles d'associations

Les statistiques qui nous intéressent sont les cooccurrences sur les titres contenus dans les expériences et les compétences. Par exemple, nous sommes intéressés par le taux de profils qui inscrivent l'occupation DEVELOPER dans leurs expériences, ceux qui inscrivent SOFTWARE DEVELOPMENT dans leurs compétences ; et ceux qui écrivent les deux. De ces trois décomptes, on peut construire des associations simples. Ceci donne lieu à quatre situations :

1. on ne peut rien conclure ;

2. de la première, on peut conclure à la deuxième (si le profil contient SOFTWARE DEVELOPMENT, alors le profil devrait contenir DEVELOPER) ;

3. de la deuxième, on peut conclure à la première (si le profil contient DEVELOPER, alors le profil devrait contenir SOFTWARE DEVELOPMENT) ;

4. 2 et 3 sont vrais ;

Nous nous donc basons sur la distribution échantillonnale - donc les décomptes - pour produire ces probabilités. Le point de vue de la théorie élémentaire des probabilités permet effectivement de tirer profit de ces taux et de créer des règles d'association. Il suffit de créer un espace de probabilités sur l'ensemble des profils LinkedIn. Par exemple, si A est l'événement *le profil contient la compétence* LEADERSHIP et B est l'événement *le profil contient l'occupation* CEO, on interprète *P(A)* comme étant la proportion des profils qui contiennent LEADERSHIP, $P(A \cap B)$ comme la proportion qui contiennent l'occupation CEO et la compétence LEADERSHIP. $P(A|B)$ est alors la proportion des profils ayant la compétence LEADERSHIP, en ne considérant que les profils qui contiennent l'occupation CEO : ceci revient à dire que

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \tag{4.1}$$

Autrement dit, on divise le nombre de fois que les entités ont été observées dans le même profil par le nombre de fois qu'une des deux entités a été observée. Les meilleures probabilités conditionnelles identifiées sont candidates pour devenir des règles d'association. Les règles d'association permettent de produire des connaissances dans le domaine du recrutement. Mais que voulons-nous dire par *les meilleures probabilités conditionnelles* ?

Si $P(A|B)$ est *assez près de 1*, nous supposons que tous les profils de CEO ont la compétence LEADERSHIP. Nous dirons alors que la compétence LEADERSHIP est *impliquée* par l'occupation CEO. Après avoir discuté des problèmes que posent les données de BPPGEN dans un contexte statistique, nous expliquerons ce que nous voulons dire par *assez près de 1*.

4.4.2 Problèmes des données pour établir des associations

L'application des probabilités conditionnelles à partir des simples décomptes n'est pas sans difficulté. Dans notre cas, nous relevons trois difficultés principales concernant le lien entre le modèle statistique et la réalité étudiée :

1. le problème de l'antécédent rare ;
2. le flou intrinsèque aux données de recrutement ;
3. le problème du sous-renseignement.

Le premier problème est commun à tous les modèles statistiques qui conditionnent sur des événements rares. Le modèle court le risque de décrire comme impossible une situation qui n'a simplement pas été observée, ou comme trop rare une situation qui a été observée peu de fois. Pour pallier ce problème, les modèles ont souvent recours au lissage, qui consiste à distribuer sur tous les événements non-observés une certaine masse de probabilité.

Nous évitons le problème en imposant des seuils. Nous risquons bien sûr de perdre des associations intéressantes. Or, notre but est d'écrire des lettres, pas de capturer l'essentiel du monde du recrutement. Il est préférable d'être muet que de prouver l'ignorance du système.

Deuxièmement, les notions mêmes de COMPÉTENCE et d'OCCUPATION sont mal définies. Bien sûr, certaines occupations sont très bien définies. Certaines sont même

définies dans la loi, comme les professions. La définition d'AVOCAT n'est pas ambiguë dans notre société. D'autres occupations ne sont pas bien définies. Qu'est-ce qu'un ANALYSTE ? Il y a des analystes financiers, des analystes d'intelligence d'affaires, des analystes en technologie, etc. L'expression ASSOCIATE désigne aussi bien les prestigieux partenaires d'une société en nom collectif, que les commis du *Walmart*.

Il en va de même pour les compétences. Que signifie la compétence OIL ? En soi, rien. La compétence peut donc être fonction des expériences avec lesquelles la compétence est observée. La compétence OIL aura un sens bien différent selon qu'elle se trouve dans un profil d'INGÉNIEUR GÉOLOGIQUE, en génie, ou de COMMODITY TRADER, en finance.

Le troisième problème est plus important. Les profils de réseaux sociaux sont un signal peu fiable des compétences réelles des individus. Comme nous l'avons vu, les profils sont assez peu renseignés. En sélectionnant les seuls qui avaient deux compétences et deux occupations, nous coupions 94% de l'échantillon. Ces profils ne sont pas pour autant très complets en regard de l'interprétation désirée. Il est évident que les profils LinkedIn ne brossent pas un portrait fidèle de l'activité professionnelle d'un individu. Comme nous l'avons dit plus haut, le problème est criant en ce qui concerne les compétences sous-entendues par une occupation. Les omissions sont fréquentes parce que les individus veulent mettre l'accent sur un aspect de leur carrière en taisant les autres, et parce qu'ils ne prennent pas le temps de tout écrire. Les compétences qui se trouvent dans les profils peuvent aussi avoir été assignées par d'autres membres de LinkedIn par la fonctionnalité *Endorsements*.

Ce dernier problème est déterminant : il est irréaliste de s'en tenir à des associations vérifiées par plus de 70% des profils.

C'est pourquoi nous avons établi le protocole suivant. Pour chacun des types d'associations énumérés plus bas, un certain nombre d'*associations-sentinelles* sont soigneusement choisies : nous savons qu'elles devraient donner lieu à une règle d'association

valide. Nous regroupons toutes les associations potentielles par 20-quantiles. Après avoir compilé les statistiques, nous évaluons les probabilités de chacune de ces *associations-sentinelles*. La borne inférieure du plus petit vingt-quantile contenant un nombre suffisant de nos *associations-sentinelles* est retenu comme la limite.

Par exemple, pour établir les associations OCCUPATION $\rightarrow$ COMPÉTENCE, nous prenons les *associations-sentinelles* : ATTORNEY et LEGAL WRITING ; PROGRAMMER et SOFTWARE DEVELOPMENT ; MANAGER et MANAGEMENT. Il est clair que les avocats doivent écrire des textes légaux : c'est une partie essentielle de l'emploi. Le développement logiciel est aussi une composante essentielle du travail de programmeur. Et finalement, il est évident que les gestionnaires prétendent s'y connaître en gestion. Si notre système exclut ces associations, il est trop restrictif. Nous devons donc trouver le seuil de support minimal pour les accepter.

Pour OCCUPATION $\rightarrow$ COMPÉTENCE, le seuil identifié était à 20%. Nous en concluons que si une compétence X est observée dans 20% des profils d'une occupation Y, alors tous les profils contenant une expérience de Y devraient contenir la compétence X s'ils étaient complets.

Il peut paraître insensé de généraliser à 100% des cas à partir de 20%. Cette généralisation n'a de sens que si les profils observés sont un proxy pour la réalité que nous souhaitons découvrir avec ces statistiques.

4.4.3 Types de règles d'association retenus

Nous retenons les types d'associations suivants. Nous nommons les types d'implications qui peuvent être établies par $x \rightarrow y$.

1. les cooccurrences des occupations entre elles (associations de type OCCUPATION $\rightarrow$OCCUPATION) ;

2. les cooccurrences entre les expériences et les compétences (associations de types OCCUPATION →COMPÉTENCE et COMPÉTENCE →OCCUPATION) ;

3. les cooccurrences des compétences entre elles (association de type COMPÉTENCE →COMPÉTENCE) ;

4. les cooccurrences entre les occupations, en tenant compte de l'ordre chronologique (associations de types AVANT→APRÈS et APRÈS→AVANT).

Pour les cooccurrences entre compétences et expériences, le lien entre les compétences et les occupations ne pouvait pas être extrait directement. Le format des données n'établit pas de lien direct entre les expériences et les compétences : le profil a une liste de compétences et une liste d'expériences. Une compétence peut être associée à un emploi antérieur, et n'avoir rien à voir avec l'occupation courante. Par exemple, le jeune programmeur qui énumère les compétences pertinentes à son ancien emploi de sauveteur. Notre espoir était que le volume important des données permettrait d'ignorer ces artefacts.

4.5 Production des faits à partir des associations

Le but de ces règles d'association est de produire des faits. Voici en détail comment sont utilisées les associations pour produire les faits de BPPGen. Les faits sont présentés ici dans l'ordre descendant de leur priorité. À la sélection de contenu, on s'assure que les entités ne sont présentes que dans un seul fait. Si plusieurs faits ont une même entité comme argument, on choisit le fait qui a la plus haute priorité.

1. PROFILE_SKILL_EXPERTISE

   Ce sont les COMPÉTENCES dont le libellé apparaît dans la description de l'offre et le TEXTE du profil, ainsi que dans les skills du profil. Autrement dit, ce sont les

compétences qui sont assez importantes pour que le candidat ait jugé bon d'écrire sur elles dans son profil. C'est un fait direct.

2. PROFILE_SKILL_PROFICIENCY

   Ce sont les COMPÉTENCES dont le libellé apparaît dans la description de l'offre, ainsi que dans les skills du profil, à l'exclusion de ceux qui apparaissent dans PROFILE_SKILL_EXPERTISE. C'est un fait direct.

3. INFERRED_REQUIRED_PROFILE_SKILL Ce sont les compétences qui apparaissent dans les skills du profil, et qui sont impliquées par l'OCCUPATION identifiée dans le champ Titre de l'offre. C'est un fait spéculatif qui utilise les implications de type OCCUPATION →COMPÉTENCE.

4. SAME_CURRENT_OCCUPATION Cette relation indique que l'OCCUPATION identifiée dans le champ Titre de l'offre se trouve aussi dans la plus récente expérience du profil, si elle n'a pas de date de fin. C'est un fait direct.

5. SAME_PREVIOUS_OCCUPATION Cette relation indique que l'OCCUPATION identifiée dans le champ Titre de l'offre se trouve aussi dans une des expériences du profil. C'est un fait direct.

6. SIMILAR_PREVIOUS_OCCUPATION La notion d'OCCUPATIONS SEMBLABLES est définie ainsi : ce sont deux occupations qui apparaissent souvent dans les mêmes profils, et dont aucune ne suit l'autre significativement (10%) souvent. Autrement dit, ce sont des libellés alternatifs, ou des occupations très liées. Par exemple, PROGRAMMING ANALYST et DEVELOPER sont similaires. Cette relation produit un fait si une des expériences du profil est similaire à l'OCCUPATION identifiée dans le champ Titre de l'offre. La notion de similarité utilise donc les implications de type AVANT →APRÈS et APRÈS →AVANT.

7. JUNIOR_PREVIOUS_OCCUPATION On déduit de l'ordre chronologique des OCCUPATIONS que l'OCCUPATION identifiée dans le champ Titre de l'offre est généralement précédée d'une des expériences du profil. Autrement dit, cette relation indique que le candidat a un parcours professionnel compatible avec l'offre proposée. Par exemple, si le candidat a une expérience de ACCOUNTANT et que l'offre propose un poste SENIOR ACCOUNTANT, cette relation sera activée. Cette notion utilise les implications de type APRÈS →AVANT.

8. EXPLICIT_OFFER_SKILL_COVERED_ONLY_BY_EXPERIENCE Une des COMPÉTENCES impliquées par une des expériences du candidat se trouve dans l'offre. Ce fait spéculatif utilise OCCUPATION →COMPÉTENCE.

9. IMPLICIT_OFFER_SKILL_COVERED_ONLY_BY_EXPERIENCE Une des COMPÉTENCES impliquées par une des expériences du candidat est impliquée par l'OCCUPATION identifiée dans le champ Titre de l'offre. Ce fait spéculatif utilise OCCUPATION →COMPÉTENCE : une fois pour l'offre, une seconde fois pour l'expérience.

D'autres types de faits ont été envisagés et auraient exigé l'utilisation d'autres champs des offres et profils. En particulier, nous voulions :

- Une fait sur la proximité géographique de l'offre et du candidat ;
- Un fait indiquant que l'éducation du candidat est suffisante, ou insuffisante, pour les exigences de l'offre ;
- Un fait indiquant que la position est une avancée ou un recul significatif dans la carrière du candidat.
- Un fait indiquant que la compagnie offrante recrute souvent des employés d'une des compagnies contenues dans les expériences du candidat.

- Un fait indiquant la taille de la compagnie de l'offre (petite, moyenne ou grande entreprise).
- Un fait indiquant le secteur d'activité de la compagnie de l'offre.

4.6 Discussion des associations

Nous avons décrit les types d'associations, et comment ces associations sont utilisées lors de la génération des lettres pour la création des faits. Dans ce cette section, nous quantifions ces règles d'association.

4.6.1 OCCUPATION → COMPÉTENCE

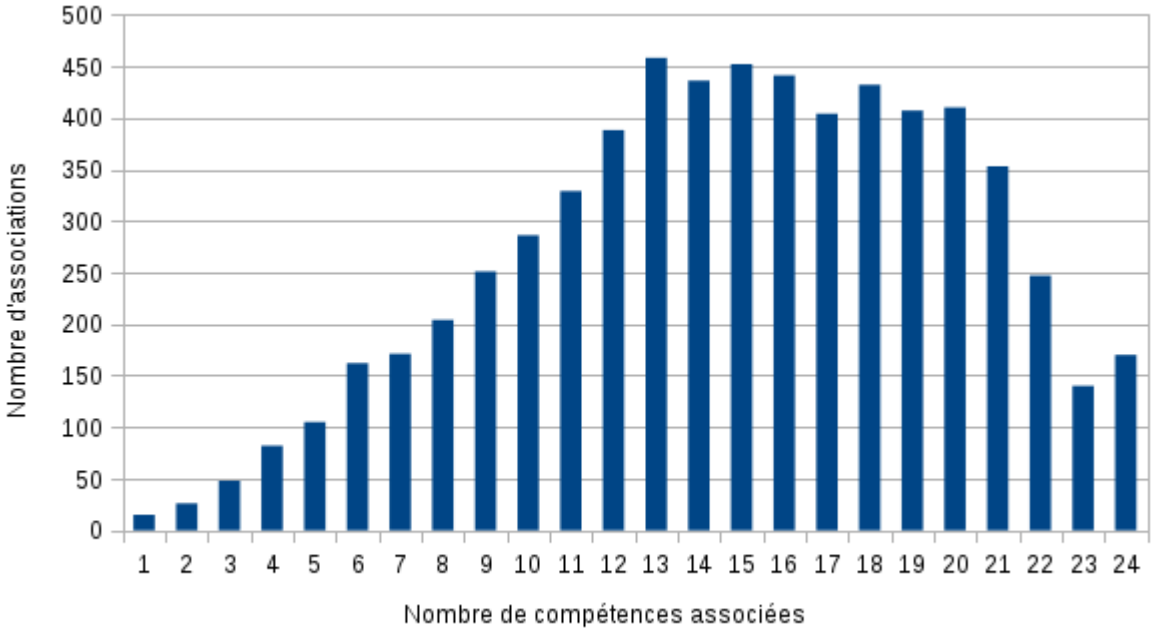
Les associations les plus nombreuses et les plus utiles étaient les associations de type OCCUPATION → COMPÉTENCE. 6467 des 7000 occupations avaient au moins une compétence associée. Au total, 93930 associations étaient produites. 1370 associations étaient associées à vingt compétences ou plus.

La distribution des occupations peut être trouvée à la figure 4.6.1

Par exemple, les compétences associées à BAKERY MANAGER sont : RETAIL, INVENTORY MANAGEMENT, MERCHANDISE, MANAGEMENT, SALES, FOOD, INVENTORY CONTROL, TIME MANAGEMENT, MICROSOFT OFFICE, BAKERY, TEAM BUILDING, STORE MANAGEMENT, LEADERSHIP, VISUAL MERCHANDISE, MICROSOFT WORD, MICROSOFT EXCEL.

Remarquons d'abord la présence de compétences universellement exigées : LEADERSHIP, MICROSOFT EXCEL,MICROSOFT WORD, MICROSOFT OFFICE. Ces compétences sont fréquentes au point d'être peu informatives. BPPGen en exclut certaines lors de la sélection de contenu par une liste noire, mais toutes demeurent visibles dans les données. TIME MANAGEMENT et TEAM BUILDING, par exemple, ne sont pas exclues

Figure 4.1 : Nombre d'occupations regroupées par nombre d'associations OCCUPATION → COMPÉTENCE



bien qu'elles soient très fréquentes.

4.6.2 COMPÉTENCE → COMPÉTENCE

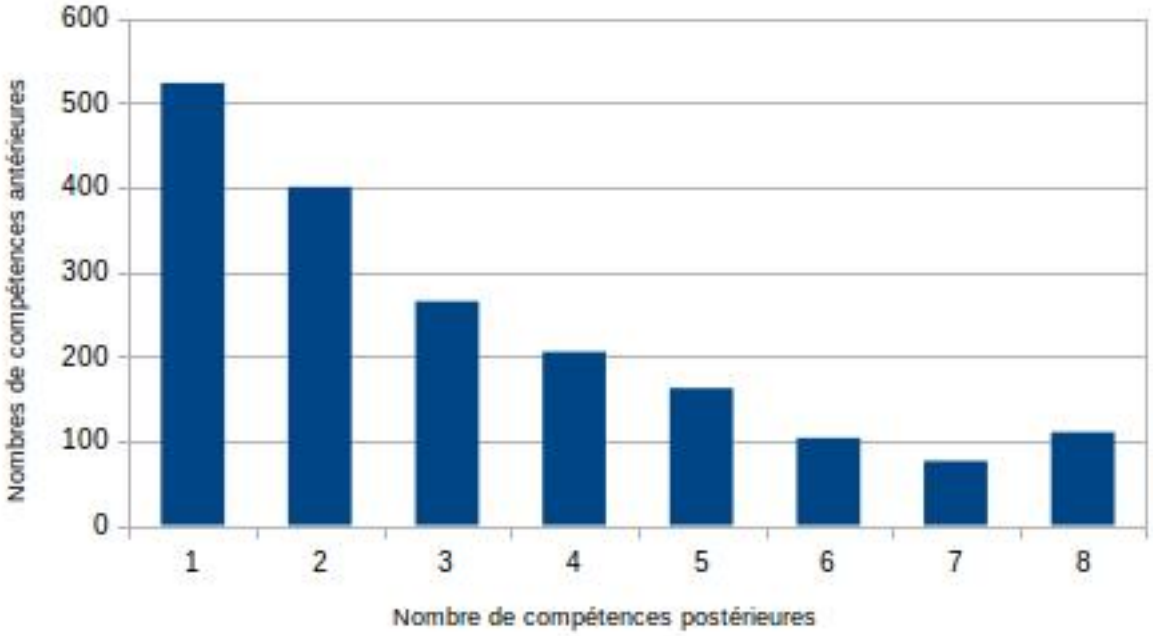
Les règles d'association sont des règles de déduction du type : si *antérieur*, alors *postérieur*. On distingue ici la compétence antérieure des compétences postérieures.

Les associations COMPÉTENCE → COMPÉTENCE sont au nombre de 5963, couvrant 1845 COMPÉTENCES, soit plus de la moitié. 110 compétences ont plus de huit compétences postérieures. Leur distribution est décrite à la figure 4.6.2.

Par exemple, les compétences postérieures à la compétence FLIGHT PLANNING sont :

FLIGHT, COMMERCIAL AVIATION, AIRCRAFT, FLIGHT SAFETY, AIRLINES, CIVIL AVIATION, AIRPORTS, PILOTING, CHARTER.

Figure 4.2 : Nombre d'occupations regroupées par nombre d'associations COMPÉTENCE → COMPÉTENCE



4.6.3 Les autres types d'associations

Pour SENIOR → JUNIOR, nous obtenons 511 occupations antécédentes pour un total de 642 associations. 312 des occupations antécédentes ont soit *Senior* ou *Sr.* dans leur libellé. Pour JUNIOR → SENIOR, 376 antécédents et 497 associations sont extraits. La même observation vaut : on découvre des associations triviales entre l'occupation X et l'occupation JUNIOR X. Par exemple, pour JUNIOR ANIMATOR, les occupations plus senior sont :

1. SENIOR ANIMATOR
2. CHARACTER ANIMATOR
3. LEAD ANIMATOR

Cet exemple est très renseigné. Une de ses associations est triviale : il suffit de chan-

ger *junior* pour *senior* ! Les deux autres nous renseignent véritablement sur l'occupation JUNIOR ANIMATOR.

Au chapitre 4, nous avions défini les OCCUPATIONS SIMILAIRES comme étant celles qui sont fortement associées à une autre, sans être clairement SENIOR $\rightarrow$ JUNIOR ou JUNIOR $\rightarrow$ SENIOR. Les résultats pour cette association sont décevants : 16 occupations seulement étaient couvertes.

Ce chapitre présentait les ressources utilisées par BPPGen pour générer les lettres de recrutement, en particulier comment les entités sont lexicalisées et comment les règles d'association sont produites. Le chapitre 5 présente les détails omis au chapitre 3.

CHAPITRE 5

FONCTIONNEMENT DE BPPGEN

Dans ce chapitre, nous terminons l'exposé sur BPPGen. Si le chapitre 3 expliquait le fonctionnement global du système, celui-ci entre dans les détails d'implantation. Après avoir défini plus formellement les objets manipulés par BPPGen, nous reprendrons les étapes du GAT décrites à la section 2.2 (*Architecture des générateurs de texte selon Dale et Reiter*, p. 12).

5.1 Quelques notions

Les GAT *data-to-text* transforment des données en texte. Les données acceptées par BPPGen sont des **concepts**. Nous définissons les concepts comme *les choses à lexicaliser*. Il y a trois types de concepts, notés AVEC CETTE POLICE pour les distinguer le sens technique donné ici du sens commun.

Les **entités** sont les concepts élémentaires, ou atomiques. Ce sont les objets, ici des compétences et des occupations. Pour BPPGen, il y a deux types d'entités : les compétences et les occupations. Par exemple, la compétence MARKETING STRATEGY et l'occupation MARKETING EXECUTIVE. L'acquisition des entités a été décrite au chapitre 4.

Les **faits** sont équivalents à des phrases simples. Un fait est une instance d'une **relation**. Autrement dit, les faits ont un type, et ce type est nommé *relation*. Les faits ont un certain nombre d'arguments positionnels, comme les fonctions à plusieurs variables. Seule une entité peut être argument d'un fait. De plus, les faits sont **directs** ou **spéculatifs** : les faits spéculatifs sont ceux qui sont inférés des données suite au processus décrit au chapitre 4 (Mêmes faits, lettres différentes, p.27) ; les faits directs sont extraits des

entrées. PROFILE_SKILL_EXPERTISE(MARKETING) est un exemple de fait : la relation est PROFILE_SKILL_EXPERTISE, et son unique argument est l'entité MARKETING.

Une notion voisine est celle d'**agrégation** : il s'agit d'une structure intermédiaire dont les arguments, au lieu d'être des entités simples, sont un ensemble d'entité. Ces agrégations de faits permettent à une phrase d'exprimer plus d'un fait de même relation. Par exemple, une agrégation de faits PROFILE_SKILL_EXPERTISE({MARKETING, EXCEL}) a un argument ({MARKETING, EXCEL}), sa relation est PROFILE_SKILL_EXPERTISE. C'est une agrégation des faits PROFILE_SKILL_EXPERTISE(EXCEL) et PROFILE_SKILL_EXPERTISE(MARKETING). Lorsque plusieurs entités sont dans la même position de l'agrégation de faits, elles sont coordonnées avec *et*.

Les **noyaux** servent à ancrer les faits. Ils composent la trame d'une section. Les faits viennent s'insérer dans un noyau. Les éléments d'un noyau sont des faits. Comme les faits, les noyaux sont composés d'arguments positionnels. Nous appellerons ces arguments des **positions de noyau**. Chaque position de noyau peut contenir plusieurs faits, de la même manière que les agrégations. Lorsque plusieurs faits sont à la même position de noyau, ils sont agrégés ou coordonnés avec *et*.

You are a specialist of Marketing Strategy serait noté ainsi en BPPGen : PROFILE_SKILL_PROFICIENCY (MARKETING STRATEGY). Ceci signifie que la relation PROFILE_ SKILL_PROFICIENCY, qui signifie *Vous êtes un spécialiste de*, est vérifiée de l'entité MARKETING STRATEGY. Nous traiterons des entités et des faits aux sections suivantes.

5.2 Structuration du document : La formule d'appel

Le chapitre 3 expliquait brièvement les générateurs de section. Cette section donne les détails d'un générateur de section qui n'utilise pas la sélection de faits.

Les sections moins complexes, typiques des systèmes de patrons (voir p. 8), devraient

néanmoins être faciles à exprimer par un *GAT* complet [18]. La formule d'appel est un bon exemple de système dont la logique est complexe mais inadaptée à la génération de texte telle que présentée au chapitre 3.

Les arbres de syntaxe produits et acceptés par JSRealB ont un type de syntagme *texte constant*, ce qui signifie qu'un générateur peut se contenter de renvoyer une chaîne de caractères, l'envelopper dans un arbre de syntaxe trivial et l'acheminer au réalisateur telle quelle.

La formule d'appel répond à un patron $(W|XY?Z)$,[1], où X est le titre, Y est le prénom et Z est le nom, et W l'expression à utiliser en cas d'anonymat. Si BPPGen détecte que le candidat est médecin, le titre X sera *Docteur* ; avocat, *Maître* ; etc. Ce sera *Monsieur* ou *Madame* si aucune profession spéciale n'est détectée. Si le niveau de formalité (un paramètre passé avec le profil du candidat) est assez bas, on peut aussi ajouter le prénom.

Le premier cas de l'alternative du patron proposé est le cas où nous n'avons pas de nom : la formule d'appel est alors simplement *Madame*, *Monsieur* ou *À qui de droit*. Peu d'attention a été consacrée à ce problème : les profils dont nous disposons sont anonymisés et le sexe ne peut pas être déterminé en général.

Les titres français des expériences professionnelles ne sont pas fiables : la féminisation des professions n'est pas commune aux deux côtes de l'Atlantique. Le nom, le prénom et le sexe seraient accessibles dans un système de production.

5.3 Sélection de contenu

Cet exemple soulignait le rôle des faits dans la lettre. Les faits contiennent l'essentiel de l'information de la lettre. Les faits sont composés de deux éléments : les entités dont on parle et ce qu'on en dit. Ce deuxième élément est le type du fait. On dira qu'un fait

[1] Le lecteur reconnaîtra ici la syntaxe des expressions régulières. Les parenthèses indiquent le groupe ; la barre, l'alternative ; le point d'interrogation, l'option ; la concaténation, la concaténation.

est une instance d'une relation. Les paragraphes qui suivent décrivent les faits contenus à l'exemple 3.1, à la page 23.

PROFILE_SKILL_EXPERTISE signifie que le profil contient des compétences également contenues dans l'offre. À l'exemple 3.1, ces faits sont annotés par [1]. Ces compétences sont distinctes de PROFILE_SKILL_PROFICIENCY en ce que le candidat a pris le soin d'inscrire ces compétences dans le texte libre du profil (champs claim, pitch et missions). Pour l'exemple 1.1, ces compétences sont GESTION DE MARKETING, GESTION, MARKETING.

SAME_PREVIOUS_OCCUPATION signifie que le candidat a déjà occupé une occupation de même libellé par le passé. Notre candidat a déjà été MARKETING MANAGER. Ce fait est noté [2].

Les items annotés par [3] dans 3.1 sont tous du type PROFILE_SKILL_PROFICIENCY. Cette relation signifie que le profil contient des compétences également contenues dans l'offre. Ici, en traduisant, ces compétences sont STRATÉGIE MARKETING, RELATIONS PUBLIQUES, PUBLICITÉ, COMMANDITES. Dans la plupart des lettres, ce type de faits compte plus d'exemplaires que les autres.

EXPLICIT_OFFER_SKILL_COVERED_ONLY_BY_EXPERIENCE (annoté [4]) signifie qu'une des compétences explicitement exigées dans l'offre ne peut pas être trouvée dans le profil, mais que nos connaissances sur le domaine du recrutement permettent de conclure que cette compétence a été acquise lors d'une expérience antérieure. Dans notre exemple, le profil ne contient pas la compétence VENTE ; nous savons statistiquement que tous ceux qui ont une expérience de GENERAL MANAGER ont une compétence de ventes ; la compétence VENTE est exigée dans l'offre ; le profil contient une expérience de GENERAL MANAGER. *Mutatis mutandis* pour COMMUNICATION CONSULTANT et CORPORATE COMMUNICATION.

IMPLICIT_OFFER_SKILL_COVERED_ONLY_BY_EXPERIENCE (noté [5]) est l'en-

semble des compétences à la fois implicites dans l'offre et dans les expériences du profil. Autrement dit, parmi toutes les compétences normalement associées au titre de l'offre, et non écrites dans l'offre, on sélectionne les compétences qu'on suppose acquises par une expérience antérieure. Dans notre exemple, le profil ne contient pas la compétence GESTION D'ÉVÉNEMENTS ; nous savons statistiquement que tous ceux qui ont une expérience de DIRECTEUR EXÉCUTIF ont une compétence de GESTION D'ÉVÉNEMENTS ; la compétence GESTION D'ÉVÉNEMENTS n'est pas exigée dans l'offre, mais se trouve assez souvent dans les profils de MARKETING MANAGER (le titre l'offre) ; le profil contient une expérience de DIRECTEUR EXÉCUTIF.

Exemple 5.1: Les faits sélectionnés pour l'exemple 3.1

PROFILE_SKILL_PROFICIENCY(MARKETING STRATEGY),

PROFILE_SKILL_PROFICIENCY(PUBLIC RELATIONS),

PROFILE_SKILL_PROFICIENCY(ADVERTISEMENT),

PROFILE_SKILL_PROFICIENCY(SPONSORSHIP),

PROFILE_SKILL_EXPERTISE(MARKETING MANAGEMENT),

PROFILE_SKILL_EXPERTISE(MANAGEMENT),

PROFILE_SKILL_EXPERTISE(MARKETING),

SAME_PREVIOUS_OCCUPATION(MARKET MANAGER),

EXPLICIT_OFFER_SKILL_COVERED_ONLY_BY_EXPERIENCE

(SALE, GENERAL MANAGER),

EXPLICIT_OFFER_SKILL_COVERED_ONLY_BY_EXPERIENCE(CORPORATE COMMUNICATIONS, COMMUNICATION CONSULTANT),

IMPLICIT_OFFER_SKILL_COVERED_ONLY_BY_EXPERIENCE

(EVENT MANAGEMENT, EXECUTIVE DIRECTOR)

5.4 Agrégation et noyaux

Une fois les faits à exprimer choisis, il faut déterminer où les insérer dans la lettre. Une liste de noyaux est produite par le générateur de section. Les faits sont ensuite insérés dans les positions de ces noyaux.

5.4.1 Ordonnancement des phrases : Noyaux

Le développement de BPPGen s'inspire de la la RST, dont la notion d'énoncé noyau et d'énoncés satellites a été préservée. Les phrases variables sont introduites par un *noyau* : ces noyaux assurent la cohésion globale du texte. Le noyau est un patron dont

les arguments sont un fait ou une conjonction de faits, et qui peut être lexicalisé de plusieurs manières différentes avec le même mécanisme de substitution et de manipulation syntaxique que les relations.

5.4.2 Présentation des noyaux

Comme nous l'avions dit plus tôt, les noyaux ont été créés pour simuler la RST. La RST construit un arbre entre les faits, dont les arêtes sont étiquetées avec une relation rhétorique. Un exemple de relation rhétorique est l'OPPOSITION. Nous pourrions reproduire cette relation par un noyau nommé OPPOSITION(X1, X2), qui signifie que les faits X1 et X2 sont vrais et qu'il est surprenant que X2 soit vrai en même temps que X1. Le noyau OPPOSITION pourrait être lexicalisé comme à l'exemple 5.2.

Exemple 5.2: Trois lexicalisations pour le noyau OPPOSITION(X1,X2)

OPPOSITION*(X1, X2)* $\Rightarrow$

#1 `S(C('mais'), X1, X2))` $\rightarrow$ *X1 mais X2.*

#2 `S(C('bien que'), X1, X2))` $\rightarrow$ *X1, bien que X2.*

#3 `S(C('quoique'), X1, X2))` $\rightarrow$ *X1, quoique X2.*

Dans BPPGen, rien n'exige que le noyau soit une relation rhétorique élémentaire. Il suffit que ce soit une structure dans laquelle on peut insérer des phrases. Par exemple, les lettres de recrutement sont configurées avec le noyau unaire présenté à l'exemple 5.3.

Exemple 5.3: Lexicalisation et réalisation du noyau MOST_IMPORTANT

```
S(X,
  SP(S(NP(Pro('which')),
       VP(V('be'),
          AdvP(Adv('exactly')),
          SP(S(NP(Pro('what')),
             S(NP(D('my').pe(1), N('client')),
               VP(V('be').t('p').pe(3), V('look').t(pr),
               PP(P('for')))))))).b(','))
```

$\Rightarrow$ *X, which is exactly what my client is looking for.*

5.5 Fonctionnement des noyaux

Nous avons présenté les noyaux, les structures dans lesquelles viennent s'insérer les faits, définissions-les plus formellement.

Un noyau est un nom et une liste. Les éléments de cette liste sont des ensembles de faits uniquement déterminés par le nom du noyau et la position dans la liste du noyau. Lors de la lexicalisation, les faits d'un même ensemble seront agrégés si possible, juxtaposés sinon. Chaque position correspond à une subordonnée ou une phrase. Cette phrase est soit la lexicalisation d'un fait, soit la lexicalisation de plusieurs faits (par agrégation ou par juxtaposition). L'exemple 5.4 illustre un noyau unaire sont l'unique argument est une conjonction de deux faits.

Exemple 5.4: Lexicalisation et réalisation du noyau MOST_IMPORTANT agrégé

MOST_IMPORTANT*({phrase1, phrase2})* ⇒

```
S(CP(C('and'),
     phrase1,
     phrase2),
  SP(S(NP(Pro('which')),
       VP(V('be'),
          AdvP(Adv('exactly')),
          SP(S(NP(Pro('what')),S(
                  NP(D('my').pe(1), N('client')),
                  VP(V('be').t('p').pe(3), V('look').t(pr),
                  PP(P('for')))))))).b(','))
```

⇒ *phrase1 and phrase2, which is exactly what my client is looking for.*

Les noyaux servent à placer les faits dans un arbre syntaxique (par substitution de variables). Comment déterminer à quel endroit sera exprimé un fait ? Nous formulons ce problème comme un problème d'affectation.

Par exemple, nous souhaitons que les faits de type PROFILE_SKILL_PROFICIENCY se trouvent dans le corps du paragraphe si possible ; et nous permettons que sa phrase comporte d'autres faits. PROFILE_SKILL_PROFICIENCY est effectivement le type de faits le plus fréquent, et il est préférable de laisser aux faits plus importants la tête du paragraphe. Un des faits plus importants est PROFILE_SKILL_EXPERTISE : nous l'utiliserons dans les exemples qui suivent.

Deux ensembles valeurs sont associées à chaque position de noyau. Le premier ensemble de valeur est une affinité entre la position de noyau et chaque relation admissible.

Une plus grande affinité entre un fait et un noyau implique que le fait à une plus grande chance d'être associé à cette position de noyau. L'autre valeur est l'aversion au poids, qui réduit la probabilité d'associer un fait à une position de noyau en fonction du nombre de faits qui ont déjà été associés à celle-ci. Ces deux valeurs sont fixées manuellement dans le fichier de configuration décrit à la section 7.3 (p. 73).

Nous identifions donc la première position du noyau ASSERTIVE, attribuons une affinité de 2 à PROFILE_SKILL_PROFICIENCY et une *aversion-au-poids* de 1,1 (ce qui favorise les énumérations).

Notons que les exemples qui suivent ne représentent pas un scénario réaliste : le nombre de faits et de noyaux de l'exemple 1.1 compliquerait la lecture.

Exemple 5.5: Opération des noyaux : entrées

Noyaux

FIRST_SENTENCE*(X1)*

ASSERTIVE*(X2, X3)*

Faits

PROFILE_SKILL_PROFICIENCY(ADVERSTISEMENT) *(f1)*

PROFILE_SKILL_PROFICIENCY(MARKETING MANAGEMENT) *(f2)*

PROFILE_SKILL_PROFICIENCY(PUBLIC RELATIONS) *(f3)*

PROFILE_SKILL_EXPERTISE(MARKETING) *(f4)*

Tableau 5.I : Exemple de matrice d'affinités

	X1	X2	X3
PROFILE_SKILL_PROFICIENCY	0.2	0.4	1
PROFILE_SKILL_EXPERTISE	1.1	1	1

5.5.1 Première ronde

L'affinité joue un rôle de score de base. Si l'affinité n'est pas définie entre une position de noyau et une relation, l'affectation à cette position de noyau est tout simplement interdite. Les affinités sont utilisées résoudre un problème classique d'affectation, en supposant que les positions de noyaux ne peuvent accueillir qu'un seul fait. Les autres faits seront attribués en deuxième ronde. L'exemple 5.6 donne un exemple d'affectation.

Les noyaux qui ont une position sans fait sont éliminés pour incomplétude. Les faits qui n'ont pas été appariés – incluant ceux qui étaient dans un noyau éliminé pour incomplétude – sont alors soumis à une deuxième ronde.

Exemple 5.6: Première ronde de l'exemple 5.5

X1 :f4

X2 :f1

X3 :f2

5.5.2 Deuxième ronde

L'*aversion-au-poids* pénalise les agrégations de faits. Autrement dit, elle régule la taille de l'ensemble de faits pour cette position de noyau. Le but de cette fonction est d'éviter que certaines phrases importantes ne deviennent trop longues, et de favoriser les juxtapositions dans les noyaux dont les lexicalisations sont construites pour les accepter harmonieusement.

La deuxième ronde de l'affectation est plus longue, parce que chaque décision d'affectation change le score entre les faits et les positions. En effet, attribuer un fait à une position de noyau change la taille, et force un recalcul du score de tous les autres faits à cette position de noyau. On associe les faits un-à-un jusqu'à ce qu'ils aient trouvé une

position sur un noyau.

L'*aversion-au-poids* permet de calculer le score d'affectation dans le deuxième tour. Le score respecte la formule suivante :

$$s_{F,P} = b_{F,P} \times n_P^{a_P} \times (I_{F \in P} + 1) \tag{5.1}$$

où F est le type de faits ; P est la position de noyau ; $b_{x,y}$ est l'affinité entre F et P ; n_P est le nombre de faits à la position de noyau P si on décide d'y placer le fait ; a_P est l'aversion au poids de P ; $I_{F \in P}$ est la fonction indicatrice qui retourne 1 si la position contient un fait qui peut être agrégé au fait courant, 0 sinon.

Remarquez que l'aversion au poids est un facteur exponentiel. Le facteur exponentiel permet de créer des positions de noyau qui attirent de plus en plus de faits. Cette possibilité a déterminé le choix de la fonction de score : un grande nombre d'énumérations très longues sont peu typiques du texte écrit par l'homme. En trouver une seule, par exemple, n'est pas aussi rare. Lorsque suffisamment de faits ont été insérés dans une position ayant une aversion-au-poids plus grande que 1, le terme exponentiel contribue à attirer tous les faits compatibles à cette position. Les deux positions du noyau ASSERTIVE ont une aversion-au-poids plus grande que 1.

L'effet de la fonction indicatrice est de doubler le score des faits qui peuvent être agrégés à un fait déjà présent. Ce modificateur est introduit pour favoriser les agrégations de faits.

L'exemple 5.7 illustre la seconde ronde de 5.6. Il reste un seul fait à placer. Il sera placé à la position X3, la deuxième position du deuxième noyau.

Exemple 5.7: Deuxième ronde de l'exemple 5.5

Choix pour f3, de type PROFILE_SKILL_PROFICIENCY *:*

X1 :f4 $s_{X1,f3} = 0.2 \times 2^{0.2} \times 1$

X2 :f1 $s_{X2,f3} = 0.4 \times 2^{1.5} \times 2$

X3 *:f2* $s_{X3,f3} = 1.0 \times 2^{1.3} \times 2$

5.6 Expressions référentielles

Nous imposons la restriction suivante dans la sélection des faits : une entité ne peut apparaître que dans un seul fait. Cette restriction nous permet d'éviter les problèmes liés aux expressions référentielles ; et le fait que les entités sont mentionnées une seule fois garantit que le sujet du paragraphe demeure le destinataire plutôt qu'une de ses compétences !

5.7 Choix lexical

5.7.1 Ressource de lexicalisation

L'acquisition automatique du lexique des entités est discutée à la section 4.3. Les lexicalisations des concepts d'ordre supérieurs (les faits et les noyaux) ont cependant été écrites à la main dans une syntaxe proche de celle de *JSRealB*, et écrites dans un fichier de configuration.

Le système fonctionne ainsi : chaque concept (les ENTITÉS, les RELATIONS et les NOYAUX) a une liste circulaire de lexicalisations. À chaque fois que le concept est rencontré, la lexicalisation pointée par le curseur est émise et le curseur est déplacé au suivant. L'exemple 5.8 montre les lexicalisations de INFERRED_REQUIRED_PROFILE_SKILL, un fait qui signifie qu'une compétence du profil est probablement implicite dans l'offre. La variable X2 sera remplacée par le deuxième argument de la relation ; les syntagmes

L réfèrent au lexique auxiliaire rotatif (LAR) décrit plus loin.

Exemple 5.8: Lexicalisations de la relation INFERRED_REQUIRED_PROFILE_SKILL

#1

```
S(SP(C('although'),
     S(NP(D('the'),N('offer')),
       VP(V('do'),
          AdvP(Adv('not')),
          V('mention' [t='i']),
          X2,
          PP(P('as'),
             NP(D('a'), AP(A('required')), N('skill')))))).a(',')),
      S(NP(Pro('I').pe(3).g('n')),
        VP(V('be'),
        AP(A('good')),
        SP(C('that'),
           S(L('you'),
           VP(V('know'),
           PP(P('about'),
              NP(Pro('me'[pe=3;g=n])))))))))),
```

⇒ *Although the offer does not mention X2 as a required skill, it is good that you know about it.*

#2

```
S(L('I'),
  VP(L('hypothesis_verb'),
     SP(S(NP(D('my')[pe=2], L('skill_noun'),
             PP(P('in'), X2)),
        L('useful_vp'))))))
```

⇒ *I deduce that your skill in X2 is useful.*

Un mécanisme supplémentaire est fourni : un lexique auxiliaire rotatif (LAR) ajoute davantage encore à la variété des constructions. Le LAR est utilisé dans les lexicalisations des concepts. Il est représenté par un syntagme **L** et lexicalisé après les concepts. Par exemple, le verbe d'hypothèse (*imaginer* ou *supposer*) est un élément du lexique auxiliaire, noté $L('imagine')$. Si deux faits de relations différentes contiennent ce syntagme, la première occurrence du verbe d'hypothèse sera *imaginer* ; la seconde, *supposer*. Ou encore, si on utilise une même lexicalisation deux fois, mais qu'un des noeuds de cette lexicalisation est un syntagme **L**, alors la valeur de ce L alternera aussi.

Cette méthode n'est pas sans problème. Si une entité doit être lexicalisée un assez grand nombre de fois, le lexique rotatif pour cette entrée retournera à son état initial. Le lexique traversera donc la même séquence de lexicalisations.

5.7.2 Ordre de lexicalisation des concepts

On peut parler de concepts supérieurs et inférieurs : les entités sont contenues dans les faits, et les faits sont contenus dans les noyaux. Lors du choix lexical, les concepts sont lexicalisés de bas en haut.

Pour chaque argument de chaque concept d'ordre supérieur, le système

1. lexicalise les concepts d'ordre inférieurs qui s'y trouvent ;
2. gère les agrégations ;
3. substitue la variable correspondante dans l'arbre de syntaxe ;
4. modifie l'arbre de syntaxe s'il le faut.

L'étape (1) a été décrite à la section précédente. L'étape (3) de substitution de variables dans un arbre de syntaxe n'est pas intéressante. Les étapes (2) et (4) méritent une

discussion.

Pour l'étape (2), un argument d'un fait agrégé peut contenir plus d'une entité (comme PROFILE_SKILL_PROFICIENCY(YOU, {MARKET STRATEGY, ADVERTISEMENT}) ; un argument d'un noyau agrégé peut contenir plus d'un fait (comme ASSERTIVE à la section **??**). Il faut donc lexicaliser des agrégations de concepts. Nous avons fait le choix d'agréger seulement les faits par la simple conjonction *et*.

Pour l'étape (4), il s'agit de faire certains réglages pour préserver la grammaticalité de la phrase. Par exemple, JSRealB ne gère pas la mise au mode subjonctif des verbes principaux des subordonnées circonstancielles introduites par *bien que*, *quoique*. Une règle de transformation permet d'appliquer cette transformation si l'arbre répond à certains patrons. Un engin de reconnaissance de patrons a été construit pour appliquer les règles.

5.8 Conclusion

Ce chapitre conclut la présentation de BPPGen. Nous avons donné les derniers détails du fonctionnement du générateur de texte. La section suivante illustre le fonctionnement de BPPGen avec un exemple supplémentaire.

CHAPITRE 6

EXEMPLE RÉCAPITULATIF

Le chapitre 3 présentait un premier exemple de BPPGen. Ici, nous illustrons l'opération du système sur un autre exemple. L'offre et le profil sont différents.

6.1 Entrées

Les tableaux 6.I, 6.II, 6.IV et 6.III correspondent aux tableaux présentés au chapitre 4.

6.1.1 Structuration du document

Les divisions sont les mêmes qu'au chapitre 3. Remarquez que la compagnie de l'offre, dont le nom a été modifié, est une société de recrutement. Ce problème survient fréquemment dans la base de données : la société qui a placé l'offre d'emploi est un intermédiaire et le véritable employeur demeure inconnu. Ce problème complique l'utilisation du nom de la société pour tirer davantage de conclusions dans BPPGen.

6.2 Sélection de contenu

Puisque la section *Qualifications* est présente, la production des faits est lancée. Au départ, BPPGen extrait l'information utilisée pour générer les faits. Cette extraction d'information du profil et de l'offre utilise la même normalisation décrite au chapitre 4 : on normalise chaque mot, et on cherche les séquences de mots connues.

Premièrement, une recherche des libellés de compétences dans la description de l'offre est lancée. L'offre mentionne directement les compétences CONSTRUCTION, PROJECT MANAGEMENT. CONTRACTOR WORK, CONSTRUCTION MANAGEMENT et MA-

Tableau 6.I : Exemple d'offre bien renseignée

title	Estimator
langid	en
place	Calgary, AB
company_name	Staffing Place Ltd.
description	Our client is an experienced General Contractor based in Calgary. Specializing in ICI Construction, both negotiated and LS projects, you will be an important element to putting together winning projects for clients. This role will report to one of the Partners/Principals based in Calgary. Summary of qualifications needed for this role : A construction related degree or diploma 5+ years experience estimating projects ranging from $5m-50$m within a General Contractor setting Lump Sum bids in ICI construction experience is an asset - however, we will consider experience gained outside of ICI Solid construction methodology with the ability to read and interpret conceptual drawings Experience in lump sum, and construction management contracts would be an asset Some exposure managing sub trade quotes, tender packages with a developing eye for potential risks within the bid process Ability to work successfully with project management team, consultants and sub trades Experience in take offs, some knowledge of project close out with a working knowledge of relevant estimating software such as Timberline

Tableau 6.II : Exemple de profil bien renseigné

countryCode	CA
personalBranding_pitch	
administrativeArea	CA-ON
personalBranding_claim	Electrical Estimator/Project Manager
skills	• Product Management • Budgets • Project Planning • Manufacturing • Value Engineering • Contract Management • Subcontracting • Electricians • Construction • MS Project • Project Estimation • Construction Management • Proposal Writing • Microsoft Project • Contractors • Change Orders

Tableau 6.III : Exemple d'expérience passée du profil au tableau 6.II

function	Project Manager/Estimator
startDate	1996-06
endDate	2014-05
missions	• Interpret bid documents & provide bid proposal using Accubid • Prepare change orders. • Purchase materials to secure projects. • Provide budget pricing for residential, commercial & institutional projects. • Obtained electrical permits from local jurisdictional agencies. • Conducted pre-installation meetings with site foreman, subcontractors, consultants and manufacture representative. • ...
companyName	Skynet Robotics Inc.

Tableau 6.IV : Exemple d'expérience courante du profil au tableau 6.II

function	Estimator
startDate	1988-03
endDate	1990-10
missions	Interpret bid documents & provide bid proposal for bid closing.
companyName	JC Budd Industries

Tableau 6.V : Exemple de lettre segmenté en sections (2)

Formule d'appel	Dear Mr. John Doe,
Présentation	I write this email after having read your LinkedIn profile. My name is Joan, I am recruiter with LittleBIGJob. Our company uses an artificial intelligence to make data-driven decisions in human resources. I only contact the best candidates based on fancy statistical techniques, and you've made the short list. The position you are being considered for is Estimator with Strato Staffing Ltd. Let me tell you why I would recommend you.
Qualifications	You are expert in Project Management, which is exactly what my client is looking for. You seem especially qualified because you have held the nearly identical job of Estimator previously, you are an expert in Construction and you are proficient in Construction Management and Contractor Work. You are an exceptional fit. I assume that your previous work as Project Manager has taught you much about Management. I do not much but your professional experience as Project Manager/Estimator suggests that you know a thing or two about Change Orders.
Contact	Would you be open to discuss this over the phone ? When would you be available ?
Salutations	Thank you for your time and your reply,

NAGEMENT. Deuxièmement, on recherche l'occupation dans le titre (title) de l'offre. On trouve la plus longue chaîne de caractères dans l'offre qui corresponde à une occupation connue. On y trouve l'occupation ESTIMATOR. Troisièmement, on cherche dans le texte libre du profil des compétences qui ont été trouvées à la première étape (pour produire les faits PROFILE_SKILL_EXPERTISE). Finalement, on identifie toutes les expériences passées et l'occupation courante (celle qui n'a pas de date de fin).

6.2.1 Sélection des associations

Pour chaque expérience on identifie toutes les compétences associées à celle-ci (les associations OCCUPATION → COMPÉTENCE). Ces compétences sont supposées acquises. Dans la lettre, on retient entre autres : ESTIMATOR → CHANGE ORDERS et PROJECT MANAGEMENT → MANAGEMENT. De même on utilisait les associations OCCUPATION → COMPÉTENCE pour l'emploi recherché : on retrouve ESTIMATOR → CHANGE ORDERS de nouveau.

6.2.2 Génération des faits

Les compétences extraites et les compétences déduites sont ensuite utilisées pour produire les faits de sorte à éviter les répétitions. Les faits produits pour l'exemple sont présentés au tableau 6.V.

Exemple 6.1: Les faits sélectionnés pour l'exemple 6.V

PROFILE_SKILL_PROFICIENCY(CONSTRUCTION MANAGEMENT),

PROFILE_SKILL_PROFICIENCY(CONTRACTOR WORK),

PROFILE_SKILL_EXPERTISE(PROJECT MANAGEMENT),

PROFILE_SKILL_EXPERTISE(CONSTRUCTION),

SAME_PREVIOUS_OCCUPATION(ESTIMATOR),

EXPLICIT_OFFER_SKILL_COVERED_ONLY_BY_EXPERIENCE(MANAGEMENT, PROJECT MANAGEMENT),
IMPLICIT_OFFER_SKILL_COVERED_ONLY_BY_EXPERIENCE(CHANGE ORDERS, PROJECT MANAGER/ESTIMATOR)

6.3 Appariement faits-noyaux

Le premier noyau a une plus grande affinité pour les faits de type PROFILE_SKILL_EXPERTISE. C'est donc le premier fait a être sélectionné. Les faits spéculatifs, ceux qui utilisent les associations décrites au chapitre 4, se trouvent effectivement à la fin du paragraphe dans les noyaux qui ont une plus grande affinité avec eux.

CHAPITRE 7

IMPLANTATION DE BPPGEN

Cette section décrit BPPGen comme un système concret, de sorte à permettre aux parties prenantes de comprendre et modifier le système. La discussion sera divisée en plusieurs parties : une brève discussion des choix technologiques ; l'architecture logicielle de BPPGen, sans égard à la nature du texte généré ; le fichier de configuration, qui contient les lexicalisations et les informations relatives aux noyaux ; le générateur de ressources, propre à la sélection de contenu de la section *Qualifications* ; et finalement une interface Web mise à la disposition des utilisateurs.

7.1 Choix technologiques

Le logiciel a été développé en Python2.7 pour des plate-formes UNIX. Python était tout indiqué pour plusieurs raisons : la grande disponibilité de librairies externes (notamment pour la manipulation de fichiers JSON, les bases de données, la racinisation, les serveurs Web), la rapidité du cycle de développement et la portabilité.

Les données étaient à l'origine des fichiers JSON. Pour accélérer les requêtes, nous avons transféré les enregistrements dans une base de données MongoDB. MongoDB est une base de données de type document dont les enregistrements sont dans un format BSON très proche du JSON des données originales.

7.2 Architecture logicielle de BPPGen

L'architecture de BPPGen respecte les étapes décrites aux chapitres précédentes. À chaque tâche correspond un objet responsable de l'accomplir.

Dans un premier temps, le générateur de texte reçoit en entrée un *DocumentPlanner*.

Le *DocumentPlanner* inspecte les entrées et crée une séquence de sections. Ces sections sont soit des fonctions quelconques qui retournent du texte JSRealB, soit un générateur de section.

Un générateur de section est un type prédéfini composé d'un générateur de faits et d'une séquence de noyaux prédéfinie. Le générateur de faits est un objet défini par l'utilisateur qui dépend fortement de l'application.

Dans un scénario typique, l'utilisateur se contentera de programmer des générateurs de section, qui utiliseront une source de données quelconques pour générer des faits. Les parties invariables peuvent être simplement écrites comme des chaînes de caractères : le constructeur du générateur de texte s'occupe du reste. Le fichier de configuration prend en charge le reste de l'application. Il suffit donc de produire des faits et de fournir le fichier de configuration qui contient les lexicalisations et les règles relatives aux noyaux. Pour la planification du document, si les générateurs de section sont fixes (comme dans la génération des lettres), il suffit de fournir une liste des générateurs de section ; mais un objet plus complexe respectant la signature du planificateur peut faire varier la structure pour convenir aux besoins de l'utilisateur.

7.3 Fichier de configuration

Un fichier de configuration JSON pour le générateur de section doit être utilisé. Ce fichier de configuration est composé de plusieurs attributs : entities (pour les entités) ; relations ; nuclei (pour les noyaux) ; et shorthands, des raccourcis qui permettent d'écrire plus succinctement des arbres de syntaxe pour accélérer l'écriture des arbres de syntaxe.

7.3.1 Entités

Le champ entities contient une table associative liant les identifiants d'entités à la liste de ses lexicalisations.

7.3.2 Relations

Ce champ est structuré de la même manière que les entités. Chaque identifiant de relation est liée à une liste de ses lexicalisations. La particularité des lexicalisations des relations est que ses lexicalisations comportent des variables, dans lesquelles s'insèrent les lexicalisations des entités ou des agrégations d'entités.

7.3.3 Noyaux

Ce champ est plus complexe. Comme pour les relations, à chaque noyau correspond une liste de lexicalisations dont certains syntagmes sont des variables, dans lesquelles s'insèrent la lexicalisation des faits ou des agrégations de faits.

Le fichier de configuration définit aussi les paramètres essentiels au fonctionnement des noyaux. Chaque noyau a donc une liste de positions de noyaux, qui correspondent aux variables dans les lexicalisations. À chaque position de noyau est associée une aversion-au-poids ainsi que les affinités entre cette position de noyau et chacune des relations admissibles à cette position.

7.4 Génération des ressources

BPPGen fournit un script qui génère les différentes ressources décrites au chapitre 4. Une des ressources générées est le fichier de configuration mentionné à l'étape précédente. Seules les entités sont générées automatiquement : la configuration des relations et des noyaux doit être écrite manuellement.

Le script utilise un objet nommé *Cooccurrence* qui permet d'observer les cooccurrences et de calculer les probabilités conditionnelles désirées. C'est finalement une distribution échantillonnale. Nous avons dû créer cet objet nous-mêmes pour permettre plus facilement la prise de cooccurrences entre plus de deux événements. Par exemple, nous

aurions pu observer des règles d'associations entre trois ou quatre Cette flexibilité n'a pas finalement pas été utilisée dans la production des faits.

L'autre ressource est utilisée dans les générations des faits de la section *Qualifications*. Cette ressource, également en format JSON, associe, pour chaque 20-quantile, une liste de 4-tuples composés de l'antécédent, de la conclusion, du nombre de cooccurrences et de la probabilité conditionnelle échantillonnale. Lors de la production des faits, on définit un seuil, et toutes les associations dont la probabilité conditionnelle échantillonnelle l'excède sont chargées en mémoire. Ces associations sont finalement utilisées pour produire les faits selon le sens que nous leur avons donné au chapitre 4.

Un objet intermédiaire utilisé dans la production des faits peut être utilisé pour inspecter ces associations simples. Malheureusement, cette ressource n'avait d'abord pas pour fonction d'explorer les données en général.

Il suffit de lancer le script nommé *build.py* pour générer ces ressources de A à Z. Pour décomposer le processus en sous-étapes, il faut altérer le code.

7.5 Interface Web

Une interface Web a été produite pour illustrer le fonctionnement de BPPGen. L'utilisateur télécharge une offre et un profil en utilisant soit un service Web, soit une base de données MongoDB comme celle décrite au chapitre 4. Dans un premier temps, on sélectionne (par identifiant ou par mots-clés) ou écrit une offre (figure 7.1). Ensuite, on peut sélectionner une liste de candidats de la même manière (figure 7.2). La lettre est générée en cliquant sur un bouton au bas de la présentation du profil.

Figure 7.1 : Capture d'écran de l'interface Web pour l'offre

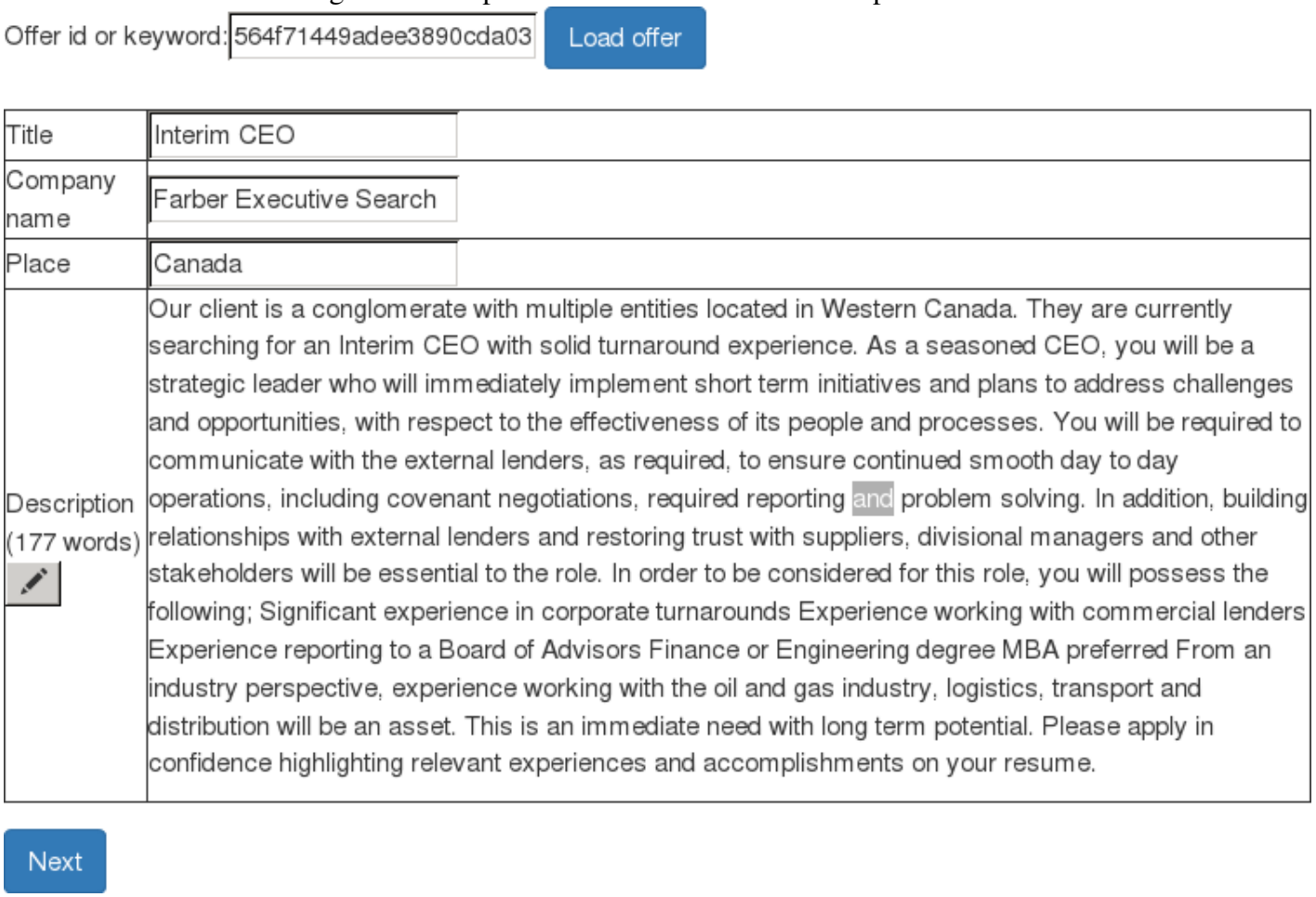
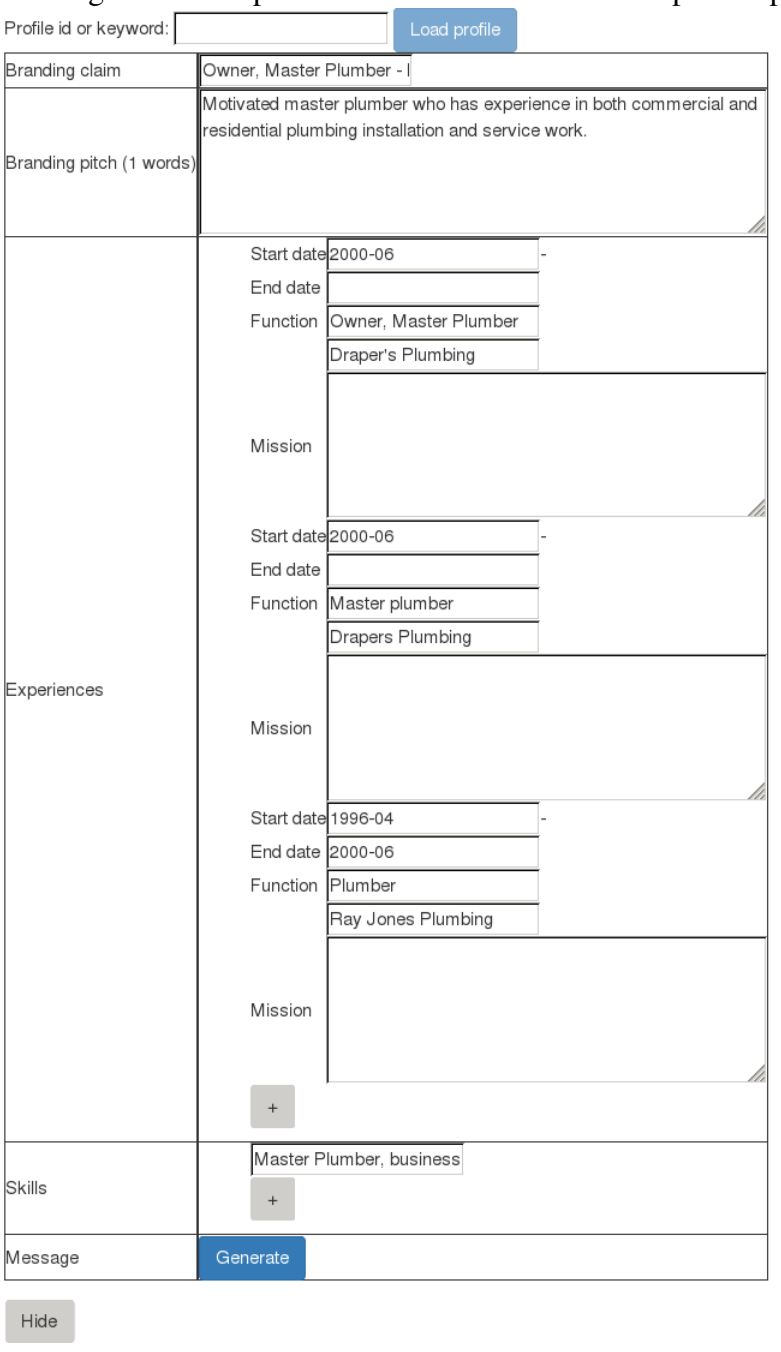


Figure 7.2 : Capture d'écran de l'interface Web pour le profil



CHAPITRE 8

TRAVAUX FUTURS

8.1 Évaluation

Si chacune des étapes de la génération de texte peut faire l'objet d'une évaluation spécifique, l'évaluation des systèmes de génération de texte prend généralement trois formes ([13]) : l'évaluation du contenu exprimé ; la qualité linguistique des textes produits, incluant la grammaire et le style ; l'adaptation à la tâche proposée.

L'évaluation du contenu exprimé a été capitale dans le développement du système. Les seuils de signification présentés à 4.4.1 ont été déterminés par des *associations-sentinelles*, puis vérifiés par des exemples générés manuellement dans les tests. Cependant, une évaluation par rappel, précision et score $F1$ aurait supposé l'existence d'un standard. Or, il n'existe aucune ressource qui mette en lien les occupations telles que définies dans ce travail et les compétences de LinkedIn. Il aurait fallu en construire un nombre suffisant, ce qui n'était pas possible étant donné les ressources du projet. L'évaluation aurait aussi été compliquée par le problème des compétences sous-entendues : la lettre doit-elle mentionner qu'une COMPTABLE doit connaître la COMPTABILITÉ ?

L'évaluation linguistique est plus difficile à quantifier. Une manière de procéder aurait été de demander à plusieurs personnes de noter, par exemple sur 5, la qualité d'un certain nombre de textes. S'il existe des cas de textes mal écrits, tous les lecteurs ne seront pas d'accord sur la qualité linguistique des textes produits. Étant donné les paramètres de la lettre de recrutement, nous demandions périodiquement au partenaire commercial de donner son avis sur les lettres. La réponse n'était pas quantitative, mais a permis de guider certains choix, notamment la longueur des différentes sections.

L'évaluation par l'adaptation à la tâche proposée n'a pas été accomplie bien que

des plans aient été prévus à cet effet. La méthode d'évaluation envisagée aurait exigé que le système soit déployé, et nous n'avions pas le temps d'attendre la fin du projet *ButterflyPredictiveProject*. Deux types de lettres seraient envoyés aux clients de LBJ : la première aurait été la lettre actuellement envoyée, un simple patron ; la deuxième aurait été écrite par BPPGen. Les deux lettres aurait contenu un hyperlien que le candidat aurait cliqué. Dans BPPGen, l'hyperlien se serait trouvé naturellement à la section *contact*. Il aurait finalement suffi de comparer le nombre de fois que l'hyperlien a été suivi pour mesurer le succès. Une telle évaluation pourrait toujours être conduite en apportant des modifications mineures à BPPGen.

Finalement, nous envisagions un troisième type de lettre pour comparer l'efficacité relative de la génération de texte et de la génération de graphiques. Elle aurait contenu une présentation graphique des associations utilisées pour construire les faits de la section *Qualifications*. Par exemple, on pourrait montrer les COMPÉTENCES généralement requises pour l'OCCUPATION de l'offre, selon qu'elles sont possédées par le candidat ou non. L'idée serait ici d'intéresser le destinataire en lui présentant des graphiques adaptés à sa carrière.

8.2 Ressource lexicale

La génération automatique de ressources lexicales pose problème. Les entrées lexicales correspondant aux entités étaient générées sans établir de correspondance à une ressource externe. Par exemple, la compétence WAITRESS en anglais, un nom féminin, n'était pas reconnu comme féminin.

Même si une simple référence à un dictionnaire aurait permis de régler certains cas, cette méthode ne pouvait pas être généralisée pour les entrées lexicales composées de plusieurs mots. Par exemple, pour déterminer le genre grammatical de l'occupation ADJOINTE ADMINISTRATIVE, il faut comprendre la structure du syntagme nominal de l'en-

tité. Il existe des algorithmes d'apprentissage permettant de faire l'analyse syntaxique : or, ces analyses dépendent d'un corpus annoté externe, et la ressemblance à ce corpus est déterminante pour le succès de l'analyse syntaxique. Or, nos syntagmes nominaux, trouvés sans contexte, sont très dissemblables aux modèles disponibles, entraînés sur des textes constitués de phrases complètes. Nous ne disposons pas de corpus d'entraînement de groupes nominaux simples.

8.3 Intégration d'un modèle de langue

Les lexicalisations des faits ont été écrites soigneusement pour minimiser les tournures non-idiomatiques. La sélection de la préposition introduisant un syntagme nominal posait en effet problème. Par exemple, on dira *expert in auditing*, mais on dira *expert on animal biology*. Le problème était si important que les lexicalisations ont été choisies pour minimiser ces expressions. La répétition de la préposition dans les syntagmes en conjonction[1] a ainsi été désactivée : elle mettait en valeur cette faille du système.

Une solution envisagée est l'utilisation d'un modèle de langue entraîné sur un plus grand corpus pour détecter les situations dans lesquelles un choix de préposition est trop improbable. Il aurait suffit d'utiliser un modèle de langue et de comparer la probabilité de la sélection de chaque préposition en fonction du contexte, et de corriger si l'écart est assez grand.

8.4 Profils psychologiques

L'adaptation des lettres au profil psychologique était une caractéristique désirée de BPPGen. En particulier, le partenaire commercial désirait écrire des lettres différentes selon le type *DISC* du destinataire. *DISC* a été développé en 1928 par un psychologue

[1] Par exemple, on répète la préposition dans *Elle se rend souvent **en** France et **en** Allemagne*.

américain. C'est un test psychométrique propriétaire - c'est-à-dire qu'il faut payer pour accéder au test -

Le problème principal était la pauvreté des données. En effet, il était pratiquement impossible de découvrir le profil *DISC* avec la moindre fiabilité. Deux approches ont été envisagés pour découvrir le profil *DISC* : utiliser un ensemble d'entraînement associant un profil à un type pour entraîner un classificateur ; créer un classificateur avec un vocabulaire prédéfini. La première option se heurte au problème des données : il aurait fallu créer soi-même cette ressource, en attribuant aléatoirement un type *DISC* à certains candidats, inconnus. La deuxième option était arbitraire puisqu'aucun tel vocabulaire n'a été développé pour *DISC*.

Un type était donc assigné aléatoirement ou comme entrée. Les phrases spécifiques à un type de personnalité étaient si spécifiques qu'une erreur dans l'évaluation du profil psychologique aurait eu un effet négatif. Faire l'éloge du leadership d'une personne discrète réduirait l'efficacité du contact. Nous avons donc modifié les phrases pour qu'elles soient d'application plus générale, au point qu'elles s'appliquaient à tous. Les profils *DISC* n'étaient donc plus utiles et cette partie de la lettre a été désactivée.

Les deux options mentionnées ci-haut avaient besoin de ressources externes pour être appliquées. Or, même si ces ressources existaient, il aurait été impossible de les appliquer aux profils des candidats. Premièrement, nous avons vu à la section Description de la base de données (p.29) que les profils sont majoritairement si peu renseignés qu'il est peu probable qu'ils contiennent des informations psychologiques pertinentes. Deuxièmement, les profils qui contiennent beaucoup de texte libre sont écrits dans un style propre à la promotion de soi-même. Le développement d'une image professionnelle est un but des réseaux sociaux professionnels comme LinkedIn. S'il fallait croire les profils, la majorité des individus se distinguent par leur leadership, leur autonomie, leur esprit d'équipe et leur approche orientée résultats. Ces traits correspondent presque

exactement à une des quatre catégories de *DISC*, le *I* pour influent.

8.5 Utilisation des informations sur la scolarité

Une des informations les plus importantes relatives au recrutement est l'éducation. Certains emplois exigent systématiquement un certain type de formation. Or, BPPGen n'examine pas cet aspect du recrutement parce qu'il suppose que cet aspect a été pris en compte lors de l'appariement.

8.6 Règles d'associations

Les faits étaient générés par une simple analyse fréquentielle et une sélection de seuils. Le but de ces analyses étant simplement de générer du contenu, nous n'avons pas eu la chance d'explorer des analyses plus fines...

Les faits ont été limités à des associations entre entités uniques : un antécédent simple, comme *Le candidat est compétent en C++*, et une conséquence simple, comme *Le candidat est programmeur*. Nous aurions pu adapter l'algorithme APRIORI décrit dans [1] pour identifier des ensembles plus importants.

L'algorithme APRIORI analyse des *transactions* (principalement, des paniers d'achats) et répond à la question : Quels items sont achetés ensemble ? Il serait simple d'adapter cet algorithme aux profils de candidats : les entités contenus dans un profil sont alors simplement associées.

8.7 Géolocalisation

La proximité géographique est déterminante pour une grande partie des candidats. Une amélioration importante de la lettre consisterait à indiquer si l'employeur est dans la même ville, dans une ville voisine, ou si l'acceptation signifie que le candidat doit

déménager.

8.8 Longueur de la lettre et des phrases

Les lettres générées étaient jugées trop longues par le partenaire commercial. BPPGen n'a pas été corrigé pour ce rapport, pour la simple et bonne raison qu'il est plus facile pour un GAT d'écourter le texte que de l'allonger. Par ailleurs, les exemples sélectionnés pour ce mémoire étaient exceptionnels par le nombre et par la variété de faits à exprimer.

Le partenaire commercial a aussi fait observer que les phrases devaient être très brèves, ce que BPPGen ne fait pas. Ce comportement peut être obtenu en multipliant le nombre de noyaux et en simplifiant les lexicalisations des faits et des noyaux.

CHAPITRE 9

CONCLUSION

Nous avons décrit les GAT, au sens de Ehud Reiter, et les avons distingué des autres systèmes permettant de générer du texte.

Nous nous sommes inspiré des travaux de cet auteur pour construire un système qui reçoit en entrée un profil issu des réseaux sociaux, une offre d'emploi d'un site Web pour produire une représentation abstraite du contenu du texte, en passant par les ENTITÉS, FAITS. Le système ordonne, agrège et filtre les faits par la méthode des noyaux, un appariement en deux temps ; produit des arbres de syntaxe en utilisant (1) une ressource de lexicalisation qui fait le pont entre la représentation abstraite et (2) la manipulation d'arbres de syntaxe.

Les ressources utilisées pour compléter ces étapes ont été générés à partir d'une base de profils de réseaux sociaux. Des statistiques sont colligées sur les cooccurrences dans cette base de profils, et ces statistiques sont utilisées pour produire diverses règles d'association entre les compétences et les titres d'emploi. Ces règles d'association sont ensuite utilisées pour générer automatiquement un contenu textuel abstrait sous forme de faits. Ces faits trouvent une place dans des noyaux prédéfinis. Ces structures sont lexicalisées en arbres de syntaxe grâce à un fichier de configuration, puis transformées en langage naturel en utilisant un logiciel spécialisé de réalisation.

BIBLIOGRAPHIE

[1] Roberto J BAYARDO JR. “Efficiently mining long patterns from databases”. In : *ACM Sigmod Record* 27.2 (1998), p. 85–93.

[2] Steven BIRD. “NLTK : the natural language toolkit”. In : *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics. 2006, p. 69–72.

[3] Robert DALE et Ehud REITER. “Computational interpretations of the Gricean maxims in the generation of referring expressions”. In : *Cognitive science* 19.2 (1995), p. 233–263.

[4] Richard DAWKINS. *Postmodernism disrobed*. 1998.

[5] Michael FRIEDMAN. *Reconsidering logical positivism*. Cambridge University Press, 1999.

[6] Albert GATT et Ehud REITER. “SimpleNLG : A realisation engine for practical applications”. In : *Proceedings of the 12th European Workshop on Natural Language Generation*. Association for Computational Linguistics. 2009, p. 90–93.

[7] Marti A HEARST. “TextTiling : Segmenting text into multi-paragraph subtopic passages”. In : *Computational linguistics* 23.1 (1997), p. 33–64.

[8] Remy KESSLER, Guy LAPALME et Éric TONDO. “Génération d’une ontologie dans le domaine des ressources humaines”. In : *CORIA 2016*. Toulouse, 2016. URL : `https://www.irit.fr/sdnri2016/SDNRI2016.pdf`.

[9] Emiel KRAHMER et Kees VAN DEEMTER. “Computational generation of referring expressions : A survey”. In : *Computational Linguistics* 38.1 (2012), p. 173–218.

[10] Marco LUI et Timothy BALDWIN. “langid. py : An off-the-shelf language identification tool”. In : *Proceedings of the ACL 2012 system demonstrations*. Association for Computational Linguistics. 2012, p. 25–30.

[11] Dieng MAMADOU ALIDOU. “Développement d’un système d’appariement pour le e-recrutement”. Mém.de mast. Montréal, Canada : Université de Montréal, 2016.

[12] William C MANN et Sandra A THOMPSON. “Rhetorical structure theory : Toward a functional theory of text organization”. In : *Text-Interdisciplinary Journal for the Study of Discourse* 8.3 (1988), p. 243–281.

[13] Chris MELLISH et Robert DALE. “Evaluation in the context of natural language generation”. In : *Computer Speech & Language* 12.4 (1998), p. 349–373.

[14] Paul MOLINS et Guy LAPALME. “JSrealB : A bilingual text realizer for web programming”. In : *ENLG 2015* (2015), p. 109.

[15] Martin F PORTER. *Snowball : A language for stemming algorithms*. 2001.

[16] François PORTET et al. “Automatic generation of textual summaries from neonatal intensive care data”. In : *Artificial Intelligence* 173.7 (1997), p. 57–87.

[17] Ehud REITER. “NLG vs. templates”. In : *arXiv preprint cmp-lg/9504013* (1995).

[18] Ehud REITER, Robert DALE et Zhiwei FENG. *Building natural language generation systems*. T. 33. MIT Press, 2000.

[19] Ehud REITER et al. “The importance of narrative and other lessons from an evaluation of an NLG system that summarises clinical data”. In : *Proceedings of the Fifth International Natural Language Generation Conference*. Association for Computational Linguistics. 2008, p. 147–156.

[20] Stuart Jonathan RUSSELL et al. *Artificial intelligence : a modern approach*. T. 2. Prentice hall Upper Saddle River, 2003.

[21] Tony C SMITH et Ian H WITTEN. “Models for computer generated parody”. In : (1993).

[22] Alan SOKAL et Jean BRICMONT. *Impostures intellectuelles*. Odile Jacob, 1997.

[23] Martin SUNDERMEYER, Ralf SCHLÜTER et Hermann NEY. “LSTM Neural Networks for Language Modeling.” In : *Interspeech*. 2012, p. 194–197.

[24] Ilya SUTSKEVER, James MARTENS et Geoffrey E HINTON. “Generating text with recurrent neural networks”. In : *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, p. 1017–1024.